



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA DE POSTGRADO Y EDUCACIÓN CONTINUA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

**APLICACIÓN DE MODELOS DE DEEP LEARNING NO SUPERVISADO EN
EL MANTENIMIENTO PREDICTIVO PARA LA DETECCIÓN TEMPRANA
DE ANOMALÍAS EN EQUIPOS INDUSTRIALES CON CONDICIONES
OPERACIONALES VARIABLES**

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS DE DATOS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

EMILIO RAÚL ROA RAMÍREZ

PROFESORA GUÍA:
VIVIANA MERUANE NARANJO

MIEMBROS DE LA COMISIÓN:
MARCOS ORCHARD CONCHA
JORGE SILVA SÁNCHEZ

SANTIAGO DE CHILE

2024

RESUMEN: TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS DE DATOS Y MEMORIA
PARA OPTAR AL TÍTULO DE INGENIERO CIVIL
ELÉCTRICO
POR: EMILIO RAÚL ROA RAMÍREZ
FECHA: 2024
PROF. GUÍA: VIVIANA MERUANE

APLICACIÓN DE MODELOS DE DEEP LEARNING NO SUPERVISADO EN EL MANTENIMIENTO PREDICTIVO PARA LA DETECCIÓN TEMPRANA DE ANOMALÍAS EN EQUIPOS INDUSTRIALES CON CONDICIONES OPERACIONALES VARIABLES

El mantenimiento predictivo se ha convertido en un elemento crucial para la optimización de la productividad industrial, con el potencial de prevenir fallos y minimizar el tiempo de inactividad. En este contexto, los modelos de deep learning no supervisados emergen como herramientas prometedoras por su capacidad para identificar patrones ocultos y anticipar eventos anómalos en datos de equipos industriales.

Este trabajo de tesis aborda la implementación de algoritmos de deep learning no supervisados, centrándose en su viabilidad para detectar anomalías y predecir fallas con antelación. Se llevó a cabo un estudio comparativo entre algoritmos clásicos de machine learning y algoritmos de deep learning, donde las arquitecturas basadas en autoencoders demostraron un desempeño superior en un conjunto de datos controlado (SKAB), y posteriormente validando en un entorno real de una planta de ósmosis inversa con bombas de alta presión.

El estudio también exploró la utilidad de un umbral variable basado en la Teoría del Valor Extremo para reducir la duración de falsas alertas generadas por los cambios en los modos operativos de los equipos. Esta técnica no solo disminuyó la frecuencia de falsos positivos sino que también acortó la duración de las alertas innecesarias, manteniendo la precisión en la detección de fallas genuinas.

Las limitaciones encontradas, como la necesidad de reentrenamiento continuo y la adaptación a nuevos modos operativos, sugieren áreas para futuras investigaciones. El trabajo sugiere la posibilidad de desarrollar modelos más dinámicos que se ajusten automáticamente a los cambios y el uso de técnicas de clustering para una identificación precisa de los modos operativos.

Las conclusiones de esta investigación refuerzan el potencial del mantenimiento predictivo asistido por deep learning en entornos industriales, marcando un avance hacia operaciones más eficientes y confiables. Se anticipa que los hallazgos de este estudio sirvan de base para futuras investigaciones y la adopción de soluciones más inteligentes en la industria.

Palabras clave: mantenimiento predictivo, deep learning no supervisado, detección de anomalías, Autoencoders, umbral variable, Teoría del Valor Extremo.

Para mamá y papá

Muchas gracias

Agradecimientos

Quisiera comenzar expresando mi agradecimiento a mi profesora guía Viviana, cuya ayuda, paciencia y apoyo fueron decisivos durante todo el proceso de investigación y escritura de esta tesis.

A mi familia, por su amor incondicional, paciencia y comprensión inagotables durante todo mi periodo de estudios.

No puedo dejar de mencionar el apoyo constante de mis amigos y compañeros, en especial a Nicolás y Vicente, quienes me ofrecieron su aliento y fueron una fuente de motivación constante.

A todos quienes de una manera u otra contribuyeron a la realización de este trabajo, les estoy profundamente agradecido. Gracias por creer en mí.

Por último pero no menos importante, a Dios por darme las capacidades y cualidades divinas para superar el estrés y presión, ayudándome a disfrutar y estar consciente de cada etapa de este proceso.

Tabla de Contenido

1. Introducción	1
1.1. Antecedentes	1
1.2. Objetivos	3
1.2.1. Objetivo General	3
1.2.2. Objetivos Específicos	3
2. Marco Teórico	5
2.1. Mantenimiento Industrial	5
2.1.1. Monitoreo Condición	6
2.2. Aprendizaje Profundo	7
2.2.1. Redes Neuronales Artificiales	7
2.2.2. Redes neuronales convolucionales	8
2.2.3. Redes Neuronales Recurrentes	10
2.2.4. Entrenamientos de Redes Neuronales	13
2.3. Aprendizaje Profundo en Mantenimiento Predictivo	14
2.3.1. Detección de Anomalías	14
2.3.2. Tipos de anomalías	15
2.3.3. Métodos de Deep Learning para la Detección de Anomalías	16
2.3.3.1. Autoencoders	17
2.3.3.2. AutoRNN	18
2.3.3.3. Variational Autoencoders	18
2.4. Detección de Anomalías en Condiciones de Cambios Operacionales	20
2.5. Evaluación y Validación de Modelos	21
2.5.1. Métricas de Evaluación	22
3. Metodología	25
3.1. Selección y descripción de los conjuntos de datos	25
3.2. Pre-procesamiento y Análisis exploratorio de los datos	26
3.3. Implementación de modelos	26
3.4. Evaluación y comparación de métricas	27
4. Casos de Estudio	29
4.1. Base de Datos SKAB	29
4.2. Conjunto de datos de operación industrial real	31
5. Resultados	34
5.1. Resultados de la Base de Datos SKAB	34
5.1.1. Descripción de la Base de Datos SKAB	34

5.1.2.	Análisis Exploratorio de Datos (EDA) de SKAB	35
5.1.3.	Arquitectura de los Modelos Implementados	37
5.1.3.1.	Isolation Forest	37
5.1.3.2.	Elliptic Envelope	38
5.1.3.3.	Autoencoder	38
5.1.3.4.	Autoencoder Convolutacional (AutoCNN)	39
5.1.3.5.	Autoencoder Recurrente (Auto RNN)	41
5.1.3.6.	Implementación	42
5.1.4.	Resultados de Modelos en SKAB	42
5.1.4.1.	Modelos Clásicos	42
5.1.4.2.	Modelos Deep Learning	42
5.1.4.3.	Resumen Métricas	43
5.2.	Resultados Base de Datos Operación Industrial Real	45
5.2.1.	Descripción de la Base de Datos Bombas de Alta Presión y Booster .	45
5.2.2.	Naturaleza de los Datos y Registro de Fallas	46
5.2.3.	Análisis Exploratorio de Datos (EDA)	46
5.2.3.1.	Bomba Booster	46
5.2.4.	Preprocesamiento de los datos	52
5.2.5.	Arquitectura del modelo implementado	53
5.2.6.	Resultados Modelos	54
5.2.6.1.	Casos de éxito	54
5.2.6.2.	Falsos positivos	58
5.2.7.	Resumen Métricas	59
5.2.8.	Umbral Adaptativo Basado en la Teoría del Valor Extremo	61
5.2.8.1.	Comparación de casos con umbral variable	62
5.2.8.2.	Comparación de métricas con y sin umbral variable	64
6.	Discusiones	65
6.1.	Discusión de Resultados	65
6.2.	Limitaciones e implicaciones prácticas	66
6.3.	Recomendaciones y Trabajo Futuro	67
7.	Conclusiones	69
	Bibliografía	71
	Anexos	75
A.	Resultados Base de Datos SKAB	75
A.1.	EDA	75
A.2.	Ejemplo SKAB	78
B.	Resultados Base de Datos Bombas de Alta Presión y Booster	84
B.1.	EDA	84

Índice de Tablas

5.1.	Grilla de RandomSearch para la Optimización de Hiperparámetros AutoMLP .	39
5.2.	Grilla de RandomSearch para la Optimización de Hiperparámetros AutoCNN .	40
5.3.	Grilla de RandomSearch para la Optimización de Hiperparámetros Autoencoder LSTM	41
5.4.	Comparativa de métricas de modelos en detección de anomalías y puntos de cambio	44
5.5.	Resumen alertas generadas por modelo IA	60
5.6.	Matriz de confusión	60
5.7.	Métricas de sensibilidad y precisión	61
5.8.	Comparación de métricas de rendimiento con y sin el uso del umbral variable.	64
A.1.	Resumen estadístico de los experimentos	75
A.3.	Tabla resumen para los 34 experimentos	76
B.2.	Estadísticas Descriptivas de los Sensores de Bombas Booster	89

Índice de Ilustraciones

2.1.	La siguiente figura ilustra la lógica computacional de una RNN en 3 pasos de tiempo adyacentes	11
2.2.	Anomalías individuales v/s colectivas. Fuente.	16
2.3.	Celda LSTM. Fuente.	19
2.4.	Celda GRU. Fuente.	20
2.5.	Matriz de Confusión. Elaboración Propia.	22
2.6.	Clasificación métricas de evaluación. Fuente.	23
4.1.	Panel frontal sistema de circulación, control y monitoreo de agua. [43]	30
4.2.	Diagrama equipos participantes en el proceso de ósmosis inversa. Elaboración propia.	32
5.1.	Matriz de Correlación BBDD SKAB.	36
5.2.	Matriz de Confusión Detección de Anomalías	42
5.3.	Matriz de Confusión Detección de Anomalías Métodos Deep Learning	43
5.4.	Matriz de Correlación Variables Bba Booster	49
5.5.	Gráfica de Error Método del Codo Bba Booster	50
5.6.	Análisis PCA Bba Booster	51
5.7.	Análisis UMAP Bba Booster	51
5.8.	Velocidad Eje Bba Booster separado por cluster	52
5.9.	Leyenda para gráficos de KPI	55
5.10.	Caso de éxito bomba booster. Descripción Falla: Filtración por la línea de refrigeración de la booster Rack 11. Se logra una anticipación de 2 días.	55
5.11.	Caso de éxito bomba HP Rack 08. Descripción Falla: Alta presión de descarga bomba HP. Anticipación de 3 horas y 25 minutos.	56
5.12.	Caso de éxito bomba HP Rack 01. Descripción Fallas: (1) Rotación Inversa (2) Filtración válvula check (3) Alta presión de descarga . Anticipaciones (1) 4 hrs 10 minutos (2) 4 hrs 24 minutos (3) 3 hrs 46 minutos	56
5.13.	Las imagenes muestran la representación gráfica de los datos de sensores y las predicciones realizadas por los modelos de inteligencia artificial para la bomba booster del rack 11. En este gráfico, los puntos negros representan las predicciones efectuadas por los modelos de IA. Por otro lado, los puntos de colores reflejan los valores reales registrados por los sensores: los puntos rojos indican una anomalía severa, los amarillos señalan una anomalía leve, y los verdes representan un comportamiento normal del equipo. Además, las franjas celestes en el gráfico marcan los periodos de detención del equipo, proporcionando un contexto visual claro sobre los momentos en los que la maquinaria estuvo fuera de servicio.	57
5.14.	Caso de falsa alerta bomba booster Rack 06	59
5.15.	Caso de falsa alerta bomba booster Rack 04	59

5.16.	Umbral Variable Rack 06	62
5.17.	Umbral Variable Rack 04	63
5.18.	Umbral Variable Rack 11	63
A.1.	Pairplot BBDD SKAB	77
A.2.	Señales para experimento 20, BBDD SKAB	78
A.3.	Ejemplo Predicciones Isolation Forest	79
A.4.	Ejemplo Predicciones Elliptic Envelope	79
A.5.	Ejemplo detección de anomalías CNN Autoencoder. Se realiza comparación entre las etiquetas y las predicciones del modelo.	80
A.6.	Ejemplo detección de puntos de cambio CNN Autoencoder	80
A.7.	Reconstrucciones CNN Autoencoder. Los puntos negros corresponden a las reconstrucciones que realiza el modelo CNN AE en base a la señal original. Los colores verde indican que es un estado sin anomalías, mientras que amarillo y rojo corresponden a anomalías leves y graves respectivamente.	80
A.8.	Ejemplo detección de anomalías MLP Autoencoder. Se realiza comparación entre las etiquetas y las predicciones del modelo.	81
A.9.	Ejemplo detección de puntos de cambio MLP Autoencoder	81
A.10.	Reconstrucciones MLP Autoencoder. Los puntos negros corresponden a las reconstrucciones que realiza el modelo MLP AE en base a la señal original. Los colores verde indican que es un estado sin anomalías, mientras que amarillo y rojo corresponden a anomalías leves y graves respectivamente.	81
A.11.	Ejemplo detección de anomalías GRU Autoencoder. Se realiza comparación entre las etiquetas y las predicciones del modelo.	82
A.12.	Ejemplo detección de puntos de cambio GRU Autoencoder	82
A.13.	Reconstrucciones GRU Autoencoder. Los puntos negros corresponden a las reconstrucciones que realiza el modelo GRU AE en base a la señal original. Los colores verde indican que es un estado sin anomalías, mientras que amarillo y rojo corresponden a anomalías leves y graves respectivamente.	82
A.14.	Ejemplo detección de anomalías LSTM Autoencoder. Se realiza comparación entre las etiquetas y las predicciones del modelo.	83
A.15.	Ejemplo detección de puntos de cambio LSTM Autoencoder	83
A.16.	Reconstrucciones LSTM Autoencoder. Los puntos negros corresponden a las reconstrucciones que realiza el modelo LSTM AE en base a la señal original. Los colores verde indican que es un estado sin anomalías, mientras que amarillo y rojo corresponden a anomalías leves y graves respectivamente.	83
B.1.	Gráficos Boxplot Bomba Booster R6	84
B.2.	Histograma Temperatura Cojinete Empuje Bomba Booster R6	84
B.3.	Histograma Temperatura Rodamiento Lado Carga Motor Bomba Booster R6	85
B.4.	Histograma Temperatura Rodamiento Lado Libre Motor Bomba Booster R6	85
B.5.	Vibración Cojinete Radial Bomba Booster R6	86
B.6.	Histograma Vibración Cojinete Empuje Bomba Booster R6	86
B.7.	Histograma Presión Entrada Bomba Booster R6	87
B.8.	Histograma Velocidad Eje Bomba Booster R6	87
B.9.	Histograma Caudal Rechazo Bomba Booster R6	88

Capítulo 1

Introducción

1.1. Antecedentes

En la industria actual el mantenimiento es un punto clave durante el desarrollo y la fabricación de un producto. Se deben minimizar tiempos y costes de fabricación, manteniendo los mejores estándares de calidad posibles. De este modo, paradas en las líneas de producción o gastos innecesarios en reparar maquinaria son inadmisibles. En la nueva era de las fábricas inteligentes, digitalizadas y conectadas, el papel del mantenimiento predictivo adquiere gran importancia. La posibilidad de poder disponer y procesar grandes cantidades de datos a tiempo real, gracias a tecnologías como el Internet de las cosas, IoT (Internet of Things), o el BigData, hace que sea muy atractivo de implementar para las diferentes industrias.

El mantenimiento predictivo se basa en el análisis de datos y la detección temprana de señales de deterioro en los equipos y maquinaria. Al monitorear constantemente los indicadores clave de rendimiento y recopilar información en tiempo real, es posible identificar patrones y tendencias que sugieran posibles fallas. Esto permite tomar medidas preventivas de manera proactiva, programar intervenciones de mantenimiento en momentos óptimos y evitar costosas paradas no planificadas en la producción. Gracias a lo anterior es que se ha convertido en una estrategia clave para garantizar la rentabilidad y la competitividad en el mercado actual.

Uno de los estudios más completos sobre el potencial del análisis industrial, como el mantenimiento predictivo, fue realizado por McKinsey en 2015 [1]. Descubriendo oportunidades de mejora en los siguientes aspectos:

- Reducción del 10 al 40% en los costos de mantenimiento: Dado que el mantenimiento planificado se basa en un cronograma, habrá casos en los que se realicen tareas de mantención cuando no sean necesarias. El mantenimiento predictivo puede evitar tales ineficiencias.
- Reducción del 10 al 20% en desperdicio: El funcionamiento no óptimo o anómalo que no se detecta puede resultar en una producción derrochadora. En tales casos, se desperdician materias primas, energía, costos de mano de obra y tiempo de máquina. Los sistemas de mantenimiento predictivo pueden descubrir problemas que pueden generar desperdicio antes de que surjan.

- Descubrimiento de nuevas oportunidades de mejora del 10 al 50%. Una vez que la recolección de datos se automatiza, a través del análisis avanzado se pueden descubrir diariamente nuevas perspectivas sobre oportunidades de optimización de procesos.

En la última década el monitoreo de equipos industriales ha despertado mucho interés [2]. Como consecuencia y potenciado por el progreso de la inteligencia artificial (IA), han surgido múltiples metodologías y aplicaciones de análisis predictivo para identificar anomalías en conjuntos de datos a gran escala y, entre otros, determinar cuándo es necesario un mantenimiento [3, 4]. La adopción de métodos de IA para mantenimiento predictivo tiene el potencial de prevenir fallas en los equipos, sin requerir una comprensión clara del proceso de producción o de los datos recopilados. La razón es que los enfoques basados en datos se pueden utilizar para entrenar modelos de aprendizaje automático con datos de funcionamiento normal, para luego testear con datos que presenten anomalías o fallas sin necesidad de conocer en profundidad el proceso subyacente.

Actualmente, los métodos de aprendizaje automático tradicionales no son capaces de procesar los grandes volúmenes de datos que se recopilan cada segundo gracias a las tecnologías de IoT, por lo que es necesario aplicar modelos basados en aprendizaje profundo (Deep Learning) [2]. Los modelos de diagnóstico basados en Deep Learning aprenden automáticamente las características de los datos de monitoreo de entrada y simultáneamente reconocen los estados de salud de las máquinas según las características aprendidas. En su mayoría incluyen las capas de extracción de características y las capas de clasificación, estas últimas determinan si el funcionamiento es normal o anómalo dependiendo de las etiquetas de los datos de entrenamiento. Sin embargo, uno de los problemas y desafíos que existen actualmente en la industria es la falta de data etiquetada. Esto motiva a encontrar soluciones basadas en aprendizaje no supervisado [5]. A diferencia del aprendizaje supervisado, donde se requiere un conjunto de datos etiquetado para entrenar al modelo, el aprendizaje no supervisado busca encontrar patrones y estructuras ocultas dentro de los datos sin la necesidad de etiquetas previas. En el aprendizaje profundo no supervisado, los algoritmos exploran y analizan grandes conjuntos de datos para identificar características significativas y representaciones útiles de la información. Esto permite que el modelo descubra información relevante sin una guía explícita, lo que lo hace especialmente valioso en situaciones donde los datos son escasos o costosos de etiquetar.

En la actualidad, los métodos de deep learning no supervisado han experimentado un rápido avance y se han convertido en una parte fundamental del campo de la inteligencia artificial. Estos métodos se enfocan en extraer patrones y características complejas directamente de los datos sin la necesidad de etiquetas o guía externa. Entre los enfoques más destacados se encuentran los autoencoders, que aprenden representaciones latentes de los datos; los GANs (Generative Adversarial Networks), que generan datos sintéticos realistas; y los VAEs (Variational Autoencoders), que permiten la generación de datos y el muestreo en el espacio latente. Estos métodos han demostrado su eficacia en diversas aplicaciones, como el procesamiento del lenguaje natural, la visión por computadora y la generación de contenido creativo. Su capacidad para extraer características relevantes y representaciones abstractas de los datos ha impulsado avances significativos en áreas como el análisis de datos complejos, la detección de anomalías y la generación de contenido multimedia. [6–9].

Uno de los principales desafíos en el uso del deep learning no supervisado en el manteni-

miento predictivo es la detección de diferentes modos de operación de los equipos [10]. En entornos industriales complejos, los equipos pueden operar en múltiples modos o condiciones, cada uno con sus propias características y patrones de comportamiento. Estos diferentes modos pueden estar asociados con distintos niveles de carga, velocidades de operación, temperaturas, entre otros factores. Para implementar un mantenimiento predictivo efectivo, es fundamental que los modelos de deep learning no supervisado sean capaces de identificar y distinguir estos diferentes modos de operación. Esto se vuelve especialmente relevante cuando los datos disponibles para el entrenamiento son altamente no lineales y presentan una gran variabilidad debido a cambios en las condiciones de operación.

Para abordar este desafío, se requiere una cuidadosa selección y diseño de arquitecturas de modelos, así como un procesamiento adecuado de la información para capturar las variaciones en los diferentes modos de operación. También es importante contar con conjuntos de datos representativos que cubran la mayor cantidad de escenarios posibles, lo que permitirá que los modelos aprendan y generalicen de manera efectiva. Además, la interpretación de los resultados obtenidos por los modelos de deep learning no supervisado también puede ser un reto, ya que las representaciones latentes aprendidas pueden ser altamente abstractas y difíciles de interpretar directamente. Por lo tanto, es necesario desarrollar técnicas de visualización e interpretación que permitan comprender y validar los resultados de manera adecuada.

En conclusión, aunque el deep learning no supervisado tiene un gran potencial para el mantenimiento predictivo, enfrenta el desafío de detectar diferentes modos de operación en equipos industriales. Superar esta dificultad permitirá mejorar la precisión y eficacia de los modelos, lo que a su vez contribuirá a una planificación más adecuada y oportuna de actividades de mantenimiento, reduciendo así los costos y tiempos de inactividad asociados con posibles fallos.

1.2. Objetivos

1.2.1. Objetivo General

El objetivo de este trabajo de tesis es desarrollar y aplicar técnicas de deep learning no supervisado en el ámbito del mantenimiento predictivo de equipos industriales con condiciones de operación variables. Se busca explorar el potencial de estos métodos para detectar patrones de fallas y anomalías en grandes volúmenes de datos no etiquetados, permitiendo una mejora significativa en la eficiencia y precisión del mantenimiento preventivo. Además, se pretende abordar los desafíos asociados a la falta de datos de falla etiquetados y las variaciones en las condiciones de operación de los equipos, con el objetivo de proporcionar una herramienta eficaz y confiable para el monitoreo y diagnóstico de condición de equipos industriales, optimizando así la producción y reduciendo costos y tiempos de inactividad.

1.2.2. Objetivos Específicos

- Estudiar e implementar diferentes arquitecturas de deep learning no supervisado, como autoencoders, variational autoencoders, y otros, para la detección de diferentes modos de operación y patrones de fallas en los datos industriales.

- Evaluar y comparar el rendimiento de los modelos de deep learning no supervisado implementados, para determinar la eficacia y ventajas de cada enfoque.
- Analizar los resultados obtenidos y proponer recomendaciones para la implementación práctica de las técnicas de deep learning no supervisado en un entorno industrial real, teniendo en cuenta las limitaciones y desafíos encontrados durante el desarrollo del estudio.
- Validar experimentalmente el enfoque propuesto en datos reales de equipos industriales, demostrando su capacidad para detectar de manera efectiva y confiable diferentes modos de operación y patrones de fallas, lo que permitirá mejorar la toma de decisiones en el mantenimiento preventivo y optimizar la eficiencia operativa en la industria.

Capítulo 2

Marco Teórico

En este capítulo se presentan los conceptos importantes que sustentan el trabajo realizado. En un comienzo se aclaran los conceptos relacionados al área del mantenimiento industrial. Luego, se explica qué es el aprendizaje profundo (Deep Learning), donde se presentan conceptos básicos de redes neuronales y su entrenamiento. Tras esto se presentan conceptos de detección de anomalías donde se define qué es una anomalía y se muestran diferentes algoritmos para su detección. Finalmente, se muestran diferentes aplicaciones de aprendizaje de máquinas en el dominio del mantenimiento predictivo, desde algoritmos clásicos hasta métodos de deep learning para la detección de anomalías y predicción de fallas.

2.1. Mantenimiento Industrial

El mantenimiento industrial es un conjunto de actividades destinadas a asegurar el correcto funcionamiento, disponibilidad y confiabilidad de los equipos, maquinarias y sistemas utilizados en las operaciones industriales. Su propósito es preservar los activos y recursos de una organización, garantizando que estos se mantengan en condiciones óptimas de operación y productividad a lo largo del tiempo.

Existen varios tipos de mantenimiento industrial [4], cada uno diseñado para abordar diferentes aspectos de la gestión y cuidado de los equipos. El mantenimiento correctivo se enfoca en intervenir cuando un componente ya ha fallado, realizando reparaciones o sustituciones para restaurar su funcionamiento. Por otro lado, el mantenimiento preventivo implica llevar a cabo tareas programadas de inspección, limpieza y ajuste para prevenir posibles fallas antes de que ocurran. Esta modalidad reduce el riesgo de interrupciones no planificadas, pero puede no ser eficiente en la detección de problemas incipientes.

Por último, el mantenimiento predictivo utiliza herramientas predictivas para determinar cuándo es necesario realizar acciones correctivas. Este tipo de mantenimiento se basa en el monitoreo continuo de los equipos y/o de los procesos productivos. Aprovechando los avances en la tecnología y el análisis de datos, se utiliza la recolección y el procesamiento de datos en tiempo real para predecir cuándo es probable que ocurran fallos o anomalías. Esto permite a las empresas tomar medidas proactivas y preventivas antes de que los problemas se conviertan en costosas interrupciones en la producción.

Al optar por una estrategia correctiva las industrias retrasan las acciones de mantenimiento y asumen el riesgo de la indisponibilidad de sus activos; por otro lado, la estrategia preventiva anticipa las intervenciones de mantenimiento, lo que resulta en un intercambio de piezas de repuesto con vida útil restante. Por lo tanto, una buena estrategia de mantenimiento debería mejorar la condición del equipo, reducir las tasas de falla del equipo y minimizar los costos de mantenimiento, al tiempo que maximiza la vida útil del equipo. Debido a este hecho, la estrategia predictiva es la que más se destaca [11], y está atrayendo la atención en la era de la Industria 4.0 debido a su capacidad para optimizar el uso y la gestión de activos.

Se pueden encontrar 3 categorías principales dentro del mantenimiento predictivo: enfoques estadísticos, enfoques basados en sistemas expertos y enfoques basados en inteligencia artificial. Dado que los enfoques basados en sistemas expertos requieren conocimientos mecánicos y teoría del equipo a monitorear, los enfoques estadísticos y con IA se han aplicado cada vez más. En [12] se comparan varios enfoques de inteligencia artificial con un enfoque estadístico para predecir cuándo un equipo estará en riesgo de falla en el futuro y los resultados sugieren que los enfoques de inteligencia artificial superan a los enfoques estadísticos.

Como se mencionó anteriormente los enfoques basados en inteligencia artificial presentan los mejores resultados en la literatura por lo que es necesario definir la teoría que existe detrás de estos enfoques para comprender de mejor forma su funcionamiento y su aplicabilidad en el contexto del mantenimiento predictivo.

2.1.1. Monitoreo Condición

Cualquier máquina, ya sea una máquina rotativa (bomba, compresor, turbina de gas o vapor, etc.) o una máquina no giratoria (intercambiador de calor, columna de destilación, válvula, etc.) eventualmente alcanzará un punto de mala salud. Es posible que ese punto no sea el de una falla o apagado real, sino uno en el que el equipo ya no actúa en su estado óptimo. Esto indica que podría ser necesario realizar alguna actividad de mantenimiento para restablecer todo el potencial operativo. En términos simples, identificar el “estado de salud” de nuestro equipo es el dominio del monitoreo condición.

La forma más común de realizar el monitoreo de condición es observar cada medición de sensor de la máquina e imponer un límite de valor mínimo y máximo en él (sistema de alarma por umbrales). Si el valor actual está dentro de los límites, entonces la máquina está en buen estado. Si el valor actual está fuera de los límites, entonces la máquina no es saludable y se envía una alarma.

Se sabe que este procedimiento de sistema de alertas por umbrales envía una gran cantidad de falsas alarmas, es decir, alarmas para situaciones que realmente son estados saludables para la máquina. También faltan alarmas, es decir, situaciones que son problemáticas pero que no están alarmadas. El primer problema no solo desperdicia tiempo y esfuerzo, sino también la disponibilidad del equipo. El segundo problema es más crucial, ya que conduce a un daño real con el costo de reparación asociado y la pérdida de producción.

Ambos problemas se deben a la misma causa: la salud de un equipo complejo no se puede juzgar de manera confiable según el análisis de cada sensor por separado. Se debe más

bien considerar una combinación de las diversas mediciones para obtener una indicación real del estado de salud del equipo. En este contexto, el aprendizaje profundo se presenta como una solución prometedora, ya que permiten aprender automáticamente patrones y relaciones complejas entre múltiples variables de entrada.

2.2. Aprendizaje Profundo

El presente trabajo se enfoca en la aplicación de algoritmos en el estado del arte de aprendizaje de máquinas, los cuales se basan en técnicas de aprendizaje profundo (Deep Learning) [13]. No existe un consenso en la literatura de qué significa aprendizaje profundo, pero es común referirse a él como el entrenamiento de redes neuronales con múltiples capas interconectadas a través de unidades de procesamiento (neuronas artificiales o perceptrón) [14]. A continuación, se presentan los fundamentos de las redes neuronales artificiales, del Deep Learning y sus aplicaciones en la clasificación y detección de anomalías.

2.2.1. Redes Neuronales Artificiales

Las redes neuronales artificiales (ANN) son modelos matemáticos comúnmente utilizados para la resolución de problemas de clasificación o regresión. La unidad básica de procesamiento de una ANN recibe el nombre de neurona o perceptrón [14], esta es una función matemática que recibe un vector de datos x como entrada, y luego aplica una transformación lineal sobre x , seguida por una función no lineal, la que recibe el nombre de función de activación. Cuando se agrupan neuronas se conforma una capa, las que reciben el nombre de capas totalmente conectadas o, en inglés, fully connected layers. La salida producida por un conjunto de m neuronas de una capa fully connected se calcula como:

$$y = \varphi(W \cdot x + b) \quad (2.1)$$

donde $x \in R^n$ es la entrada de la capa, $y \in R^m$ es la salida de una capa de m neuronas, $W \in R^{(m \times n)}$ es una matriz de parámetros denominados pesos, $b \in R^m$ es un vector que contiene los llamados sesgos (biases en inglés) de la capa, y $\phi(\cdot)$ es la función de activación no lineal que se aplica tras la transformación lineal de x . Existe una gran variedad de funciones de activación que se pueden escoger para $\phi(\cdot)$, las más utilizadas son las funciones sigmoide, tangente hiperbólica (tanh) y la unidad lineal rectificadora (ReLU) [15]. Sus ecuaciones son mostradas a continuación:

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (2.2)$$

$$\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.3)$$

$$\text{ReLU}(x) = \max\{0, x\} \quad (2.4)$$

Los parámetros W y b se ajustan durante el proceso de entrenamiento, en donde la red

neuronal se expone a un conjunto de datos de entrada junto con sus salidas esperadas. A medida que la red procesa los datos, se calcula la diferencia entre las salidas reales y las salidas esperadas, y esta diferencia se utiliza para ajustar cada uno de los parámetros de las conexiones.

Cuando las redes neuronales son utilizadas para clasificación, normalmente se utiliza en la última capa del modelo una función de activación que traduzca la salida en probabilidades de clasificación. La función sigmoid(x) cumple esta propiedad, ya que produce salidas acotadas entre (0, 1), las que pueden ser interpretadas como la probabilidad de activación para una neurona en la presencia o ausencia de una clase, una propiedad también útil para la clasificación binaria. Una generalización de la función sigmoide, útil para modelos de clasificación multiclase, es la función de activación softmax, la que usualmente recibe el nombre de capa de salida softmax.

Para un problema de clasificación donde se desea encontrar la clase correcta entre un total de K clases, la función softmax comprende K neuronas x_i , $i \in 1, \dots, K$, donde se desea asignar una probabilidad a cada una, por lo tanto, se requiere que el valor de salida de todas las K neuronas sume 1. Esto se logra al definir la función softmax como:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}}, \quad \text{para } i \in \{1, \dots, K\} \quad (2.5)$$

Las ANNs compuestas exclusivamente de capas fully connected se encuentran limitadas a entradas en formato de vectores, además de no tener en consideración la presencia de correlación entre características adyacentes. Para sobreponerse a esta limitación, y preservar el grado de correlación espacial o temporal que pueden llegar a tener datos como imágenes o series de tiempo existen arquitecturas específicas que abordan estos requerimientos. Las redes neuronales convolucionales y las redes neuronales recurrentes.

2.2.2. Redes neuronales convolucionales

Las redes neuronales convolucionales (CNN) son capaces de capturar tanto relaciones espaciales como temporales que puedan existir en los datos de entrada. El componente principal de las CNN's son las capas convolucionales, las cuales mediante una operación de convolución aplican un filtro o kernel a la entrada de la capa. De forma similar a una capa fully connected, las salidas de las capas convolucionales son calculadas como sigue:

$$y = \varphi(W * x + b) \quad (2.6)$$

donde x corresponde a la entrada de la capa, y es la salida de la capa, W es el conjunto de filtros a aplicar por convolución sobre la entrada, b es el vector de sesgos, que posee una componente para cada filtro, y $\phi(\cdot)$ es la función de activación. En este caso, la operación \star entre x y W es una convolución. Si se considera el caso donde las entradas x son imágenes, se tiene que $x \in R^{(e \times f \times g)}$, $y \in R^{(u \times v \times l)}$ son tensores de 3 dimensiones, mientras $W \in R^{(u \times v \times l)}$ y $b \in R^l$. El cálculo de cada elemento $y_{(i,j,k)}$ de $y \in R^{(u \times v \times l)}$ se deriva desde la operación de convolución como sigue:

$$y_{i,j,k} = \sum_{m,n,p} (x_{i-m,j-n,p} \cdot W_{m,n,p,k} + b_k) \quad (2.7)$$

donde cada elemento i, j, k del tensor y se calcula al desplazar los filtros de W sobre el tensor x y aplicar la ecuación 2.7. Cada vez que W se mueve sobre las 2 primeras dimensiones de x , este se salta S píxeles, S recibe el nombre de stride. Tras aplicar las capas convolucionales, las primeras dos dimensiones de y son más pequeñas que las de x . Las dimensiones espaciales (primera y segunda dimensión) del tensor x e y , se relacionan entre sí como:

$$U = \frac{E - D}{S} + 1 \quad (2.8)$$

donde U es el tamaño de cualquiera de las dimensiones espaciales de y , E es el tamaño de la respectiva dimensión espacial x , D es la respectiva dimensión espacial de W y S es el stride utilizado en la operación de convolución.

Cuando se trabaja en el dominio de las imágenes, usualmente se utilizan filtros cuadrados, con lo que $d = 0$ en las dimensiones de $W \in R^{(d \times o \times t \times l)}$, y se habla de que la capa tiene l filtros de tamaño d . En caso de utilizarse dominios temporales, donde las entradas son series de tiempo, las relaciones anteriores resultan equivalentes si se considera la primera dimensión espacial de las variables equivalente a la dimensión temporal, y la segunda dimensión espacial se reemplaza por un valor 1, i.e. $f = 0 = v = 1$.

Existen muchos otros tipos de capas que componen a las CNNs, incluso, día a día se crean nuevos componentes para estos modelos, ya que son sistemas en el estado del arte del aprendizaje de máquinas. A continuación, se muestran algunas de las capas más comunes dentro de las arquitecturas de CNNs, las que también componen los modelos utilizados en este trabajo de tesis:

- **Zero-padding:** Es una técnica utilizada para preservar las dimensiones espaciales o temporales de la entrada $x \in R^{(e \times f \times g)}$ a la salida $y \in R^{(u \times v \times l)}$ de una capa convolucional. El zero-padding consiste en agregar 0's a los bordes de las dimensiones espaciales o temporales de la entrada x de una capa convolucional.
- **Capas de pooling:** Las capas de pooling se utilizan en CNNs para reducir la dimensión espacial de sus entradas. Las más utilizadas son el pooling por máximo (max pooling en inglés) y pooling por promedio (average pooling en inglés). Max pooling retorna el valor máximo dentro de una ventana de su entrada x , de forma similar a un filtro convolucional, esta ventana de extracción de valor máximo recorre las dimensiones espaciales (o temporales) de la entrada. La capa de average pooling opera de modo similar a max pooling, pero se extrae el promedio de los valores en la ventana. Un caso particular de average pooling es el global average pooling, donde la ventana utilizada cubre todo el rango de la entrada y la colapsa a un único valor en las dimensiones espaciales (o temporales).
- **Capa de batch normalization:** El batch normalization [15] funciona como una capa de normalización entrenable que posee comportamientos diferentes durante el entrenamiento y la evaluación del modelo. Durante el entrenamiento, para cada batch (grupo de instancias de tu conjunto de datos) de entrenamiento, esta capa calcula el promedio

y varianza de cada característica, para normalizarlas y computar una media móvil exponencial del promedio y la varianza del conjunto de entrenamiento. Una vez finalizado el entrenamiento del modelo, para su evaluación, la población completa de estadísticas ajustadas durante el entrenamiento se utiliza para normalizar las entradas. Batch normalization no sólo normaliza los valores de entrada para que tengan un valor promedio cercano a 0 o una desviación estándar cercana a 1, también comprende una ponderación lineal de estas entradas, lo que permite su escalamiento y desplazamiento. Esta capa permite al modelo enfatizar o ignorar características específicas de la entrada, actuando como un regularizador que acelera el proceso de entrenamiento.

- **Dropout:** Es una operación usualmente aplicada a la salida de las capas fully connected, aunque recientemente se está usando sobre capas convolucionales para eliminar filtros completos. Dropout se utiliza como un regularizador del modelo para evitar el sobreajuste de capas con gran cantidad de neuronas. Similar a la capa de batch normalization, dropout realiza operaciones diferentes durante entrenamiento y evaluación. La operación de dropout se caracteriza por la tasa de dropout (dropout rate en inglés) $DR \in [0, 1]$, parámetro que, en la fase de entrenamiento del modelo, define la probabilidad de reemplazar cada uno de sus entradas por 0, y multiplicar los valores no reemplazados por 0 por $1/(1 - DR)$, de tal forma que la suma sobre todos los valores de entrada se mantenga en el mismo orden de magnitud. En cada paso de entrenamiento un porcentaje DR de las salidas de la capa con dropout no serán utilizadas, reduciendo el tamaño efectivo de la capa. Por otro lado, cuando se usa el modelo para evaluar, se desactiva el dropout. El efecto deseado de aplicar dropout es forzar al modelo a no depender de neuronas específicas en cada capa.

Las CNN están específicamente diseñadas para manejar una variedad de formas en 2D y, por lo tanto, se emplean ampliamente en el reconocimiento visual, el análisis de imágenes médicas, la segmentación de imágenes, el procesamiento del lenguaje natural y muchas otras áreas [16–21]. La capacidad de descubrir automáticamente características esenciales a partir de la entrada sin necesidad de intervención humana la hace más poderosa que una red tradicional. Existen varias variantes de CNN en este campo, que incluyen el grupo de geometría visual (VGG) [19][19], AlexNet [17], Xception [18], Inception [22], ResNet [20], entre otros, que pueden ser utilizadas en diversos dominios de aplicación según sus capacidades de aprendizaje.

2.2.3. Redes Neuronales Recurrentes

Las redes neuronales recurrentes (RNN) son una arquitectura especializada en el procesamiento de datos secuenciales, como series de tiempo, texto o cualquier otro tipo de secuencia [23]. La característica distintiva de las RNN es su capacidad para mantener estados internos o "memoria" mientras procesan cada elemento de una secuencia. Esto les permite aprender dependencias a largo plazo y capturar la estructura temporal de los datos. En una RNN, cada unidad de procesamiento o neurona se conecta a sí misma, permitiendo la retroalimentación de la salida anterior con la entrada actual. Esta propiedad es esencial para capturar relaciones de dependencia temporal y modelar la evolución en el tiempo.

El componente clave en las RNN es la célula recurrente, que se encarga de mantener y actualizar la memoria de la red a medida que procesa los elementos de la secuencia. Sin em-

bargo, las RNN tradicionales tienen limitaciones en su capacidad para capturar dependencias a largo plazo, lo que llevó al desarrollo de variantes más avanzadas, como las Long Short-Term Memory (LSTM) y las Gated Recurrent Units (GRU). Estas variantes abordan el problema del desvanecimiento de gradientes y permiten un mejor aprendizaje de patrones a largo plazo.

La ecuación general de una célula recurrente se expresa como:

$$H^{(t)} = \varphi \left(W_{hd}X^{(t)} + W_{hh}H^{(t-1)} + B \right) \quad (2.9)$$

donde:

- $H^{(t)}$ es el estado oculto de la célula en el momento t .
- $X^{(t)}$ es la entrada en el momento t .
- $H^{(t-1)}$ es la memoria (o el estado oculto anterior) en el momento $t - 1$.
- W_{hd} y W_{hh} son matrices de pesos.
- B es el vector de sesgos.
- $\varphi(\cdot)$ es la función de activación.

De la relación entre las variables ocultas $H^{(t)}$ y $H^{(t-1)}$, se desprende que estas variables capturan y retienen la información histórica de la secuencia hasta su paso de tiempo actual. Dado que el estado oculto usa la misma definición del paso de tiempo anterior en el paso de tiempo actual, el cálculo de $H^{(t)}$ es recurrente. Por esta razón, las redes neuronales con estados ocultos basados en cálculos recurrentes se denominan redes neuronales recurrentes.

Para el paso de tiempo t , la salida de la capa de salida de la RNN se calcula de manera similar a una red fully connected:

$$Z_t = W_{qh}H^{(t)} + b_q \quad (2.10)$$

donde los parámetros de la RNN incluyen los pesos W_{hd} , W_{hh} y B de la capa oculta, junto con los pesos W_{qh} y el sesgo b_q de la capa de salida.

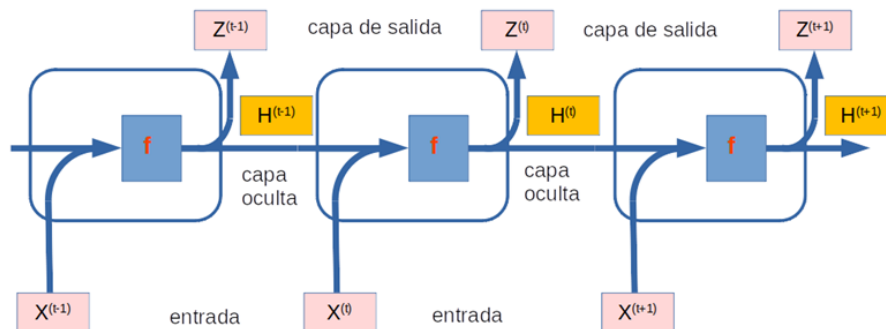


Figura 2.1: La siguiente figura ilustra la lógica computacional de una RNN en 3 pasos de tiempo adyacentes

Como ya se mencionó la RNN estándar sufren del problema del desvanecimiento del gradiente. Este problema surge durante la fase de entrenamiento, en donde las redes neuronales usan el algoritmo de descenso del gradiente (ver sección 2.2.4), que va calculando las derivadas de los parámetros de la red de forma iterativa. Por esto, es que cuando tenemos una red con muchas capas el valor de gradiente cada vez es más cercano a 0 ya que estamos multiplicando muchas veces un valor pequeño.

El problema es similar en las redes neuronales recurrentes. Las RNN modelan datos en los que es importante la estructura temporal, como frases con palabras, y también se entrenan usando backpropagation. Cada intervalo de tiempo sería como una capa y por lo tanto una RNN sería equivalente a una red con tantas capas como intervalos de tiempo y sufrirían el mismo problema de desvanecimiento del gradiente. Existen varias soluciones para este problema, en donde algunas de las arquitecturas más populares son:

- **Long short-term memory (LSTM):** Es una arquitectura RNN muy popular introducida por [24] que utiliza unidades de procesamiento especiales para abordar el problema del desvanecimiento del gradiente. Una celda de memoria en una unidad LSTM puede almacenar datos durante largos períodos y el flujo de información hacia dentro y fuera de la celda es gestionado por tres compuertas. La 'Compuerta de Olvido' determina qué información del estado anterior de la celda será memorizada y qué información se eliminará porque ya no es útil, mientras que la 'Compuerta de Entrada' determina qué información debe ingresar al estado de la celda y la 'Compuerta de Salida' determina y controla las salidas. Dado que resuelve los problemas de entrenar una red recurrente, la red LSTM se considera una de las RNN más exitosas.
- **Bidirectional LSTM (BiLSTM):** Esta arquitectura conecta dos capas ocultas que funcionan en direcciones opuestas hacia una única salida, lo que les permite aceptar datos tanto del pasado como del futuro. A diferencia de las redes recurrentes tradicionales, las RNN bidireccionales están entrenadas para predecir tanto direcciones temporales positivas como negativas al mismo tiempo. Un LSTM bidireccional, comúnmente conocido como BiLSTM, es una extensión del LSTM estándar que puede mejorar el rendimiento del modelo en problemas de clasificación de secuencias [25]. Es un modelo de procesamiento de secuencias que consta de dos LSTMs: uno toma la entrada en sentido directo y el otro la toma en sentido inverso. El LSTM bidireccional en particular es una elección popular en tareas de procesamiento de lenguaje natural.
- **Gates recurrent units (GRU's):** Una Unidad Recurrente con Compuertas (GRU) es otra variante popular que utiliza métodos de compuertas para controlar y gestionar el flujo de información entre las celdas en la red neuronal, introducida por [26]. El GRU es similar a un LSTM, sin embargo, tiene menos parámetros, ya que tiene una compuerta de reinicio y una compuerta de actualización, pero carece de la compuerta de salida. Por lo tanto, la diferencia clave entre un GRU y un LSTM es que un GRU tiene dos compuertas (compuertas de reinicio y actualización), mientras que un LSTM tiene tres compuertas (compuertas de entrada, salida y olvido). La estructura del GRU le permite capturar dependencias de secuencias largas de datos de manera adaptativa, sin descartar información de partes anteriores de la secuencia. Por lo tanto, el GRU es una variante ligeramente más simplificada que a menudo ofrece un rendimiento comparable y es significativamente más rápido de calcular [27]. Aunque se ha demostrado que

los GRU's tienen un mejor rendimiento en ciertos conjuntos de datos más pequeños y menos frecuentes [27], ambas variantes de RNN han demostrado su eficacia al producir resultados.

2.2.4. Entrenamientos de Redes Neuronales

El objetivo de utilizar una ANN f_θ de parámetros $\theta \in \Theta$, es aproximar una función $y = f(x)$, con $x \in X$. En la práctica no se tiene acceso a la distribución completa de los datos de entrada X , sino que a un subconjunto de N muestras de datos $\{(x^{(i)}, y^{(i)})\}_{i=1}^N$ de la función a aproximar, el que recibe el nombre de conjunto de entrenamiento. Encontrar el mejor conjunto de parámetros θ^* para la ANN $f_\theta(x)$ requiere resolver el siguiente problema de optimización:

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}} C(\theta) = \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^N L(y^{(i)}, f_\theta(x^{(i)})) \quad (11) \quad (2.11)$$

donde C es un funcional de error definido por la función L que recibe el nombre de función de pérdida o *loss function*, en inglés. La optimización mostrada en la ecuación (11), se logra a través de técnicas de optimización basadas en gradiente descendente, cuando se escoge L como una función diferenciable (e.g. entropía cruzada). Los parámetros θ son iterativamente ajustados por la siguiente regla, hasta la convergencia:

$$\theta_k = \theta_{k-1} - \mu \nabla_\theta C(\theta) \quad (11) \quad (2.12)$$

Debido a que las ANNs se componen de múltiples capas consecutivas, el cálculo directo de $\nabla_\theta C(\theta)$ requiere extensivos recursos computacionales. Sin embargo, los gradientes pueden ser eficientemente calculados utilizando la técnica de retro-propagación (*back-propagation*), algoritmo que propaga el error a través del modelo, desde su salida hasta que alcanza la primera capa de la ANN. *Back-propagation* se basa en la regla de la cadena para obtener la derivada de funciones.

Incluso cuando se utiliza *back-propagation*, para las redes neuronales entrenadas en grandes cantidades de datos, el cálculo del gradiente exacto $\nabla_\theta C(\theta)$ se torna computacionalmente costoso. Como solución a este problema, se utiliza una estimación no sesgada del gradiente $\nabla_\theta \tilde{C}(\theta)$, donde el gradiente es calculado sobre una pequeña porción aleatoria de los datos, porción que recibe el nombre de *batch* y la cantidad de muestras de los datos en un *batch* recibe el nombre de *batch size* (BS). De esta forma, la regla de optimización para un *batch* $B \subset X$ es:

$$\theta_k = \theta_{k-1} - \mu \frac{1}{|B|} \sum_{i \in B} L(y^{(i)}, f_\theta(x^{(i)})) \quad (13) \quad (2.13)$$

donde μ es una constante llamada tasa de aprendizaje o *learning rate*, en inglés, y establece cuán grande es el paso de entrenamiento realizado. Esta técnica de entrenar por *batches*, es una forma de gradiente descendente estocástico (SGD por sus siglas en inglés).

Una alternativa ampliamente utilizada a la regla de optimización de la ecuación (13), es Adam [28], un algoritmo de optimización con tasa de aprendizaje adaptativa que ajusta μ_k de

forma automática. Adam utiliza los gradientes elevados al cuadrado para escalar la tasa de aprendizaje e incluye el promedio móvil de los gradientes en su formulación, estrategia que es conocida como *momentum*, y es utilizada para evitar la convergencia a un mínimo local durante la optimización. Los principales hiperparámetros de Adam son β_1 y β_2 , los que se relacionan al promedio móvil de los gradientes y sus cuadrados, respectivamente, y regulan la tasa a la que se ajusta la tasa de aprendizaje μ_k .

2.3. Aprendizaje Profundo en Mantenimiento Predictivo

El mantenimiento predictivo representa un salto cualitativo en la gestión de activos industriales al permitir intervenciones basadas en el estado real del equipo, anticipando fallos antes de que estos ocurran. La integración del aprendizaje profundo en el mantenimiento predictivo ha mostrado en la literatura una capacidad significativa para mejorar la precisión y la eficiencia de las predicciones en tiempo real. A continuación, se explicará en mayor detalle los conceptos básicos que permiten el uso de Deep Learning en el mantenimiento predictivo.

2.3.1. Detección de Anomalías

La detección de anomalías es un campo dentro del aprendizaje automático en donde se identifican elementos raros, eventos u observaciones que generan sospechas al diferenciarse significativamente de la mayoría de los datos. Normalmente, los datos anómalos se pueden relacionar a algún tipo de problema o evento raro como, por ejemplo, fraude bancario, problemas médicos, defectos estructurales, equipo defectuoso, etc. Esta conexión hace que sea muy interesante poder elegir qué puntos de datos pueden considerarse como anómalos, ya que identificar estos eventos suele ser muy útil desde una perspectiva práctica en la industria.

Un aspecto importante para cualquier técnica de detección de anomalías es la manera a través de la cual las anomalías son reportadas (*outputs*). Típicamente los *outputs* pueden ser de dos tipos:

- Puntaje o *score*: La mayoría de los modelos de aprendizaje de máquinas para detección de anomalías existentes buscan aprender un puntaje de normalidad (*score*), que permita mapear el espacio de todas las posibles entradas a un valor escalar. De tal forma que mientras más parecida a la distribución normal de datos sea la muestra de entrada, más alto será el puntaje de normalidad.
- Etiquetas o *labels*: Las técnicas en esta categoría asignan una etiqueta (normal o anómalo) a cada instancia de los datos de test.

Las técnicas de detección de anomalías basadas en puntuación permiten al analista utilizar un umbral específico del dominio del problema para seleccionar las anomalías más relevantes. Las técnicas que proporcionan etiquetas binarias a las instancias de prueba no permiten directamente a los analistas tomar esa elección, aunque esto puede ser controlado indirectamente a través de las elecciones de parámetros dentro de cada técnica. Los modelos que aprenden

un puntaje de normalidad son normalmente evaluados en términos de AUROC (*Area Under the Receiver Operating Characteristic*) o AUPR (*Area Under the Precision Recall*).

Como se muestra en [29], existen muchos modelos con diferentes estrategias para lograr obtener clasificaciones o un puntaje de normalidad. Según la estrategia utilizada en el aprendizaje los modelos pueden dividirse en aquellos basados en:

- **Aprendizaje supervisado:** Las soluciones basadas en aprendizaje supervisado asumen que se cuenta con un conjunto de datos de entrenamiento etiquetado, tanto para puntos normales como anómalos. Un enfoque típico en este tipo de casos es la construcción de modelos que puedan clasificar cada dato según las etiquetas normal o anómalo. Existen dos problemas principales que surgen en la detección de anomalías supervisada. En primer lugar, las instancias anómalas son considerablemente menos numerosas en comparación con las instancias normales. Los problemas derivados de este desbalance de clases han sido abordados múltiples veces en la literatura [30, 31]. En segundo lugar, obtener etiquetas precisas y representativas, especialmente para la clase de anomalías, suele ser un desafío. Algunas soluciones en la literatura [32] proponen diversas técnicas que introducen anomalías artificiales en un conjunto de datos normales para obtener un conjunto de datos de entrenamiento etiquetado.
- **Aprendizaje semisupervisado:** En este caso los modelos de detección son entrenados con conjuntos de entrenamiento compuestos únicamente por data normal, es decir, se asume que los datos anómalos son desconocidos para los modelos. Este es el enfoque más explorado en la literatura [33–35] y es conocido como detección de anomalías por una clase (*one-class anomaly detection*)
- **Aprendizaje no supervisado:** En este caso, se tiene un conjunto de entrenamiento donde anomalías y datos normales están mezclados y se busca separarlos sin utilizar información de sus etiquetas [36]. Esta es el área que presenta los mayores desafíos, especialmente en casos donde la cantidad de anomalías en los datos es escasa. Las técnicas en esta categoría asumen que las instancias normales son mucho más frecuentes que las anomalías en la data de prueba, en caso de no cumplirse esta suposición estos enfoques suelen sufrir de un alto número de falsos positivos.

2.3.2. Tipos de anomalías

El campo de la detección de anomalías se divide en tres tipos principales, que reflejan las diferentes maneras en que las anomalías pueden presentarse en datos de equipos industriales:

1. **Anomalías basadas en un punto:** Este enfoque identifica anomalías en observaciones individuales, aislando aquellos puntos de datos que difieren significativamente de la mayoría. Las técnicas de aprendizaje profundo aplicables incluyen redes neuronales densas y autoencoders, que modelan la distribución de los datos “normales” y son eficientes en resaltar puntos de datos que no se ajustan a esta distribución. *Ejemplo:* Un sensor de presión que registra un valor significativamente fuera del rango normal, a pesar de que las condiciones operativas son estándar.

2. **Anomalías contextuales:** Este tipo considera el contexto en el que se recogen los datos, reconociendo que lo que es normal en una situación puede no serlo en otra. Las redes neuronales recurrentes son particularmente útiles aquí, ya que tienen en cuenta la secuencia y el tiempo, permitiendo detectar anomalías que solo se manifiestan en ciertos contextos operacionales o temporales. *Ejemplo:* Una vibración que es normal a altas velocidades puede ser anómala a velocidades más bajas, y viceversa.
3. **Anomalías Colectivas:** Se centra en anomalías que surgen de la relación entre varios puntos de datos. En otras palabras, aunque cada punto de dato individual parece normal, su colectivo puede ser anómalo. Aquí, técnicas como las redes neuronales convolucionales (CNN) y autoencoders secuenciales pueden identificar patrones atípicos en secuencias de datos o en conjuntos de características interrelacionadas. *Ejemplo:* La falla inminente de un rodamiento puede no ser detectable en una sola medida de vibración, pero un análisis de las mediciones colectivas a lo largo del tiempo podría revelar patrones inusuales.

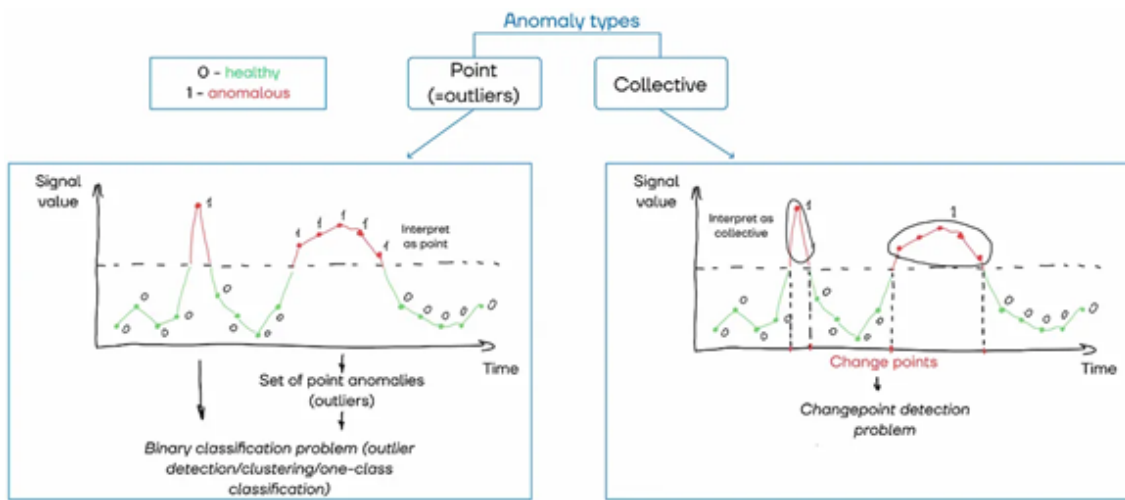


Figura 2.2: Anomalías individuales v/s colectivas. Fuente.

Cada uno de estos enfoques puede desplegarse mediante algoritmos de aprendizaje profundo, los cuales son elegidos en función de la naturaleza de los datos y el tipo de anomalía que se espera encontrar. La selección adecuada y la sintonización de estos modelos son cruciales para capturar las sutilezas de los datos y proporcionar alertas tempranas y fiables sobre condiciones anómalas que podrían preluir fallos en equipos.

2.3.3. Métodos de Deep Learning para la Detección de Anomalías

En el dominio específico de la detección de anomalías, el *Deep Learning* ofrece una gran variedad de arquitecturas y algoritmos que pueden ser entrenados para identificar estas anomalías [35, 37]. Los autoencoders, se destacan entre las demás arquitecturas gracias a la codificación eficiente de patrones normales y la identificación de desviaciones a través de errores de reconstrucción. Más allá de los autoencoders, las Redes Neuronales Convolucionales (CNN) se aplican cuando la correlación espacial entre los datos es crítica, mientras que las Redes Neuronales Recurrentes (RNN) y sus variantes, como las Long Short-Term Memory networks (LSTM) y las Gated Recurrent Units (GRU), sobresalen en el procesamiento y análisis de secuencias temporales. Estas arquitecturas permiten no solo captar las irregularidades

puntuales, sino también identificar patrones anómalos que se desarrollan a lo largo del tiempo, los cuales son cruciales para predecir fallos potenciales en equipos y componentes industriales.

La sinergia entre distintas arquitecturas de redes neuronales profundas se ha convertido en un enfoque prometedor para capturar la complejidad de los patrones de datos en múltiples dimensiones. Combinando la capacidad de los autoencoders para identificar la reconstrucción de datos normales con la potencia secuencial de las Redes Neuronales Recurrentes, se pueden construir modelos híbridos que aprovechen las fortalezas de ambos enfoques. Por ejemplo, los autoencoders secuenciales, que integran una estructura recurrente en la codificación y decodificación, permiten detectar anomalías no solo en datos individuales sino también en comportamientos a lo largo del tiempo, haciendo que el modelo sea excepcionalmente sensible a las secuencias anómalas que indican una posible degradación del equipo.

Por otra parte, la fusión de Redes Neuronales Convolucionales con autoencoders, conocida como Convolutional Autoencoders, permite analizar datos que poseen una estructura espacial inherente, como imágenes termográficas o patrones de vibración, aprovechando la capacidad de las CNN para extraer características visuales y espaciales significativas que luego son codificadas en una representación más densa por la parte del autoencoder. Esta combinación es particularmente útil para identificar fallas en componentes que alteran su forma visual o patrones vibratorios. Estos modelos híbridos, al integrar múltiples modalidades de datos y temporalidades, representan el vanguardismo en la detección de anomalías.

2.3.3.1. Autoencoders

Los autoencoders representan una clase especial de Redes Neuronales diseñadas para la tarea de reconstrucción de datos. A diferencia de las redes neuronales tradicionales, que se enfocan en clasificar datos complejos como imágenes, sonidos o videos en categorías predefinidas, los autoencoders se entrenan para reproducir la entrada en la salida. Este proceso de aprendizaje no supervisado no requiere etiquetas para categorizar los datos, sino que optimiza la red para generar una copia lo más fiel posible del dato de entrada. Esta peculiaridad los convierte en herramientas poderosas para la extracción de características, reduciendo la dimensionalidad de los datos y obteniendo una representación más manejable y esencial, conocida como espacio latente. Un autoencoder se compone de tres componentes clave:

1. **Encoder:** La función del encoder es comprimir los datos de entrada en una representación de menor dimensión, lo que se puede lograr mediante algún tipo de red neuronal.
2. **Bottleneck:** Este es el punto de la red donde la representación es la más compacta. Se le conoce como espacio latente y encapsula las características fundamentales de los datos, siendo el resultado de un proceso de aprendizaje donde el modelo identifica y preserva solo la información más relevante.
3. **Decoder:** A partir del espacio latente, el decoder se encarga de reconstruir la entrada original, tratando de minimizar las diferencias entre la entrada real y su reconstrucción.

Durante el entrenamiento, el autoencoder aprende ajustando sus pesos y sesgos para minimizar una función de pérdida que mide la discrepancia entre los datos de entrada y su

reconstrucción. El resultado es una red capaz de detectar la presencia de ruido y otras irregularidades, diferenciando las entradas normales de las anómalas. La utilidad de este enfoque radica en su habilidad para identificar datos que no se ajustan al modelo de “normalidad” que la red ha aprendido, lo que lo convierte en un mecanismo eficaz para la detección temprana de anomalías en contextos donde los datos se pueden reconstruir con precisión, como en la monitorización de condiciones de maquinaria y equipos industriales.

2.3.3.2. AutoRNN

Los Autoencoders Recurrentes, también conocidos como AutoRNN, combinan la capacidad de los autoencoders con la potencia de las Redes Neuronales Recurrentes (RNN) para manejar secuencias y datos temporales. Esta fusión resulta en una arquitectura capaz de aprender patrones temporales y dependencias a largo plazo dentro de las secuencias de datos. En estas arquitecturas tanto el encoder como el decoder están construidas a partir de unidades recurrentes donde las que más se destacan son las redes LSTM y GRU.

LSTM significa *Long Short-Term Memory*, y consiste en una serie de celdas de memoria que pueden almacenar y actualizar información durante largos intervalos de tiempo. Cada celda de memoria tiene tres puertas (Figura 2.3): una puerta de entrada, una puerta de salida y una puerta de olvido. La puerta de entrada decide qué información añadir al estado de la celda, la puerta de salida decide qué información emitir desde el estado de la celda y la puerta de olvido decide qué información descartar del estado de la celda. Las puertas son aprendidas por la red basada en la entrada y el estado oculto previo.

GRU significa unidad recurrente con compuertas, y es una versión simplificada de LSTM. Tiene solo dos compuertas (Figura 2.4): una compuerta de reinicio y una compuerta de actualización. La compuerta de reinicio decide cuánto del estado oculto anterior conservar, y la compuerta de actualización decide cuánto de la nueva entrada incorporar al estado oculto. El estado oculto también actúa como el estado de la celda y la salida, por lo que no hay una compuerta de salida separada. El GRU es más fácil de implementar y requiere menos parámetros que el LSTM.

El rendimiento de LSTM y GRU depende de la tarea, los datos y los hiperparámetros. En general, LSTM es más poderoso y flexible que GRU, pero también es más complejo y propenso al sobreajuste. GRU es más rápido y eficiente que LSTM, pero puede que no capture las dependencias a largo plazo tan bien como LSTM. De igual forma que con redes recurrentes, los autoencoders se pueden mezclar con redes convolucionales. Estas arquitecturas se conocen como AutoCNN.

2.3.3.3. Variational Autoencoders

Los Variational Autoencoders (VAEs) constituyen una ampliación de los autoencoders tradicionales, incorporando principios de la inferencia variacional en su arquitectura para modelar la distribución de los datos. A diferencia de los autoencoders estándar que generan una representación latente fija, los VAEs introducen un enfoque probabilístico que asigna a cada punto de entrada una distribución de probabilidades en el espacio latente. Esto significa que, en lugar de codificar una entrada como un único punto, el VAE aprende los parámetros

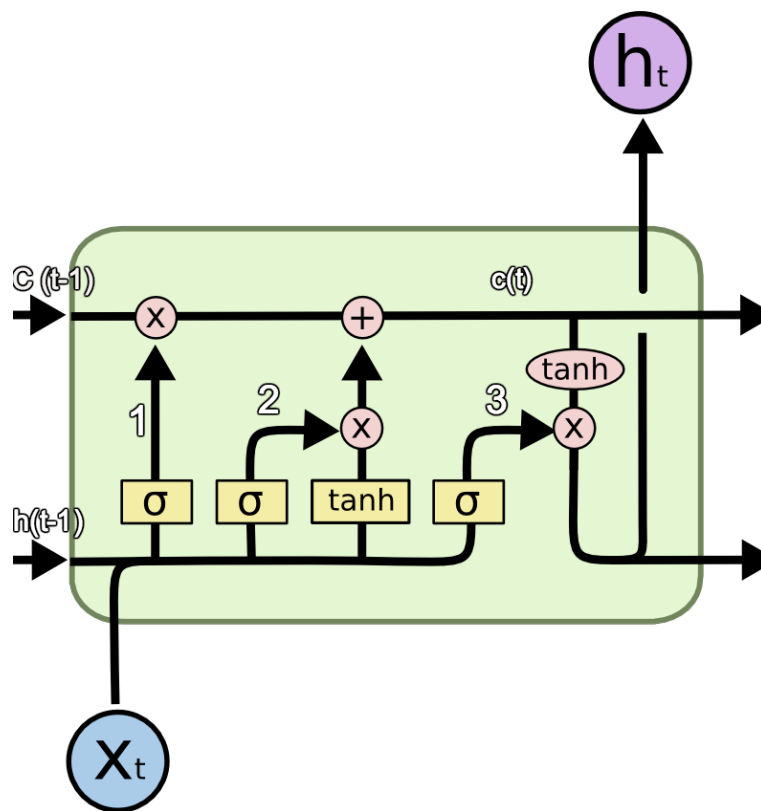


Figura 2.3: Celda LSTM. Fuente.

de una distribución probable para esa entrada.

La estructura del VAE se puede descomponer en dos partes principales:

1. **Encoder Probabilístico:** Transforma los datos de entrada en parámetros de una distribución probabilística en el espacio latente. Generalmente, estos parámetros son las medias y varianzas de distribuciones normales que definen cómo se dispersan los datos de entrada en este nuevo espacio.
2. **Decoder Probabilístico:** Muestra aleatoriamente puntos de la distribución aprendida en el espacio latente y los reconstruye en el espacio de datos original. Este proceso asegura que los datos reconstruidos puedan variar ligeramente, introduciendo una capacidad de generación y una comprensión más profunda de la estructura de los datos.

El entrenamiento de un VAE se basa en la maximización de la evidencia de baja variación (*ELBO*, por sus siglas en inglés), que busca optimizar la red para que sea capaz de generar datos que se parezcan a los de entrenamiento, a la vez que mantiene la regularización de las distribuciones en el espacio latente. Esto se logra mediante una función de pérdida que tiene dos componentes: uno que mide la reconstrucción y otro que penaliza las desviaciones de la distribución latente con respecto a una distribución estándar.

Los VAEs son especialmente valiosos cuando se busca no solo detectar anomalías, sino también comprender la distribución y la generación de los datos normales y anómalos. En mantenimiento predictivo, la capacidad de un VAE para modelar la probabilidad de diferentes estados operativos hace que sea particularmente útil para identificar condiciones de

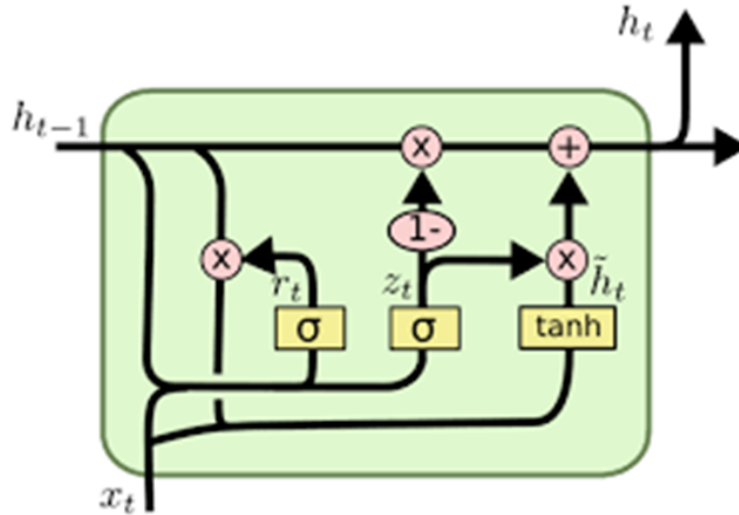


Figura 2.4: Celda GRU. Fuente.

funcionamiento inusuales que podrían indicar el inicio de un fallo.

En resumen, los métodos de *Deep Learning* descritos en esta sección ofrecen un espectro de herramientas poderosas para la detección de anomalías en el mantenimiento predictivo. Desde los autoencoders, que proporcionan una forma eficiente de reducir la dimensionalidad y resaltar las irregularidades, hasta los AutoRNN y VAEs, que traen capacidades avanzadas para entender y procesar secuencias temporales y distribuciones de datos. La selección de la arquitectura y técnica adecuada dependerá del tipo específico de datos y del contexto de aplicación en la industria. Este entendimiento de las herramientas de aprendizaje profundo para la detección de anomalías sienta las bases para explorar, en las siguientes secciones, cómo se pueden implementar y evaluar estas arquitecturas maximizando así su potencial para mejorar su fiabilidad y eficiencia.

2.4. Detección de Anomalías en Condiciones de Cambios Operacionales

Un área significativa de investigación en la detección de anomalías implica abordar los desafíos presentados por los cambios operacionales. Estos cambios pueden ocasionar múltiples falsas alarmas y obsolescencia temprana de los modelos de detección, lo que dificulta la identificación precisa de fallas reales. Un estudio relevante en este contexto es el de [38], donde se explora el uso de deep autoencoders para el monitoreo continuo y la detección temprana de fallas en turbinas eólicas.

El enfoque propuesto por [38] se centra en un modelo de autoencoder profundo (DAE), construido a partir de múltiples máquinas de Boltzmann restringidas (RBM). Este modelo se entrena con datos normales para aprender las interrelaciones entre distintas variables operacionales. Lo innovador de su metodología es la implementación de un umbral adaptativo, determinado mediante la teoría del valor extremo, para mitigar el problema de las

falsas alarmas causadas por fluctuaciones agudas y disturbios en la velocidad del viento durante la operación normal. Este umbral adaptativo permite una discriminación más precisa entre las variaciones operativas normales y las anomalías potencialmente indicativas de fallas.

En [39] se aborda el problema de la evolución en la distribución de un flujo de datos en tiempo real y cómo un aprendizaje adaptativo debe ajustarse a estos cambios para realizar predicciones precisas. El estudio es particularmente relevante en el contexto de la detección de anomalías, donde es crucial diferenciar entre los cambios naturales en la distribución y las verdaderas anomalías.

Dada la naturaleza dinámica de los flujos de datos en tiempo real, donde la distribución puede cambiar con el tiempo, un umbral fijo podría resultar en una alta tasa de falsas alarmas o una baja tasa de detección de anomalías en casos de desviación conceptual. Para abordar esto diseñaron un enfoque adaptativo de ventana deslizante que actualiza el umbral cuando es necesario, basándose en la distribución de las puntuaciones de error. Los resultados experimentales demuestran que este método mejora el rendimiento de los detectores de anomalías base mediante la actualización dinámica del umbral de las puntuaciones cuando es necesario, en lugar de utilizar un umbral fijo o un umbral adaptativo con tamaños de ventana fijos.

Estos enfoques representan un avance significativo en la detección de anomalías bajo condiciones de cambios operacionales, demostrando cómo las técnicas de aprendizaje profundo, combinadas con estrategias de umbrales adaptativo y dinámicos, pueden mejorar la precisión y la fiabilidad en entornos industriales complejos.

2.5. Evaluación y Validación de Modelos

Evaluar algoritmos de detección de anomalías no es una tarea sencilla. La selección de un algoritmo adecuado para un propósito específico implica enfrentarse a una variedad de problemas matemáticos y considerar distintas métricas, cada una adecuada para tareas específicas. Frecuentemente, investigadores y científicos de datos optan por una métrica estándar, como el *F1-score*, simplemente porque es recomendada para tareas de clasificación en general. Sin embargo, esta aproximación puede no ser siempre adecuada en el contexto de la detección de anomalías, donde las características únicas de los datos y los objetivos específicos del modelo pueden requerir un enfoque más matizado.

La evaluación y validación adecuadas son cruciales para garantizar no solo la precisión del algoritmo, sino también su capacidad para funcionar efectivamente en el entorno operativo real. Los desafíos incluyen, pero no se limitan a, el manejo de datos desbalanceados (donde las anomalías son significativamente menos frecuentes que los datos normales), la interpretación de alertas generadas por el modelo, y la adaptación a la evolución de patrones de datos con el tiempo (cambios operacionales). Este panorama complejo hace que la selección de métricas y estrategias de validación adecuadas sea una tarea fundamental y a menudo desafiante en el desarrollo de soluciones de mantenimiento predictivo basadas en *deep learning*.

2.5.1. Métricas de Evaluación

La detección de anomalías, en su esencia, implica diferenciar entre ‘normal’ y ‘anómalo’, transformando así el problema en uno de clasificación binaria. Esta transformación es central en el enfoque de detección de valores atípicos y detección de puntos de cambio. En el contexto de detección de valores atípicos, el objetivo es identificar los datos que se desvían significativamente del patrón general de los datos normales. Por otro lado, la detección de puntos de cambio se enfoca en identificar los momentos en los que ocurre un cambio significativo en la tendencia o comportamiento de una serie temporal.

La matriz de confusión es un concepto clave para entender los diferentes resultados de una predicción. Así es como se ve en una tabla:

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Figura 2.5: Matriz de Confusión. Elaboración Propia.

- VP (verdadero positivo): la predicción ha sido ‘valor anómalo’ y es verdadera;
- VN (verdadero negativo): la predicción ha sido ‘valor normal’ y es verdadera;
- FP (falso positivo): la predicción ha sido ‘valor anómalo’ y es falsa;
- FN (falso negativo): la predicción ha sido ‘valor normal’ y es falsa.

La detección de anomalías a menudo se transforma en un problema de clasificación binaria, también conocido como un problema de detección de valores atípicos (outliers, anomalías individuales), o en un problema de detección de puntos de cambio (anomalías colectivas). Dependiendo del tipo de anomalía que se quiere detectar se utilizan métricas específicas para evaluar los modelos. Las métricas se pueden categorizar en las siguientes clases (Figura 2.6):

1. **Métricas de clasificación binaria:** Categorizan el resultado de la predicción en normal o anómalo basado en las etiquetas de los datos.
 - *Recall:* Esta es la proporción de puntos de datos verdaderamente positivos con respecto al número total de puntos anómalos verdaderos (es decir, los resultados verdaderamente positivos más los falsos negativos en la matriz de confusión).
 - *Precision:* Esta es la proporción de puntos de datos verdaderamente positivos respecto al número total de puntos clasificados como anómalos (es decir, los resultados verdaderamente positivos más los falsos positivos en la matriz de confusión).

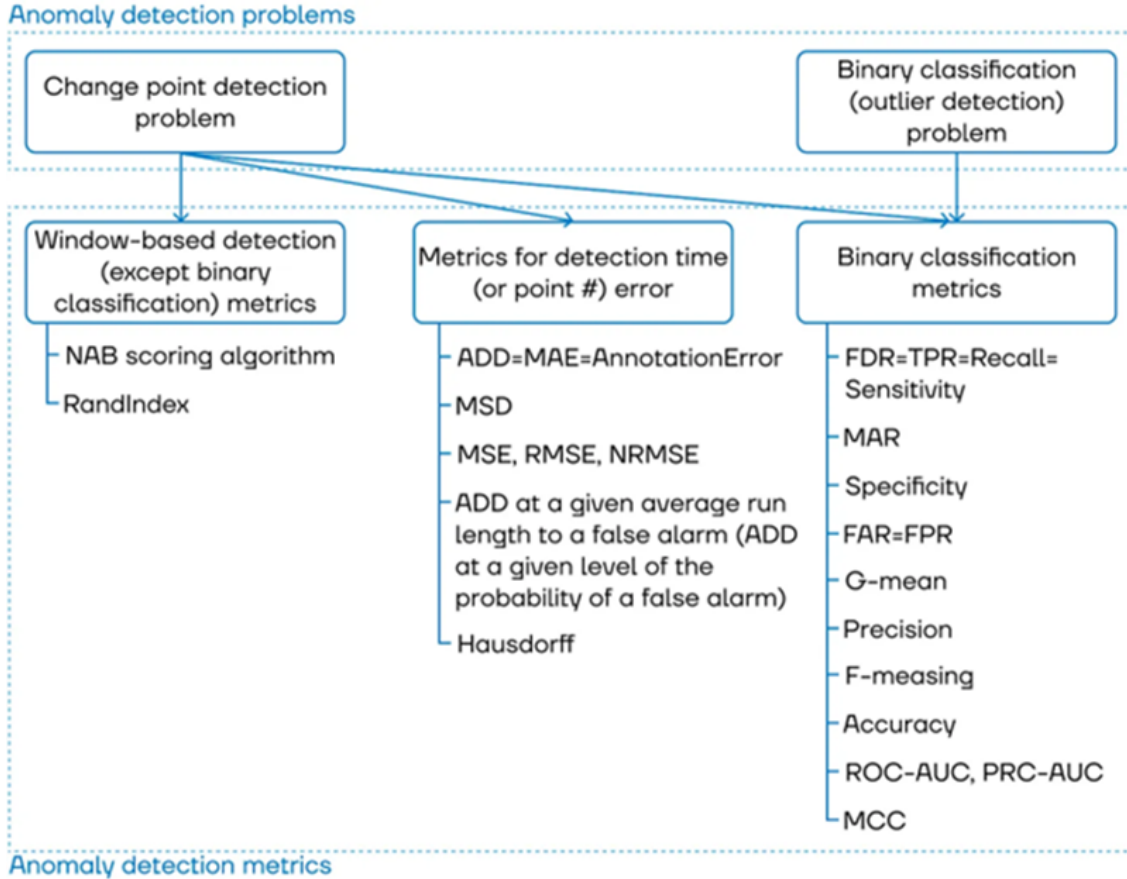


Figura 2.6: Clasificación métricas de evaluación. Fuente.

- *ROC-AUC (Receiver Operating Characteristic, area under the curve), PRC-AUC (Precision-Recall curve, area under the curve)*: Estas son herramientas útiles al predecir la probabilidad de un resultado binario. Se presentan más detalles en [40].

2. **Métricas de detección basadas en ventanas:** Ayudan a hacer coincidir el punto de cambio predicho con la ventana de detección alrededor del punto de cambio real de una manera diferente a la clasificación binaria.

- *NAB scoring algorithm*: Este algoritmo recompensa la detección temprana y correcta y penaliza los falsos positivos y los falsos negativos [41]. El puntaje final NAB se calcula de la siguiente manera:
 - Definición de Pesos y Conjuntos:** Se establecen los pesos ϕ_{TP} , ϕ_{FP} , ϕ_{TN} , y ϕ_{FN} correspondientes a los verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos, respectivamente. Se define D como el conjunto de ventanas de datos y para cada ventana de datos $d \in D$, se identifica A_d como el conjunto de instancias detectadas como anomalías.
 - Función de Puntuación Sigmoidal Escalada:** La función de puntuación para cada detección individual dada una ventana de anomalía y la posición relativa de la detección se define como:

$$\sigma(y) = (\phi_{TP} - \phi_{FP}) \left(\frac{1}{1 + e^{5y}} \right) - 1$$

donde y es la posición relativa de la detección dentro de la ventana de anomalía. Esta función asigna el máximo y mínimo de ϕ_{TP} y ϕ_{FP} para detecciones dentro de la ventana, respectivamente.

- c) **Cálculo del Puntaje en Bruto:** Para cada archivo de datos, el puntaje en bruto S_d se obtiene sumando las puntuaciones de todas las detecciones individuales y ajustando por el número de falsos negativos:

$$S_d = \left(\sum_{y \in Y_d} \sigma'(y) \right) + \lambda_{FN} f_d$$

donde FN representa el número de falsos negativos en el archivo.

- d) **Puntaje NAB Normalizado:** El puntaje NAB final se normaliza como sigue:

$$NAB_{\text{score}} = 100 \times \frac{S_d - Score_{\text{null}}}{Score_{\text{perfect}} - Score_{\text{null}}}$$

donde $Score_{\text{perfect}}$ y $Score_{\text{null}}$ son los puntajes de un detector perfecto y un detector nulo, respectivamente.

- *RandIndex*: Su valor indica que dos ventanas (la predicha y la real) o bien coinciden, o no coinciden en pares de puntos. [42]

3. **Métricas basadas en tiempo de detección:** Ayudan a evaluar la diferencia en tiempo (o número de puntos) entre el punto predicho y el punto de cambio real.

- *ADD (average detection delay)*: Esta métrica denota la diferencia entre el punto de cambio predicho y el punto de cambio real y se puede medir en tiempo o en número de puntos. El valor absoluto de la diferencia en tiempo entre los puntos de cambio predichos y reales se suman y se normaliza sobre cada punto de cambio. [40]
- *MSE (error cuadrático medio)*, *RMSE (raíz del error cuadrático medio)*, *NRMSE (error cuadrático medio normalizado)*: Estas son alternativas al MAE (error absoluto medio). Sin embargo, los errores se elevan al cuadrado, y la medida resultante será muy grande si existen algunos valores atípicos dramáticos en los datos clasificados. [40]

Capítulo 3

Metodología

En esta sección se describe la metodología llevada a cabo para cumplir con los objetivos establecidos en el trabajo de tesis. Como se mencionó en la Sección 1, el objetivo general de esta investigación es desarrollar y aplicar técnicas de *deep learning* no supervisado en el mantenimiento predictivo de equipos industriales, enfrentando desafíos como la falta de datos de falla etiquetados y las variaciones en las condiciones operativas. Para abordar este objetivo, se han definido varios objetivos específicos que guían el desarrollo y la estructura de nuestra metodología. Estos incluyen desde la realización de una revisión exhaustiva de la literatura hasta la validación experimental de las técnicas propuestas en datos reales de equipos industriales.

La metodología adoptada en este trabajo de tesis se estructura en varios pasos clave, cada uno alineado con los objetivos específicos. Estos pasos incluyen la recopilación y preparación de conjuntos de datos representativos, el diseño e implementación de arquitecturas de aprendizaje profundo, y la evaluación y comparación de los modelos desarrollados. A continuación, se detalla cada uno de estos pasos.

3.1. Selección y descripción de los conjuntos de datos

El primer paso en la metodología propuesta es la cuidadosa selección de conjuntos de datos representativos. Estos conjuntos deben poseer las características y desafíos inherentes a las condiciones industriales reales, como la falta de etiquetas y la presencia de variaciones operacionales. Identificar casos de estudio pertinentes es crucial para desarrollar y validar soluciones de detección de anomalías eficientes. Se busca, en particular, conjuntos de datos que permitan tanto la detección de comportamientos anómalos como el abordaje de los retos que supone trabajar con datos no etiquetados y la adaptabilidad ante los cambios operacionales.

La selección de estos conjuntos de datos se describe y justifica en el Capítulo 4: Casos de Estudio, donde se proporciona un análisis detallado de cada caso de estudio elegido. Allí se discuten las características específicas de cada conjunto de datos, la relevancia dentro del contexto de la detección de anomalías y cómo cada uno aporta al objetivo de implementar soluciones efectivas de mantenimiento predictivo con aprendizaje automático no supervisado.

3.2. Pre-procesamiento y Análisis exploratorio de los datos

Antes de aplicar cualquier técnica de aprendizaje profundo, es esencial realizar un análisis exploratorio detallado de los datos (EDA). Esta etapa inicial incluye una exploración minuciosa de las características básicas de los conjuntos de datos, como distribuciones de variables, estadísticas descriptivas y la identificación de valores atípicos o faltantes. Este paso es fundamental para adquirir una comprensión profunda de la naturaleza y estructura de los datos con los que estamos trabajando.

Una vez que tenemos una comprensión clara de los datos, el siguiente paso es abordar cualquier problema de calidad de datos, incluyendo la gestión de valores faltantes y la mitigación de valores atípicos. Dependiendo de la naturaleza y el volumen de los datos faltantes, se pueden emplear diversas técnicas, como la imputación, eliminación de registros o la utilización de algoritmos que manejen eficientemente estas ausencias. Para los valores atípicos, se evaluará su impacto y se decidirá si deben ser ajustados o eliminados, siempre considerando la preservación de la integridad y representatividad de los datos originales.

El EDA también implica la búsqueda de patrones, tendencias y relaciones intrínsecas en los datos. Esto incluye el análisis de series temporales, la correlación entre variables y la identificación de comportamientos típicos y atípicos. Este análisis puede revelar características útiles y comprensiones que guiarán el diseño de nuestros modelos de deep learning, asegurando que se enfoquen en aspectos relevantes y significativos de los datos.

Finalmente, los datos serán transformados y preparados para su uso en modelos de aprendizaje profundo. Esto puede incluir normalización o estandarización de las variables, codificación de categorías, creación de características derivadas, y estructuración de los datos de manera que sean adecuados para el tipo de modelo de deep learning seleccionado. Esta etapa es crucial para garantizar que los modelos puedan aprender de manera efectiva y eficiente de los datos.

3.3. Implementación de modelos

Se comenzará la implementación de modelos el dataset SKAB, que está etiquetado y nos permite una validación y comparación directa de los modelos a través de métricas. Este enfoque nos ayudará a establecer un baseline sólido con técnicas clásicas de aprendizaje automático, como árboles de decisión, así como a evaluar la efectividad de arquitecturas de deep learning más complejas, incluyendo CNN, RNN, LSTM, GRU y autoencoders. La elección de estos modelos se basa en el estudio bibliográfico y en el estado del arte.

Una vez establecido el baseline y evaluadas las arquitecturas en el dataset SKAB, se avanzará con el modelo más prometedor en el dataset no supervisado de la planta de ósmosis inversa. Este paso nos permitirá validar y adaptar nuestro modelo a un entorno de operación real y más complejo, proporcionando insights valiosos sobre la aplicabilidad y robustez de nuestro enfoque en un escenario práctico.

Para asegurar la eficacia y precisión de los modelos implementados, se llevará a cabo un proceso de Random Search para la optimización de hiperparámetros en cada uno de los algoritmos y arquitecturas utilizadas. Esta técnica de optimización nos permite explorar de manera eficiente el espacio de hiperparámetros, identificando las configuraciones que ofrecen el mejor rendimiento.

3.4. Evaluación y comparación de métricas

Cada modelo implementado será evaluado y comparado utilizando un conjunto estandarizado de métricas. Esta comparación nos permitirá identificar las arquitecturas que ofrecen el mejor desempeño en términos de eficacia en la detección de anomalías. Para la base de datos de SKAB se tomarán varias métricas en consideración gracias a la presencia de etiquetas, mientras que en el dataset de la planta de ósmosis inversa, al ser un enfoque no supervisado, se considerará el error de reconstrucción como métrica principal.

En SKAB, para evaluar las anomalías colectivas, se empleará el NAB (Numenta Anomaly Benchmark) scoring algorithm [41]. Este algoritmo es valioso por su capacidad para recompensar la detección temprana y correcta de anomalías, mientras penaliza tanto los falsos positivos como los falsos negativos, proporcionando así una medida integral del rendimiento del modelo en contextos temporales.

Adicionalmente, se utilizará la métrica de ADD (Average Detection Delay) para medir el retraso promedio en la detección de anomalías, ofreciendo una visión de la rapidez con que el modelo puede identificar un comportamiento anómalo desde su inicio. El MSE (Mean Squared Error) también se incluirá como una medida del error cuadrático medio entre los puntos de cambio predichos y los reales, proporcionando un indicador cuantitativo de la precisión de la localización de las anomalías.

En el caso de la clasificación binaria de anomalías, se adoptará el F1 score como una medida balanceada de la precisión y la sensibilidad del modelo. Esta métrica es especialmente útil en situaciones donde las clases están desbalanceadas, como es a menudo el caso en la detección de anomalías.

La tasa de falsos positivos, que indica la frecuencia con la que se identifican erróneamente estados normales como anómalos, será otra métrica clave. La minimización de esta tasa es crítica para reducir las alertas innecesarias y mantener la confianza en las alertas del sistema. Por otro lado, la tasa de fallo de alarmas (Missing Alarm Rate) se utilizará para evaluar la frecuencia con la que el modelo no logra identificar una anomalía real. Este equilibrio entre la detección y la omisión de anomalías es fundamental para ajustar la sensibilidad del modelo y garantizar su utilidad práctica.

Una vez identificados los modelos más prometedores, procederemos a realizar ajustes finos (fine-tuning) para optimizar su rendimiento. Esto puede incluir la modificación de hiperparámetros, la experimentación con diferentes estructuras de capas y neuronas, y la aplicación de técnicas como el aumento de datos, si es aplicable. El objetivo de esta fase es refinar los modelos para maximizar su eficiencia y precisión, especialmente en el contexto de los datos

industriales reales.

Los modelos que muestren un desempeño superior en la base de datos SKAB serán posteriormente probados el dataset de la planta de osmosis inversa. Este conjunto de datos nos permitirá observar cómo cada modelo se adapta a condiciones operacionales variables, incluyendo diferentes velocidades de operación que son críticas para el mantenimiento predictivo de maquinaria. El análisis en este conjunto nos proporcionará una comprensión valiosa de la robustez de los modelos frente a cambios en el entorno operativo.

Capítulo 4

Casos de Estudio

En esta sección se presentan los casos de estudio, los cuales permiten validar la metodología propuesta y evaluar la eficacia de los modelos de deep learning para la detección de anomalías bajo condiciones experimentales controladas y en escenarios industriales reales. El primer caso de estudio se basa en la utilización de la base de datos SKAB (Skoltech Anomaly Benchmark), un entorno controlado y experimental que provee datos operativos normales junto con anomalías debidamente etiquetadas. Esta configuración permite un análisis detallado y preciso del comportamiento del modelo en condiciones ideales. El segundo caso de estudio se adentra en el complejo mundo de los datos industriales reales, presentando retos adicionales como fluctuaciones de señales, diversidad en los modos de operación, y problemas inherentes a la recopilación de datos, tales como ruido, datos faltantes y la presencia de patrones operativos no documentados. La confrontación del modelo con este conjunto de datos tiene como fin último corroborar su robustez y aplicabilidad en un contexto práctico, marcando así la transición de la teoría a la práctica.

4.1. Base de Datos SKAB

SKAB (Skoltech Anomaly Detection Benchmark) [43] es una base de datos diseñada específicamente para servir como referencia y punto de comparación para diversas arquitecturas en la detección de anomalías. Este conjunto de datos cuenta con 2 etiquetas para cada fila de datos, ya que permite trabajar con dos problemas principales: la detección de valores atípicos, donde las anomalías se consideran y marcan como anomalías de un solo punto, y la detección de puntos de cambio, donde las anomalías se consideran y marcan como anomalías colectivas.

La relevancia de SKAB radica en su capacidad para proporcionar un entorno estandarizado y controlado para probar y comparar ya sea modelos clásicos o métodos de *deep learning*. Esto es esencial para evaluar objetivamente la eficacia y precisión de diversas técnicas en la detección de anomalías. SKAB contiene 35 archivos individuales en formato .csv, que representan una serie de experimentos, cada uno conteniendo una sola anomalía. Los tipos de anomalías están relacionados al bloqueo de válvulas, desalineamiento del motor y variación en sensores de aceleración y amperaje.

Los datos recopilados de los sensores instalados en el banco de pruebas (Figura 4.1) incluyen múltiples variables como aceleración de vibración, amperaje, presión, temperatura y flujo, proporcionando así una rica fuente de información para el análisis.

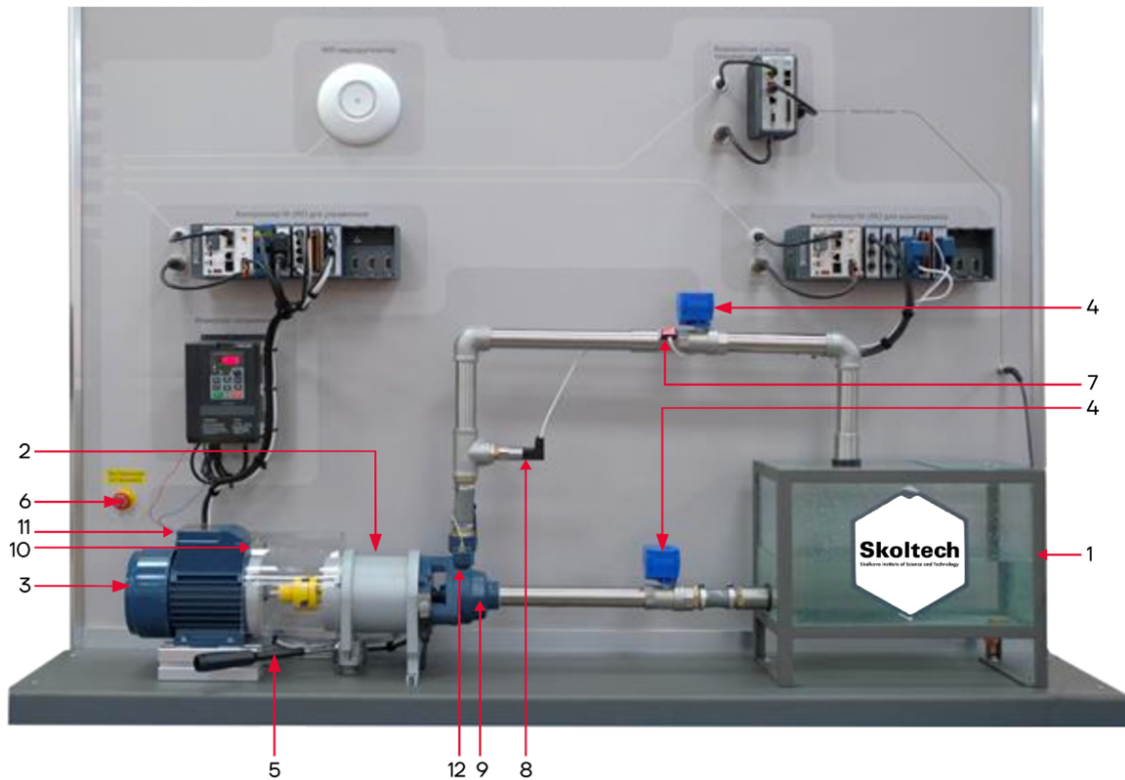


Figura 4.1: Panel frontal sistema de circulación, control y monitoreo de agua. [43]

Las partes que se pueden observar en la Figura 4.1 son:

- un tanque con agua
- una bomba de agua
- motor eléctrico
- válvula solenoide
- una palanca mecánica para desalineación de ejes
- botón de parada de emergencia
- inversor
- compactRIO
- medidor de presión
- sensores de vibración
- medidor de flujo
- termopar

4.2. Conjunto de datos de operación industrial real

El segundo conjunto de datos utilizado en esta investigación proviene de una planta desaladora de agua de ósmosis inversa. Este dataset es particularmente valioso debido a su origen en un entorno industrial real, ofreciendo una perspectiva auténtica sobre los desafíos operativos y las condiciones variables a las que están sujetos los equipos industriales.

La ósmosis inversa es un proceso crítico para las operaciones mineras del norte del país, ya que es utilizado para convertir agua de mar en agua desalinizada. Este proceso implica forzar el paso del agua a través de membranas semipermeables bajo alta presión, lo que permite la separación de sal y otros minerales disueltos. Es un método eficiente y sostenible para producir agua para uso industrial en áreas donde el agua dulce es escasa.

Dentro de este proceso existen múltiples equipos claves que participan activamente, entre los cuales se encuentran:

- **Bombas de Captación de Agua de Mar:** Estas bombas son responsables de extraer agua del mar y suministrarla al sistema de ósmosis inversa.
- **Bombas de Alta Presión:** Estas bombas aumentan la presión del agua de mar para facilitar el proceso de ósmosis a través de las membranas.
- **Bombas Booster:** Utilizadas para mantener la presión necesaria en el sistema y asegurar un flujo constante a través de las membranas.
- **Filtros Racks de Ósmosis Inversa:** Contienen las membranas semipermeables donde ocurre el proceso de separación de sal y minerales.
- **Recuperadores de Energía (ERI):** Estos dispositivos recuperan parte de la energía utilizada en el proceso para mejorar la eficiencia energética del sistema.
- **Bombas de Impulsión:** Encargadas de transportar el agua desalinizada a su destino final.

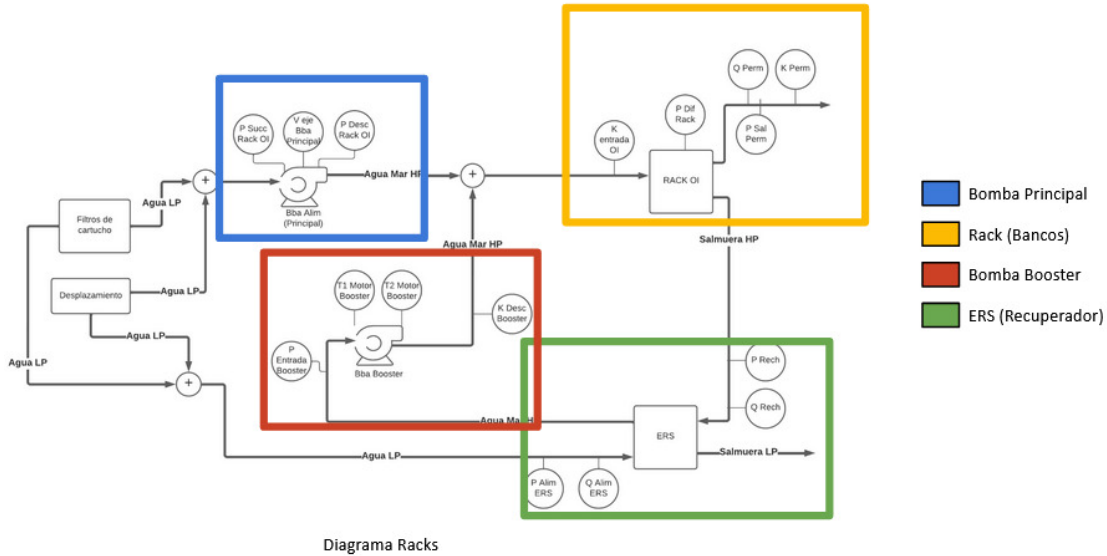


Figura 4.2: Diagrama equipos participantes en el proceso de ósmosis inversa. Elaboración propia.

Para el propósito de este estudio, se concentró la recopilación de datos en las bombas de alta presión y las bombas booster (4.2). La elección de estos equipos no fue solo por su importancia crítica en el proceso, sino también debido a su exposición a cambios operacionales frecuentes. Estos equipos experimentan variaciones significativas en sus condiciones de operación, lo que los hace candidatos ideales para evaluar la eficacia de los modelos de deep learning en el manejo de cambios operativos y la detección precisa de anomalías.

Estos cambios operacionales en las bombas pueden presentar desafíos únicos en términos de mantenimiento predictivo, ya que las fluctuaciones en su funcionamiento pueden enmascarar o simular señales de fallas incipientes. Por lo tanto, analizar los datos de estas bombas proporciona una oportunidad valiosa para examinar cómo los modelos desarrollados se adaptan y responden a dichas variaciones operativas, diferenciando efectivamente entre comportamientos normales fluctuantes y verdaderas anomalías.

Las principales variables que se monitorean, debido a que proporcionan información vital del funcionamiento operativo del proceso son:

- **Vibraciones:** Monitoreada en las bombas y otros equipos mecánicos para detectar signos tempranos de desgaste o fallas.
- **Temperaturas:** Esencial para el control del proceso y para la prevención de daños en los equipos.
- **Presiones:** Tanto en las bombas como en las membranas, la presión es un indicador clave del rendimiento del proceso.
- **Caudales:** El flujo de agua a través del sistema debe mantenerse dentro de rangos óptimos para la efectividad de la ósmosis.
- **Conductividad:** Se mide para evaluar la salinidad del agua y la eficiencia de la desalinización.

- **Velocidad:** Monitoreo de las bombas y otros componentes para distinguir rangos de operaciones.

Cabe destacar que, aunque este estudio se enfoca en las bombas de alta presión y las bombas booster, los modelos y métodos desarrollados tienen el potencial de ser aplicados a otros equipos críticos en la planta de ósmosis inversa. Equipos como los filtros racks de ósmosis inversa, los recuperadores de energía (ERI) y las bombas de impulsión también pueden beneficiarse de un monitoreo predictivo basado en deep learning. Proponemos como una futura línea de investigación la expansión de esta metodología para incluir estos otros componentes, lo que podría proporcionar una visión aún más completa del estado de salud de toda la planta y mejorar aún más las estrategias de mantenimiento preventivo.

Capítulo 5

Resultados

5.1. Resultados de la Base de Datos SKAB

A continuación, se presentan los resultados obtenidos a partir de la base de datos SKAB, que desempeña un papel esencial como punto de referencia en la evaluación de algoritmos de detección de anomalías. Los experimentos realizados y las métricas resultantes están diseñados para proporcionar una comprensión detallada del rendimiento de diversos modelos de aprendizaje automático y deep learning en la identificación de patrones anómalos. La siguiente exposición de análisis y visualizaciones ofrece una visión clara de la capacidad de cada modelo para discernir entre el comportamiento normal y el anómalo.

5.1.1. Descripción de la Base de Datos SKAB

La base de datos SKAB consta de una serie de archivos, donde cada uno corresponde a un experimento individual que captura una anomalía única. Excepcionalmente, hay un archivo libre de anomalías que se utiliza como referencia para el modo de operación normal. La base de datos es una serie temporal multivariada obtenida de los sensores instalados en un banco de pruebas (Figura 4.1). La estructura de la carpeta de datos es la siguiente:

1. **anomaly-free:** Datos obtenidos bajo condiciones normales de funcionamiento sin presencia de anomalías.
2. **valve1:** Datos de experimentos con una válvula cerrada en la salida del flujo de la bomba (16 experimentos).
3. **valve2** Datos de experimentos con una válvula cerrada en la entrada del flujo a la bomba (4 experimentos).
4. **other:** Datos de otros experimentos, que incluyen simulaciones de fugas y adiciones de fluido, así como distintos comportamientos de desequilibrio del rotor y variaciones en la cantidad de agua en el circuito (14 experimentos).

En total se tienen 34 experimentos donde para cada serie temporal se tienen las siguientes columnas:

- **datetime:** Registra las fechas y horas exactas de cada medición (YYYY-MM-DD hh:mm:ss).

- **Accelerometer1RMS y Accelerometer2RMS:** Mediciones de aceleración de vibración (en unidades de gravedad).
- **Current:** Corriente del motor eléctrico (Amperios).
- **Pressure:** Presión en el circuito después de la bomba de agua (Bar).
- **Temperature:** Temperatura del cuerpo del motor (Grados Celsius).
- **Thermocouple:** Temperatura del fluido en el circuito de circulación (Grados Celsius).
- **Voltage:** Voltaje en el motor eléctrico (Voltios).
- **RateRMS:** Caudal de circulación del fluido (Litros por minuto).
- **anomaly:** Etiqueta de si el punto es anómalo (0 o 1).
- **changepoint:** Etiqueta de si el punto es un punto de cambio para anomalías colectivas (0 o 1).

5.1.2. Análisis Exploratorio de Datos (EDA) de SKAB

El Análisis Exploratorio de Datos (EDA) se inició con la elaboración de una tabla resumen que encapsula los aspectos clave de los 34 experimentos contenidos en la base de datos SKAB (A.3). Esta tabla fue diseñada para proporcionar una vista panorámica de cada experimento, incluyendo variables como la duración del experimento, el porcentaje de tiempo en modo anormal, la cantidad de puntos de cambio de estado y la presencia de valores faltantes. La construcción de esta tabla resumen facilita la comprensión inicial de los datos disponibles y establece una base sólida para los análisis siguientes.

Desde esta tabla, se pueden observar los siguientes puntos:

- La duración de los experimentos es de aproximadamente 20 minutos, excepto para el experimento **anomaly_free**, cuya duración fue de 2 horas y 46 minutos.
- El porcentaje de tiempo con un modo anormal está en el rango del 25.83% al 57.74%, pero para la mayor parte de los experimentos, este valor se encuentra alrededor del 35%.
- En la mayoría de los experimentos, se registran 4 puntos de cambio de estado.
- No hay valores faltantes en los datos.

Además, se creó una tabla estadística detallada de los datos (A.1), desglosada por tipo de experimento. Desde esta tabla, se pueden distinguir los siguientes puntos:

- El valor promedio de la aceleración de vibración en los experimentos sin anomalías (**anomaly free**) es casi un orden de magnitud más alto que en los experimentos con cierre de válvulas de entrada y salida (**valve1, valve2**).
- La tasa de flujo de agua promedio en los experimentos sin anomalías (**anomaly free**) es cuatro veces mayor que en los experimentos con cierre de válvulas de entrada y salida (**valve1, valve2**).

- Los valores promedio de temperatura, presión, corriente y termopar en los experimentos sin anomalías (anomaly free) son más altos que en los experimentos con cierre de válvulas de entrada y salida (valve1, valve2).

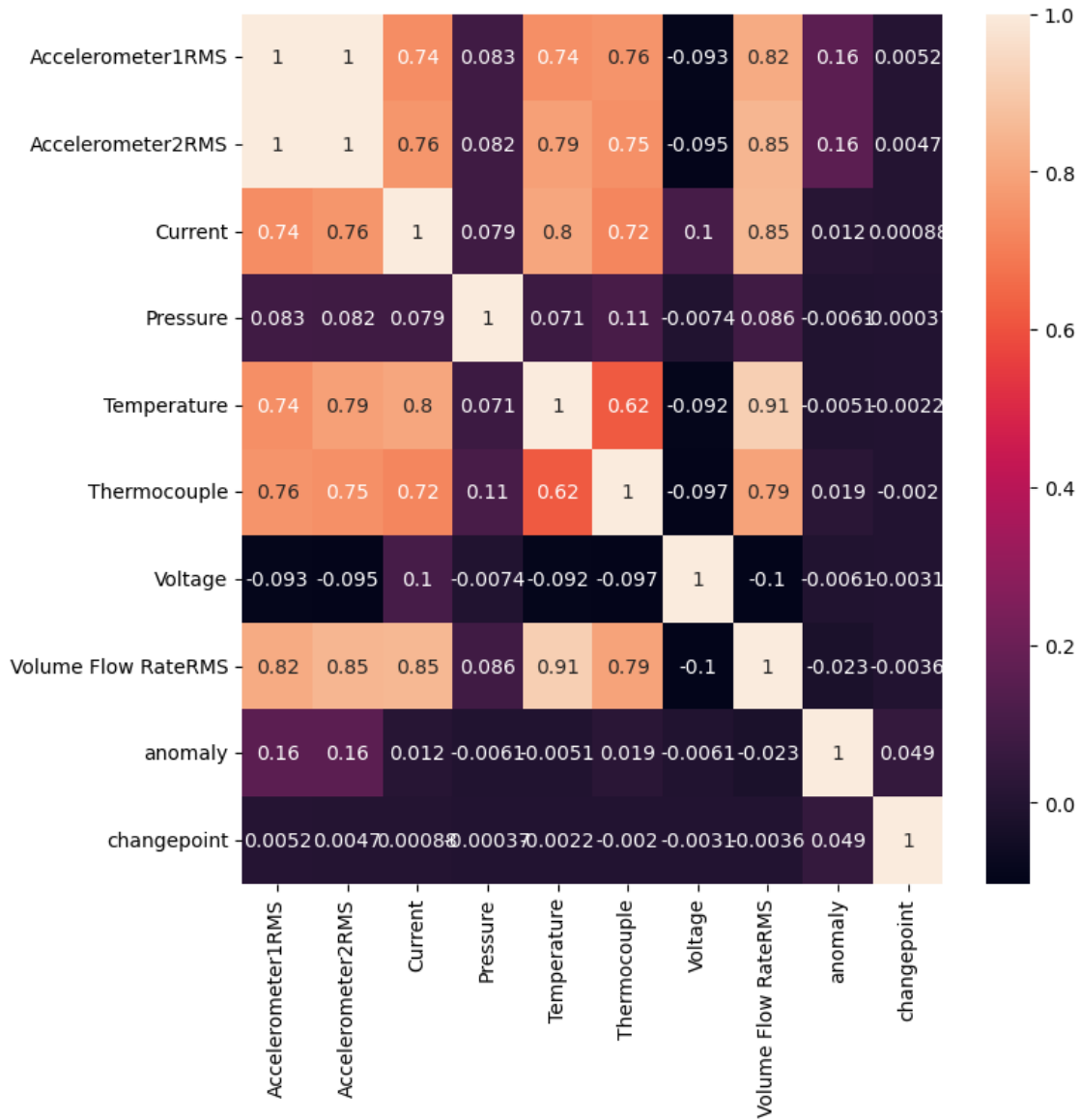


Figura 5.1: Matriz de Correlación BBDD SKAB.

A continuación, se procedió a obtener la matriz de correlación (5.1). Esta matriz visualiza las relaciones entre diferentes variables medidas durante los experimentos. Los siguientes puntos resumen los hallazgos clave de esta matriz:

- **Alta Correlación entre Variables de Vibración y Flujo:** Existe una fuerte correlación positiva entre las mediciones de vibración (Accelerometer1RMS y Accelerometer2RMS) y la tasa de flujo volumétrico (Volume Flow RateRMS). Esto indica que a medida que la vibración del equipo aumenta, también lo hace el flujo de volumen, lo que puede sugerir una relación directa entre el funcionamiento del equipo y el comportamiento del flujo de fluido.

- **Baja Correlación con Anomalías y Puntos de Cambio:** Tanto las anomalías (anomaly) como los puntos de cambio (changepoint) muestran bajas correlaciones con las demás variables. Esto sugiere que las anomalías y los puntos de cambio no están fuertemente influenciados por las variaciones en las mediciones sensoriales recogidas y podrían estar más relacionados con eventos puntuales o condiciones operativas específicas.
- **Relación entre Variables Eléctricas y Térmicas:** Se observa una correlación significativa entre las variables eléctricas (Current) y las variables térmicas (Temperature y Thermocouple). Esto podría reflejar el hecho de que un aumento en el consumo de corriente del motor eléctrico está asociado con un aumento en la temperatura de funcionamiento.
- **Relaciones Negativas con el Voltaje:** Se observa una correlación negativa entre el voltaje (Voltage) y varias otras variables, aunque estas correlaciones son en su mayoría débiles. Esto podría indicar que ciertas condiciones operativas que llevan a un menor voltaje no están directamente asociadas con un aumento en la mayoría de las mediciones de sensor.

Cabe destacar que la correlación no implica causalidad, y aunque dos variables puedan estar correlacionadas, no significa que una cause la otra. Las correlaciones fuertes, sin embargo, pueden ser indicativas de relaciones que merecen una investigación más profunda. Para buscar relaciones entre variables, también se creó y se usó la función pairplot de la librería seaborn, la cual combina todos los posibles gráficos e histogramas entre pares de variables (A.1).

En A.2 se puede observar como ejemplo las señales para el experimento N° 12 de la base de datos.

5.1.3. Arquitectura de los Modelos Implementados

En esta sección, se detalla la arquitectura de los modelos de aprendizaje profundo implementados para la detección de anomalías y puntos de cambio. Se proporcionan descripciones de las configuraciones de las capas, las funciones de activación utilizadas, los detalles de los procesos de entrenamiento y los criterios de evaluación.

5.1.3.1. Isolation Forest

El modelo Isolation Forest fue configurado con los siguientes parámetros:

- **n_estimators:** Número de árboles de decisión, establecido en 100.
- **max_samples:** El número de muestras a extraer de los datos para entrenar cada árbol base. Se dejó en su valor predeterminado.
- **contamination:** Proporción esperada de outliers en los datos, establecida en 0.0005 para reflejar una baja prevalencia de anomalías.
- **max_features:** Número de características a extraer para entrenar cada árbol base, establecido en 1 para utilizar todas las características disponibles.

Con estos parámetros, el Isolation Forest busca aislar y detectar anomalías de forma rápida y eficiente, aprovechando su capacidad para trabajar con conjuntos de datos de alta dimensionalidad.

5.1.3.2. Elliptic Envelope

El modelo Elliptic Envelope se utilizó para identificar las anomalías bajo la suposición de que los datos siguen una distribución gaussiana. Se ajustaron los siguientes parámetros:

- **assume_centered:** Establecido en False, lo que indica que los datos no están centrados y que el modelo debe estimar la ubicación central como parte de su ajuste.
- **support_fraction:** La proporción de puntos que se consideran soporte del modelo de distribución estimada, fijada en 0.9, lo que implica que se utiliza el 90% de los datos para el ajuste.
- **contamination:** La proporción de outliers esperada en el conjunto de datos, establecida en 0.1, que indica que se espera que el 10% de los puntos sean anómalos.

5.1.3.3. Autoencoder

El Autoencoder se estructura en tres componentes principales: el codificador o encoder, el espacio latente y el decodificador o decoder. La arquitectura del modelo se define de la siguiente manera:

- **Codificador:**
 - Capa de entrada que recibe secuencias con la dimensión de entrada (número de features)
 - Varias capas Dense, cada una con una activación 'relu', para procesar las características. Estas capas reducen progresivamente la dimensión, conduciendo hacia el espacio latente.
 - Capa Dense final que representa el espacio latente, con una dimensión más pequeña que la entrada original.
- **Espacio Latente:**
 - Representación comprimida de los datos de entrada, sirviendo como punto de transición entre el codificador y el decodificador.
- **Decodificador:**
 - Capas Dense que progresivamente aumentan en dimensión, utilizando también activación 'relu', para reconstruir los datos desde el espacio latente.
 - La última capa Dense reconstruye la entrada con las mismas dimensiones originales, completando el proceso de autoencoder.

Para el entrenamiento del Autoencoder, se utilizó el optimizador Adam con la función de pérdida de error cuadrático medio (`mean_squared_error`). Adam [44] es un optimizador eficiente que combina las ventajas de dos extensiones del descenso de gradiente estocástico: Adaptive Gradient Algorithm (AdaGrad) y Root Mean Square Propagation (RMSProp). Esta elección es adecuada para problemas de reconstrucción debido a su capacidad para manejar gradientes dispersos y adaptarse eficientemente a diferentes tipos de datos y arquitecturas gracias al ajuste automático del learning rate durante el entrenamiento.

En el desarrollo de la arquitectura del autoencoder con capas MLP, se empleó el método de RandomSearch para optimizar los hiperparámetros del modelo (5.1). Este enfoque permitió explorar eficientemente un amplio espacio de posibles configuraciones, evaluando una diversidad de combinaciones. Se determinó que la configuración óptima consistía en un codificador y un decodificador con 2 capas cada uno, 4 unidades por capa, un tamaño de lote de 64 y un total de 100 épocas. Esta combinación específica de hiperparámetros demostró ser la más efectiva en términos de precisión y eficiencia en la tarea de reconstrucción de datos, lo que se traduce en una mejor capacidad para detectar anomalías.

Tabla 5.1: Grilla de RandomSearch para la Optimización de Hiperparámetros AutoMLP

Hiperparámetro	Rango o Valores Probados
epochs	50, 75, 100
batch_size	16, 32, 64
capas	2, 4, 8
neuronas por capa	4, 6, 8

5.1.3.4. Autoencoder Convolucional (AutoCNN)

El Autoencoder CNN implementado en este estudio es una red neuronal profunda diseñada específicamente para la detección de anomalías en conjuntos de datos con patrones espaciales complejos. La arquitectura de este modelo se compone de varias capas de convolución y convolución inversa, organizadas para comprimir y luego reconstruir la entrada original. A continuación, se detallan las capas y configuraciones clave del modelo:

- **Capa de Entrada:** Recibe datos con una forma de entrada de (ventana de tiempo, número de features).
- **Capa de Resize:** Transforma la entrada bidimensional a 3 dimensiones, preparándola para el procesamiento convolucional.
- **Bloques Convolutivos:** Cada bloque convolutivo se compone de las siguientes capas y operaciones:
 1. *Capa Convolutiva 1 (Conv2D):* Aplica una operación convolutiva con un número especificado de filtros y un tamaño de kernel dado, manteniendo el tamaño con el uso de padding.
 2. *BatchNormalization:* Normaliza la salida de la capa anterior, reduciendo la variabilidad interna de las activaciones y mejorando el proceso de entrenamiento.
 3. *Activation:* Aplica la función de activación ReLU
 4. *Capa Convolutiva 2 (Conv2D):* Una segunda capa convolutiva con las mismas especificaciones que la primera para profundizar en la extracción de características.
 5. *BatchNormalization y Activation:* Se repiten las operaciones de normalización y activación.

6. *Submuestreo (MaxPool2D)*: Opcionalmente, se realiza submuestreo para reducir las dimensiones espaciales y aumentar la abstracción de las características.

• **Bloques de Deconvolución** Cada bloque de deconvolución está diseñado para reconstruir progresivamente las dimensiones espaciales a su tamaño original:

1. *Capa de Deconvolución 1 (Conv2DTranspose)*: Realiza una operación de deconvolución para expandir las dimensiones espaciales.
2. *BatchNormalization y Activation*: Aplica operaciones de normalización y activación.
3. *Capa de Deconvolución 2 (Conv2DTranspose)*: Una segunda capa de deconvolución para continuar la reconstrucción de las características.
4. *BatchNormalization y Activation*: Aplica operaciones de normalización y activación.
5. *Sobremuestreo (UpSampling2D)*: Opcionalmente, realiza un sobremuestreo para aumentar aún más las dimensiones espaciales, si upsampling está habilitado.

• **Capa final**: Restaura la salida del modelo a la forma inicial

Para el entrenamiento del Autoencoder CNN, se utilizó el optimizador Adam con la función de pérdida de error cuadrático medio (`mean_squared_error`). Adam [44] es un optimizador eficiente que combina las ventajas de dos extensiones del descenso de gradiente estocástico: Adaptive Gradient Algorithm (AdaGrad) y Root Mean Square Propagation (RMSProp). Esta elección es adecuada para problemas de reconstrucción debido a su capacidad para manejar gradientes dispersos y adaptarse eficientemente a diferentes tipos de datos y arquitecturas gracias al ajuste automático del learning rate durante el entrenamiento.

Para afinar los hiperparámetros del modelo y asegurar su óptimo rendimiento, se implementó un enfoque de `RandomSearch()`. Esto permitió identificar la mejor combinación de valores en una grilla (5.2) para maximizar la eficacia del modelo en la detección de anomalías.

Tabla 5.2: Grilla de `RandomSearch` para la Optimización de Hiperparámetros AutoCNN

Hiperparámetro	Rango o Valores Probados
<code>epochs</code>	50, 75, 100
<code>batch_size</code>	4, 8, 16, 32
<code>time_size</code>	8, 16, 32
<code>block_num</code>	2, 4, 6
<code>filter_num</code>	4, 8, 16, 32
<code>kernel_v</code>	2, 4
<code>kernel_h</code>	2, 4

Después de ejecutar `RandomSearch()` en 10 iteraciones y evaluaciones, los siguientes valores demostraron proporcionar el mejor rendimiento para nuestro modelo:

- **epochs**: 100
- **batch_size**: 8

- **time_size:** 8
- **block_num:** 4
- **filter_num:** 32
- **kernel_v:** 2
- **kernel_h:** 2

5.1.3.5. Autoencoder Recurrente (Auto RNN)

El Autoencoder recurrente se estructura en dos partes principales: el codificador (encoder) y el decodificador (decoder), las cuales están formadas por redes recurrentes pudiendo tener celdas GRU o LSTM. A continuación se detallan las capas que componen cada parte:

- **Encoder:**

- Capa densa de entrada con la forma (Ventana de tiempo, número de features).
- Capas LSTM/GRU intermedias con dimensiones desde (Ventana de tiempo, número de features) hasta (Ventana de tiempo, espacio latente). Donde el espacio latente es la dimensión del cuello de botella del autoencoder. El número de capas se determina como un hiperparámetro más realizando una búsqueda de grilla.

- **Decoder:**

- El mismo número y dimensiones de capas LSTM/GRU utilizadas en al encoder pero en orden inverso para reconstruir la dimensión original desde (Ventana de tiempo, espacio latente) a (Ventana de tiempo, número de features).

Para el entrenamiento similar al Auto CNN, se utilizó el optimizador Adam con la función de pérdida de error cuadrático medio (mean_squared_error) y se aplicó una búsqueda de grilla (5.3) con RandomSearch para encontrar la mejor configuración de hiperparámetros.

Tabla 5.3: Grilla de RandomSearch para la Optimización de Hiperparámetros Autoencoder LSTM

Hiperparámetro	Rango o Valores Probados
epochs	50, 75, 100
batch_size	8, 16, 32
time_size	10, 20, 30, 40
layers_dim	(6, 4, 2), (4, 2), (6, 4), (4)

Mejor configuración:

- **epochs:** 100
- **batch_size:** 8
- **time_size:** 10
- **layers_dim:** 4

5.1.3.6. Implementación

La implementación de los modelos se realizó utilizando el framework de Tensorflow y Keras, y el código está disponible en [este enlace](#).

5.1.4. Resultados de Modelos en SKAB

A continuación, se presentan los resultados obtenidos de aplicar cada modelo a los datos cuidadosamente preprocesados, revelando su capacidad para identificar y clasificar comportamientos anómalos en un entorno de simulación de condiciones industriales controladas.

5.1.4.1. Modelos Clásicos

Como se mencionó previamente en la metodología, se seleccionaron inicialmente dos modelos clásicos de aprendizaje de máquinas para servir como punto de referencia: Isolation Forest y Elliptic Envelope. Estos modelos fueron escogidos por su eficacia probada en la identificación de valores atípicos y anomalías en conjuntos de datos multidimensionales. Isolation Forest, conocido por su capacidad para aislar anomalías en conjuntos de datos grandes y dispersos, y Elliptic Envelope, que asume una distribución gaussiana de los datos normales, se implementaron para evaluar su rendimiento en la detección de comportamientos anómalos dentro de la base de datos SKAB. A continuación se muestran las matrices de confusión (5.3) para el problema de detección de anomalías y un experimento de ejemplo para visualizar la forma en que se están realizando las predicciones (A.3).

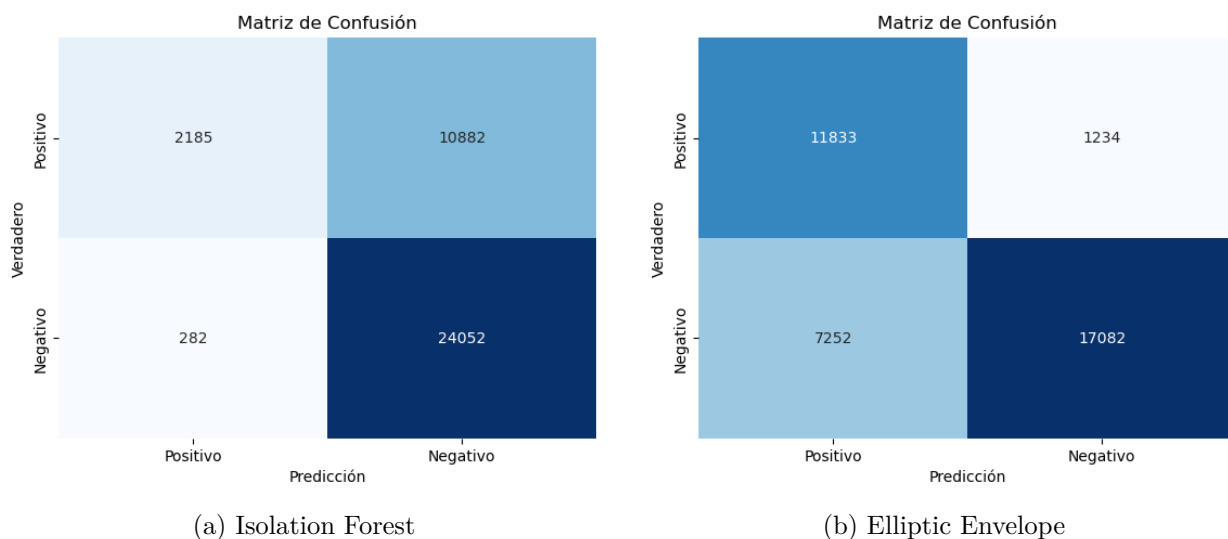


Figura 5.2: Matriz de Confusión Detección de Anomalías

5.1.4.2. Modelos Deep Learning

A continuación se implementaron diversas arquitecturas de deep learning no supervisado para la detección de anomalías, superando las capacidades de los modelos clásicos anteriormente implementados. Para cada modelo implementado, se presentan las matrices de confusión que ilustran la efectividad en la clasificación de los datos como normales o anómalos. Estas matrices proporcionan una visión clara del rendimiento de cada modelo, a partir de la

cual se pueden extraer métricas claves para evaluar el rendimiento. (A.1)

Además, en el anexo (A.16) tanto, se incluyen ejemplos visuales de las reconstrucciones realizadas por cada arquitectura de deep learning, así como casos específicos de detección exitosa de anomalías y puntos de cambio. Estos ejemplos demuestran de manera práctica cómo los modelos procesan y responden a diferentes tipos de datos, proporcionando una comprensión más profunda de su funcionamiento y eficacia.

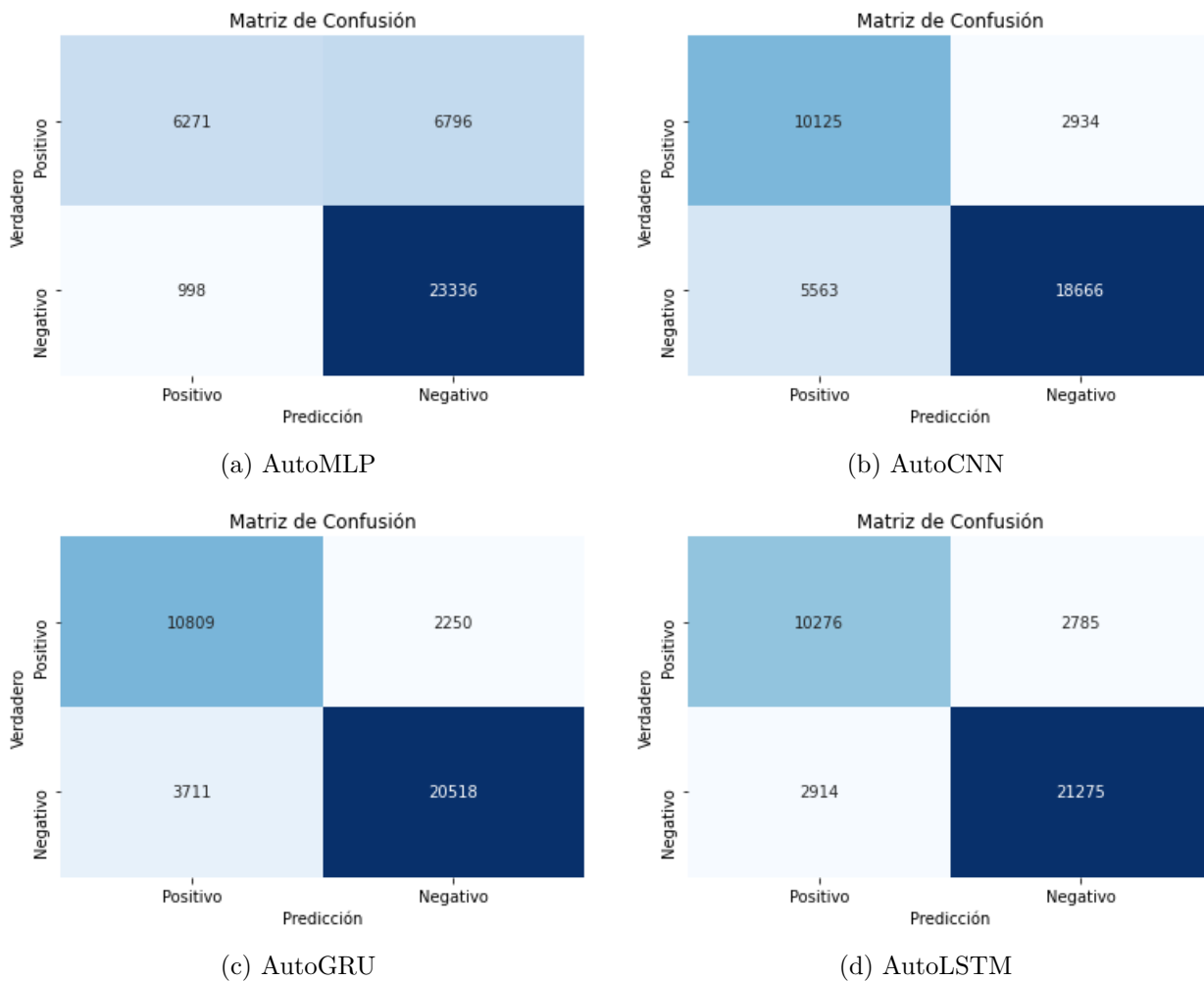


Figura 5.3: Matriz de Confusión Detección de Anomalías Métodos Deep Learning

5.1.4.3. Resumen Métricas

Tras la implementación y evaluación de los modelos de aprendizaje automático y deep learning en la base de datos SKAB, se ha compilado una tabla comparativa para resumir y contrastar sus respectivos rendimientos. Esta tabla (5.4) sintetiza los resultados clave y proporciona una perspectiva clara sobre la eficacia de cada modelo en la tarea de detección de anomalías y puntos de cambio. Se incluyen métricas esenciales como el F1 score, la Missing Alarm Rate (MAR) y la False Alarm Rate (FAR) para la detección de anomalías, y el Numanta Anomaly Benchmark (NAB) score para la detección de puntos de cambio.

Tabla 5.4: Comparativa de métricas de modelos en detección de anomalías y puntos de cambio

Problema	Arquitectura	F1 score	MAR	FAR
Anomaly Detection	LSTM AE	0.78	21.32 %	12.05 %
	GRU AE	0.78	15.32 %	17.23 %
	Elliptic Envelope	0.74	9.44 %	29.8 %
	Conv AE	0.71	22.47 %	22.96 %
	MLP AE	0.61	52.53 %	3.84 %
	Isolation Forest	0.28	83.28 %	1.16 %
Problema	Arquitectura	NAB score		
Changepoint Detection	LSTM AE	26.47		
	GRU AE	25.24		
	Isolation Forest	9.24		
	Conv AE	8.83		
	MLP AE	-3.56		
	Elliptic Envelope	-5.8		

5.2. Resultados Base de Datos Operación Industrial Real

En esta sección, se presentan los resultados obtenidos al aplicar los modelos de detección de anomalías desarrollados en la base de datos industrial real. Esta validación es crucial para entender la aplicabilidad y efectividad de las técnicas propuestas en un entorno práctico y operativo.

5.2.1. Descripción de la Base de Datos Bombas de Alta Presión y Booster

En el marco de este estudio, se ha realizado una recopilación exhaustiva de datos de 12 bombas de alta presión y 12 bombas booster en la planta de ósmosis inversa de Minera Escondida. Estas bombas están asociadas a 12 racks de ósmosis inversa, cada uno con su propio conjunto de sensores para un monitoreo preciso y detallado. La elección de estas bombas y la cantidad de racks monitoreados refleja la escala y la complejidad del sistema de ósmosis inversa en la planta.

El foco de la recolección de datos se ha centrado en dos tipos de equipos críticos: las bombas de alta presión y las bombas booster. Estos equipos desempeñan roles esenciales en el proceso de ósmosis inversa, manejando las variaciones de presión necesarias para la filtración efectiva y el transporte del agua.

Se han recopilado diversas variables operativas que son indicativas del rendimiento y estado de cada tipo de bomba. Para las bombas de alta presión, los siguientes sensores fueron utilizados para recopilar datos:

- **Vibraciones en diferentes ejes (X, Y, axial):** Para detectar posibles desbalances o desgaste en los componentes mecánicos.
- **Temperaturas de los cojinetes y el motor:** Clave para el monitoreo del calor en los equipos y la prevención de sobrecalentamientos.
- **Velocidad del eje principal:** Indicador clave del rendimiento y el modo de operación de la bomba.
- **Presiones de succión y descarga:** Esenciales para evaluar la eficiencia y la operatividad de la bomba.

Para las bombas booster se recopilaron sensores que incluyen:

- **Temperatura y vibraciones de los rodamientos y cojinetes:** Para detectar posibles desbalances o desgaste en los componentes mecánicos.
- **Presión de entrada:** Esenciales para evaluar la eficiencia y la operatividad de la bomba.
- **Velocidad del eje y caudal de rechazo,** que son indicadores de la eficiencia y la efectividad del proceso de ósmosis inversa.

En total, se han recogido en promedio 1,500,000 registros de datos por equipo, lo que refleja la extensa y rigurosa monitorización realizada en la planta de ósmosis inversa. Estos datos fueron recolectados durante un período de un año, desde el 1 de enero de 2023 hasta el 1 de enero de 2024, asegurando una amplia cobertura a lo largo de diferentes ciclos operativos y condiciones.

La tasa de muestreo de los datos fue de 30 segundos. Esta frecuencia de recopilación de datos ha sido elegida para capturar con precisión las variaciones operativas de los equipos, mientras se mantiene un equilibrio entre la resolución temporal y la manejabilidad del volumen de datos. Esta densidad de datos es esencial para detectar patrones sutiles y transitorios, que son indicativos de posibles anomalías o cambios en el estado de los equipos.

5.2.2. Naturaleza de los Datos y Registro de Fallas

Una particularidad importante de este conjunto de datos es que no está etiquetado; es decir, no se tiene una clasificación previa de los puntos de datos como 'normales' o 'anómalos'. No obstante, se recopiló un registro de mantenimiento que documenta las detenciones y fallas de los equipos, proporcionando una referencia valiosa para la verificación de las anomalías detectadas por los modelos.

A pesar de la falta de etiquetas, el registro de mantenimiento nos permite establecer un marco de trabajo para el modelo de detección de anomalías. Utilizando las fechas y horas documentadas de las detenciones y fallas, es posible correlacionar los eventos registrados con las anomalías identificadas por el modelo, permitiendo una validación indirecta del desempeño del mismo.

Este enfoque no solo aporta una base para la evaluación de los modelos en un entorno no supervisado, sino que también refleja un escenario común en la industria, donde a menudo la data de fallas es escasa o no está disponible.

5.2.3. Análisis Exploratorio de Datos (EDA)

El Análisis Exploratorio de Datos (EDA) representa una fase crucial, ya que ofrece las primeras percepciones y entendimientos del conjunto de datos recopilado de la planta de ósmosis inversa. A través del EDA, buscamos descubrir patrones, identificar anomalías, probar hipótesis y verificar suposiciones con la ayuda de estadísticas descriptivas y visualizaciones gráficas.

5.2.3.1. Bomba Booster

Se comenzó con un resumen estadístico detallado de los sensores monitorizados en las bombas booster (B.2). Al observar las medidas de tendencia central y dispersión (media y desviación estándar), se puede apreciar la variabilidad en las condiciones operativas de estos equipos. Por ejemplo, la temperatura del rodamiento del lado de carga muestra una media de 32.60°C con una desviación estándar de 1.82°C, lo que indica una operación consistente con algunas fluctuaciones. Del mismo modo, la presión de entrada y la velocidad del eje de la bomba booster muestran variaciones significativas, lo que podría ser indicativo de los cambios

operativos experimentados por estas bombas.

Por otra parte, se obtuvieron histogramas y gráficos tipo boxplot para una visualización clara de la distribución de los datos para cada sensor. En base a las gráficas tipo boxplot(B.1) los principales hallazgos fueron:

- **Temperatura de Rodamiento (Lado de Carga y Lado Libre):** Se observa que la temperatura del lado libre tiende a ser más alta que la del lado de carga, lo cual es consistente a lo largo de los datos, con una mediana claramente más elevada. La presencia de valores atípicos sugiere fluctuaciones ocasionales que podrían suponer sobrecalentamientos y requerir de un seguimiento detallado.
- **Vibraciones (Cojinete de Empuje y Radial):** Las vibraciones presentan una distribución con una mediana centrada y pocos valores atípicos, indicando un funcionamiento generalmente estable. No obstante, la vibración radial muestra una mayor dispersión en comparación con la vibración en el cojinete de empuje, lo que podría señalar una mayor variabilidad en ese componente.
- **Presión de Entrada, Velocidad Eje y Caudal Rechazo:** Estas variables muestran un rango de valores concentrados alrededor de un punto central con valores atípicos cercanos a cero lo que sugiere que estos puntos puedan ser producto de lecturas de sensores cuando el equipo se encuentra detenido.

Los histogramas complementan el análisis de los boxplots al ofrecer una vista más detallada de la forma de la distribución de los datos:

- **Presión de Entrada (B.7):** El histograma muestra una distribución bimodal, lo que indica dos modos operativos predominantes para la presión de entrada de las bombas booster. Esto podría estar relacionado con diferentes estados o configuraciones de operación de las bombas.
- **Caudal Rechazo (B.9):** El histograma del caudal de rechazo muestra una distribución con un pick muy pronunciado cerca de cero y varios picks considerables alrededor del valor 1000. Esto sugiere que hay momentos en los que el caudal de rechazo es mínimo o inexistente, pudiendo ser detenciones del equipo, mientras que el resto de picks indican diferentes modos operativos donde el caudal se mantiene en torno a un valor elevado.
- **Temperatura Cojinete Empuje (B.2):** La distribución de la temperatura del cojinete de empuje muestra una tendencia central con una variabilidad moderada, indicando que la temperatura se mantiene relativamente estable durante la operación. Sin embargo, hay un pequeño número de valores que se extienden hacia temperaturas más altas.
- **Temperatura Rodamientos (B.3B.4):** Ambos histogramas de la temperatura de los rodamientos muestran distribuciones similares con una tendencia central y una cola que se extiende hacia valores más altos. Los valores extremos en las colas podrían ser indicativos de sobrecalentamientos ocasionales o anomalías en el sistema de enfriamiento de las bombas.
- **Velocidad del Eje (B.8):** La velocidad del eje presenta una distribución multimodal con varios picks, lo que sugiere que la bomba opera en múltiples estados de velocidad,

aunque predominana mayoritariamente dos valores. Esta variabilidad podría estar directamente relacionada con las necesidades operativas de la planta y la variación de la demanda de flujo de agua.

En base a la exploración univariable de cada sensor se puede notar que existen claras correlaciones entre diferentes variables. Para visualizar esto, se construyó una matriz de correlación (5.4) para identificar y visualizar la relación lineal entre las variables. Al observar la matriz se pueden extraer los siguientes hallazgos:

- **Temperaturas de Rodamiento (Lado de Carga y Lado Libre) y Cojinete de Empuje:** Hay una fuerte correlación positiva entre las temperaturas del rodamiento y el cojinete de empuje, lo que sugiere que estas variables tienden a aumentar y disminuir conjuntamente. Esto es lógico, ya que están físicamente relacionadas en la bomba y es probable que las condiciones que afectan a una afecten también a la otra.
- **Vibración del Cojinete de Empuje y Radial:** Las vibraciones presentan una correlación moderada entre ellas, lo que indica que mientras hay una relación entre las vibraciones en diferentes partes de la bomba, no es tan fuerte como en el caso de las temperaturas.
- **Presión de Entrada, Velocidad del Eje, Caudal de Rechazo y Setpoint:** Estas variables muestran una fuerte correlación entre sí, lo que indica que cuando la bomba opera a una mayor velocidad de eje, hay una tendencia a tener una mayor presión de entrada y caudal de rechazo. Esto puede reflejar el comportamiento esperado del sistema de bombeo, donde un aumento en la velocidad del eje se asocia con un aumento en el caudal y la presión.
- Las correlaciones entre temperaturas y vibraciones son más bajas, lo que sugiere que los cambios en la temperatura no siempre están directamente relacionados con cambios en la vibración. Esto podría deberse a que los factores que causan el cambio de temperatura no necesariamente afectan la vibración y viceversa.

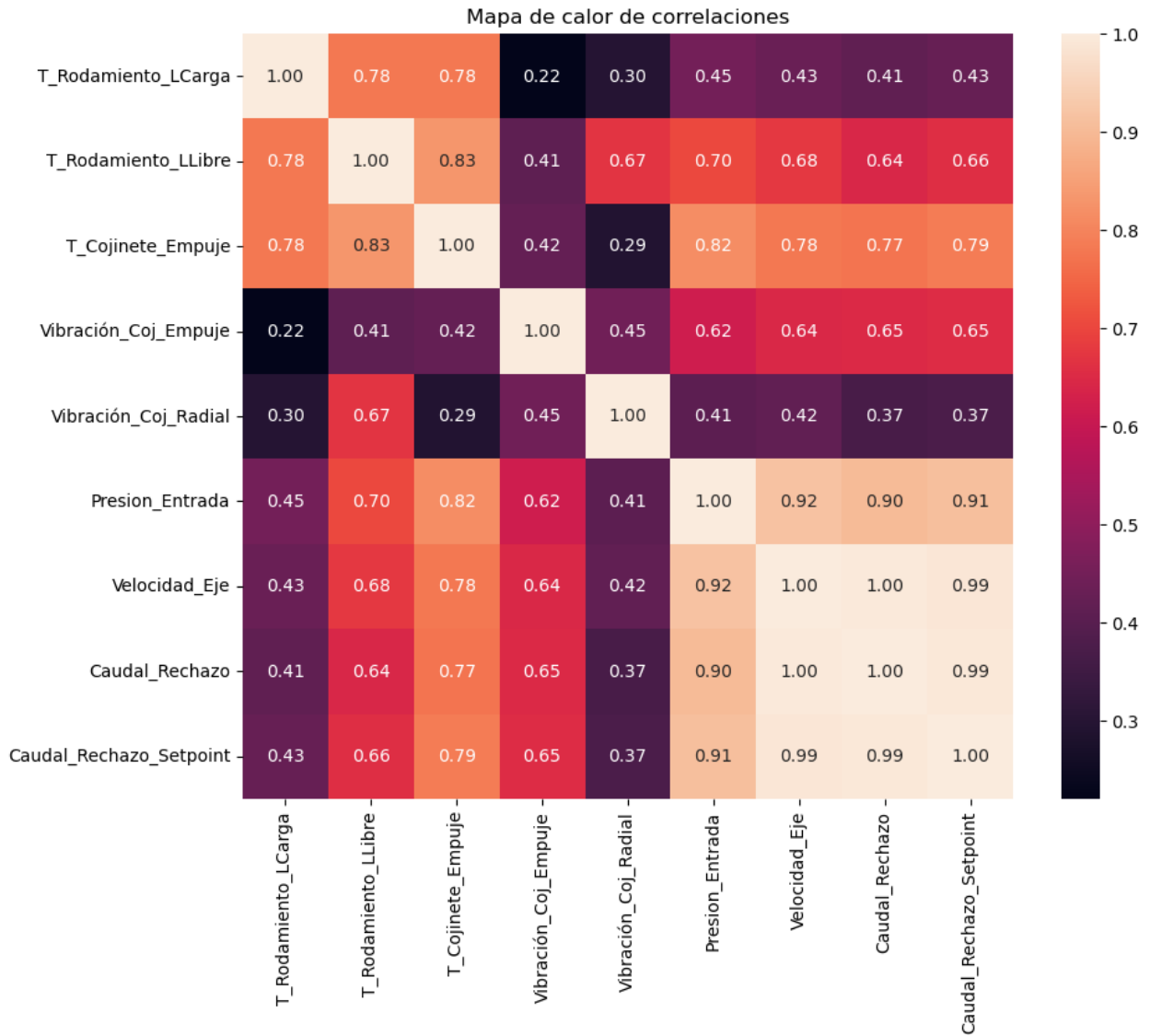


Figura 5.4: Matriz de Correlación Variables Bba Booster

Las fuertes correlaciones observadas en la matriz pueden ser indicativas de comportamientos normales del sistema y deben ser consideradas al construir modelos de detección de anomalías para evitar falsas alarmas cuando estos patrones de correlación se presenten. Por otro lado, las relaciones débiles pueden ser exploradas para detectar anomalías que rompan estas correlaciones típicas.

El análisis de clustering es una técnica de aprendizaje automático no supervisado que se utiliza para agrupar datos similares en conjuntos o 'clusters'. Este método es particularmente útil en la detección de anomalías para identificar diferentes agrupaciones naturales dentro de los datos operativos que representan diferentes modos de operación, como operación normal, condiciones de estrés o incluso fallas iniciales.

Para explorar la posibilidad de distinguir diferentes modos de operación, se llevó a cabo un análisis de clusters. Este proceso comenzó con el entrenamiento de modelos K-means utilizando el método del codo. Este método busca el punto en el que agregar otro cluster no ofrece

un modelo significativamente mejor, lo cual se visualiza buscando un cambio brusco o 'codo' en el gráfico de error (5.5). Este punto representa un equilibrio entre la máxima compresión de datos con el mínimo de clusters y la máxima exactitud al asignar cada dato en su propio cluster. Se evaluó la suma de errores cuadráticos para cada número de clusters y se observó que el error disminuye significativamente hasta llegar a tres clusters, punto después del cual se observa el cambio brusco. Este 'codo' en la curva indica que tres es el número óptimo de clusters para este conjunto de datos, proporcionando un equilibrio entre la compacidad de los clusters y la cantidad de los mismos.

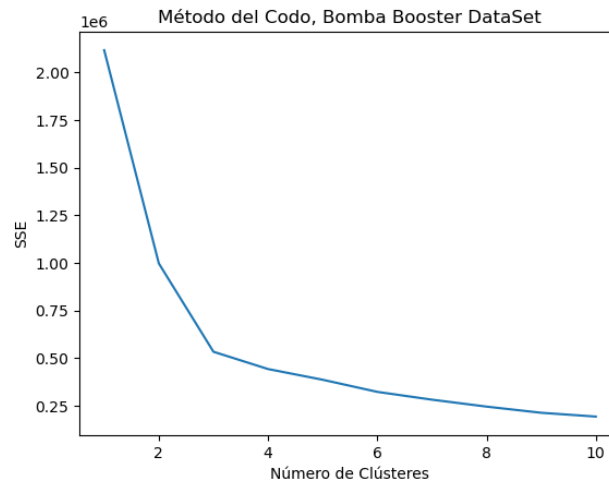


Figura 5.5: Gráfica de Error Método del Codo Bba Booster

Con tres clusters definidos, se procedió a realizar un análisis de componentes principales (PCA) con el objetivo de reducir la dimensionalidad de la data. De las 9 dimensiones originales, se extrajeron las 3 componentes principales, lo que permitió visualizar la segmentación en un espacio tridimensional (5.6). El resultado fue una clara separación de los datos en tres grupos distintos.

Además del PCA, se aplicó el análisis UMAP (Uniform Manifold Approximation and Projection) como otra técnica de reducción de dimensionalidad para comparar los resultados. Aunque UMAP (5.7) también identificó los tres clusters, estos se presentaron algo más juntos en comparación con los resultados del PCA. Esta proximidad en la representación de UMAP podría sugerir una mayor densidad o solapamiento en las características de los datos, aunque los grupos seguían siendo distinguibles.

Este análisis de clusters, UMAP y PCA ofrece una nueva perspectiva sobre los datos operativos. Cada cluster puede representar un modo operativo distinto de las bombas, como operación estándar, transición o incluso condiciones anómalas. La visualización en tres dimensiones con PCA no solo confirma la viabilidad de diferenciar estos estados, sino que también proporciona una herramienta valiosa para la interpretación visual.

Visualización en 3D con PCA

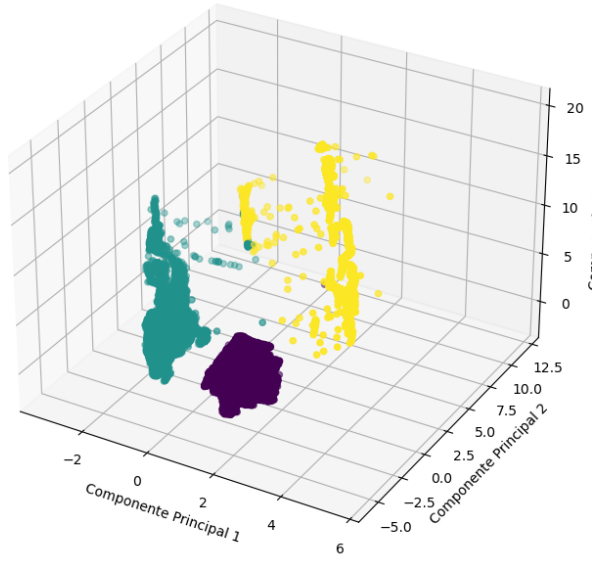


Figura 5.6: Análisis PCA Bba Booster

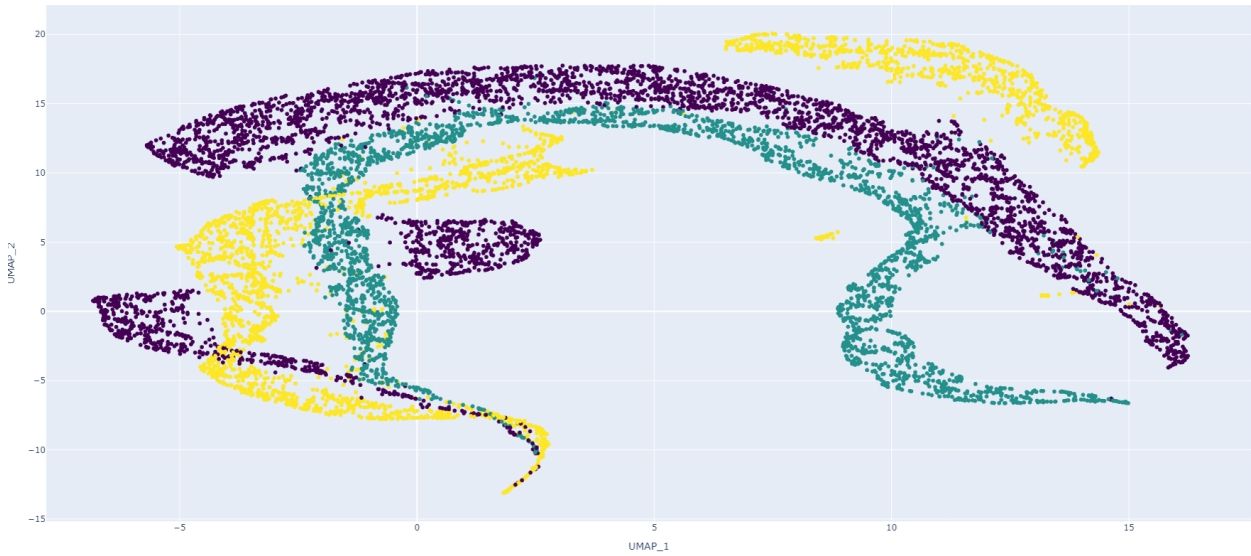


Figura 5.7: Análisis UMAP Bba Booster

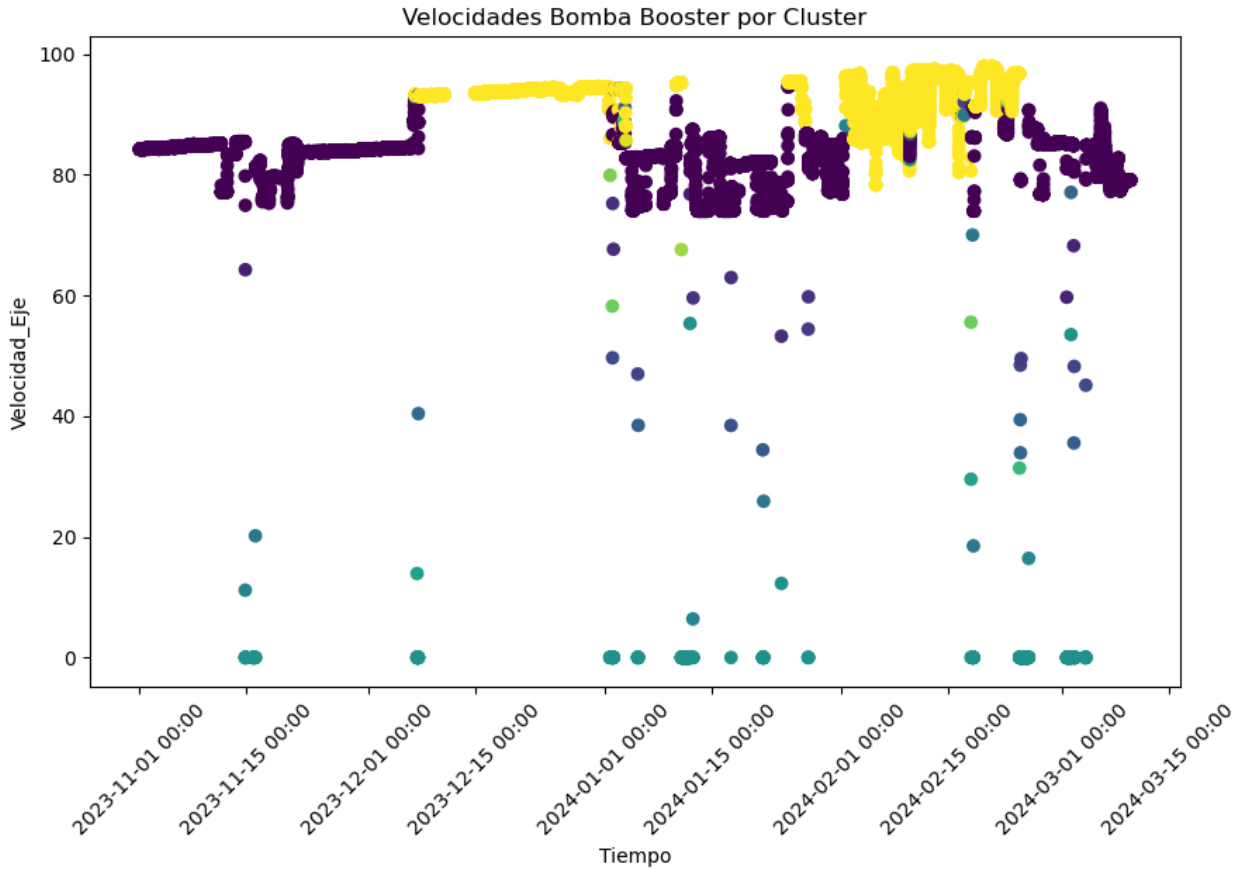


Figura 5.8: Velocidad Eje Bba Booster separado por cluster

El gráfico ilustrado en la Figura 5.8 proporciona una visualización detallada de los clusters identificados para las velocidades de la bomba booster. A través de la representación de los datos, es posible inferir posibles modos de operación correspondientes a cada cluster. De acuerdo con el gráfico, el cluster de color verde se asocia predominantemente con los períodos en los que el equipo está detenido, lo que sugiere una condición de inactividad o mantenimiento. Por otro lado, el cluster morado muestra un rango de velocidad del motor consistentemente entre el 80 % y el 85 %, lo que podría interpretarse como el modo de operación estándar del equipo. Finalmente, el cluster amarillo se caracteriza por velocidades de operación más altas, en el rango del 85 % al 90 %, lo cual podría indicar una condición de trabajo más exigente o un estado operacional acelerado. Estos insights destacan la relevancia del análisis de clustering para discernir y monitorizar los diferentes estados operativos de las bombas, lo que puede ser crucial para la implementación de estrategias de mantenimiento preventivo y la optimización del rendimiento de la maquinaria.

5.2.4. Preprocesamiento de los datos

El preprocesamiento de los datos se realizó con el objetivo de asegurar que el conjunto de datos estuviera en un formato adecuado para la implementación de los modelos de deep learning. Los pasos seguidos fueron los siguientes:

- **Conversión a Formato Numérico:** Todos los valores que estaban en formato de texto, especialmente aquellos marcados como 'no data available', fueron transformados

a un formato numérico. Durante este proceso, cualquier valor que no se pudo convertir correctamente fue asignado como NaN (Not a Number) para su posterior manejo.

- **Limpieza de Valores Faltantes:** Se procedió a eliminar todas las instancias con valores NaN, ya que estos representan la ausencia de datos y podrían sesgar los resultados del análisis.
- **Eliminación de Datos Durante Detenciones de Equipos:** Utilizando los insights obtenidos del EDA, se identificaron y eliminaron los períodos en los que los equipos estaban detenidos. Estos intervalos se caracterizan por tener valores anormalmente bajos y/o faltantes en las lecturas de los sensores, los cuales no son representativos del comportamiento normal u operativo de los equipos.
- **Filtrado de Datos Transientes:** Se desarrolló una función para eliminar los datos transientes asociados con la partida y la detención de los equipos. Estos periodos transitorios pueden incluir fluctuaciones atípicas en los datos que no son indicativas de condiciones operativas estables o de fallos.

5.2.5. Arquitectura del modelo implementado

A partir de las evaluaciones comparativas realizadas utilizando la base de datos SKAB, que está debidamente etiquetada, se identificó que los modelos basados en Deep Learning, y más específicamente los autoencoders recurrentes (Long Short-Term Memory, GRU), exhiben un rendimiento superior en la tarea de detección de anomalías. Este hallazgo es consistente con la literatura existente que destaca la eficacia de los modelos de secuencias recurrentes para capturar dependencias temporales complejas en series de tiempo, lo que es crucial para identificar comportamientos anómalos en datos de sensores industriales.

Con base en estos resultados preliminares, se decidió implementar este tipo de arquitectura para el análisis de los datos de bombas de la planta de ósmosis inversa. La arquitectura seleccionada consiste en una capa de entrada que recibe los datos de entrada para su procesamiento. Seguidamente, una capa densa con activación ReLU prepara la entrada para la codificación por una red LSTM. Este es el núcleo de la arquitectura que actúa como el espacio latente del autoencoder, capturando la esencia de la secuencia de entrada en una representación de menor dimensión. Finalmente, otra capa LSTM y una capa densa reconstruyen las señales original a partir de esta representación latente. El tamaño de las redes viene dado por la dimensión del espacio latente.

El modelo utiliza una función de pérdida de error cuadrático medio y se optimiza utilizando el algoritmo Adam. Luego, en base al error se calcula un KPI, el cual es un valor entre 0 y 1 que refleja la anomalía de los datos. Este KPI se calcula en función de un coeficiente de pendiente (slope) y un nivel de contaminación predefinido, los cuales determinan la sensibilidad del modelo a las anomalías.

Hiperparámetros LSTM Autoencoder Bomba Booster

Tras probar varias configuraciones de hiper parámetros se definieron los siguientes como los que minimizan el error de reconstrucción en los datos de entrenamiento y validación.

- **Slope:** 0.4, que regula la sensibilidad del KPI ante desviaciones en la reconstrucción.
- **Contaminación:** 0.01, valor estimado de proporción de puntos anómalos en el conjunto de datos
- **Overlap:** 0, lo que significa que las secuencias no se solapan. Esto para lograr la visualización de los reconstrucciones.
- **Épocas:** 50
- **Tamaño del lote (batch size):** 256
- **Ventana temporal (time size):** 10, definiendo la cantidad de pasos temporales en cada secuencia de entrada.
- **Feature size:** 9, número de sensores con data de entrada.
- **Tamaño de la capa latente (lat size):** 4
- **Tasa de aprendizaje (learning rate):** 0.01, para controlar la velocidad de actualización de los pesos de la red durante el entrenamiento.

5.2.6. Resultados Modelos

En esta sección, se presentan los resultados de aplicar los modelos LSTM autoencoder en la base de datos de la planta de osmosis inversa. Los modelos se entrenaron y validaron utilizando varias configuraciones de hiperparámetros, evaluando y seleccionando aquellas que resultaron en una disminución óptima del error de reconstrucción en los conjuntos de entrenamiento y validación.

Se realizó un exhaustivo análisis de las métricas de desempeño de los modelos, centrándose en la precisión de la reconstrucción y la estabilidad del modelo durante el entrenamiento. Se observó que las configuraciones de hiperparámetros que minimizaban el error de reconstrucción proporcionaban los mejores resultados en términos de capacidad del modelo para aprender y representar adecuadamente los patrones en los datos.

Posteriormente, se llevó a cabo la evaluación de los modelos utilizando eventos de fallas conocidas en 12 bombas booster y 12 bombas de alta presión. Se probó la capacidad de los modelos para detectar anomalías y predecir fallas con anticipación, registrando el tiempo de anticipación ante una falla. Las alertas generadas por los modelos de IA se recopilaron y se compararon con la información en terreno para determinar correctamente si correspondían a verdaderos positivos o falsos positivos. Con base en esta evaluación, se generó una matriz de confusión y se calcularon diversas métricas de desempeño, incluyendo la sensibilidad y precisión.

5.2.6.1. Casos de éxito

A continuación, se presentarán algunos casos de éxito en la detección temprana de anomalías por parte de los modelos LSTM autoencoder (5.10, 5.11, 5.12). Estos casos destacan situaciones en las que los modelos lograron identificar con precisión eventos anómalos, que

posteriormente desencadenaron en una falla. Asimismo, se presentan los casos en los que se generaron falsos positivos (5.14, 5.15), es decir, alertas incorrectas de anomalías, y se analizarán las posibles causas de estos errores.

Los gráficos presentados son una representación visual de los modelos de inteligencia artificial aplicados. La leyenda del gráfico (5.9) distingue tres niveles de indicadores de desempeño (KPI) mediante un código de colores: los puntos verdes representan un estado normal con un KPI de 0 a 0.4, indicando un funcionamiento sin anomalías. Los puntos amarillos señalan un estado insatisfactorio con anomalía leve, con valores de KPI entre 0.4 y 0.7, lo que sugiere la necesidad de supervisión o mantenimiento no urgente. Los puntos rojos, que denotan un KPI superior a 0.7, alertan sobre un estado inaceptable con anomalías severas, lo cual requiere atención inmediata para prevenir una falla inminente o mitigar una condición crítica.

Las franjas en el gráfico aportan información adicional sobre la operatividad de los equipos: las franjas de color café simbolizan detenciones no programadas, reflejando interrupciones inesperadas en el funcionamiento, mientras que las franjas celestes representan detenciones programadas, las cuales corresponden a paradas planificadas para mantenimiento o inspecciones regulares. La intersección de puntos rojos con las franjas café puede indicar una correlación entre las anomalías detectadas por los modelos de IA y las detenciones no programadas resultantes, destacando la utilidad del sistema en la identificación oportuna de problemas que podrían llevar a paros no deseados.

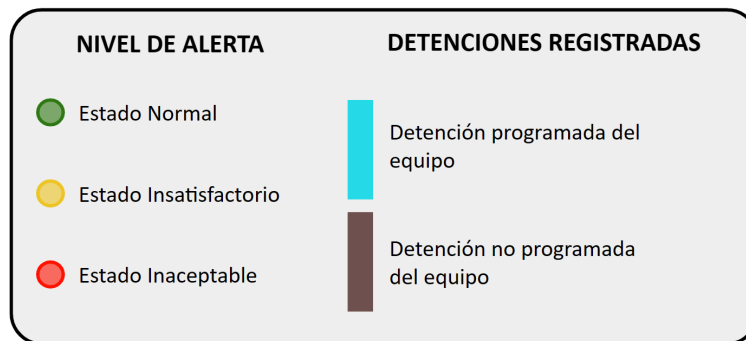


Figura 5.9: Leyenda para gráficos de KPI

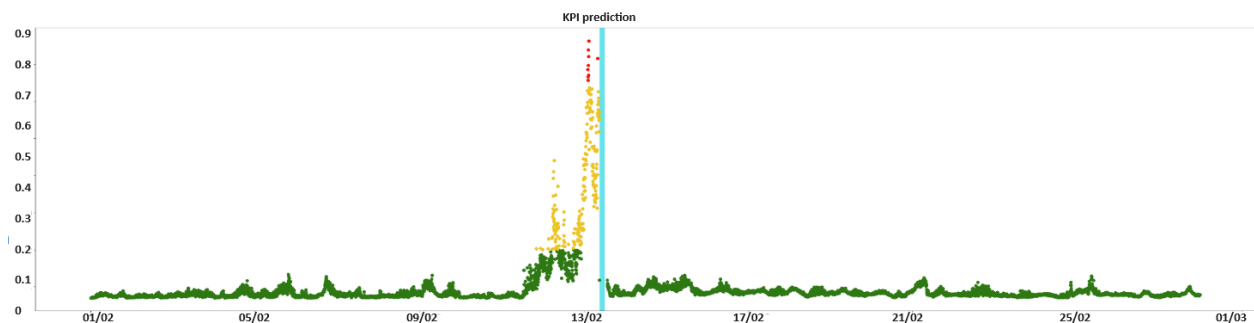


Figura 5.10: Caso de éxito bomba booster. Descripción Falla: Filtración por la línea de refrigeración de la booster Rack 11. Se logra una anticipación de 2 días.

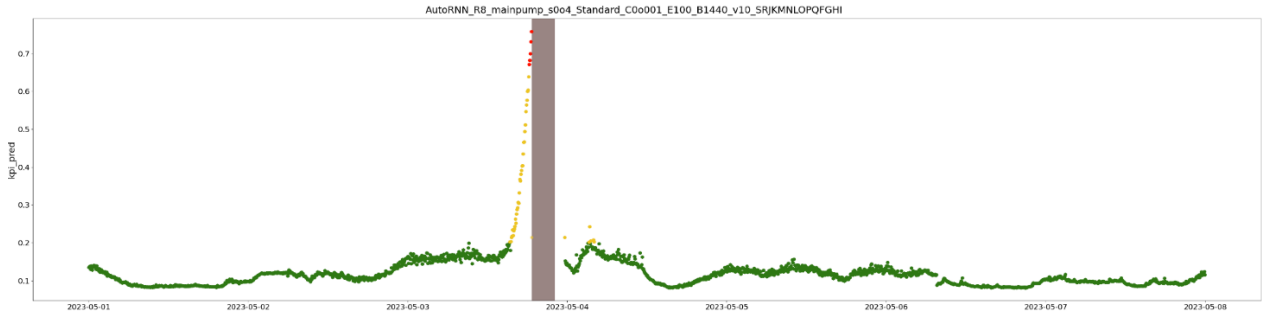


Figura 5.11: Caso de éxito bomba HP Rack 08. Descripción Falla: Alta presión de descarga bomba HP. Anticipación de 3 horas y 25 minutos.

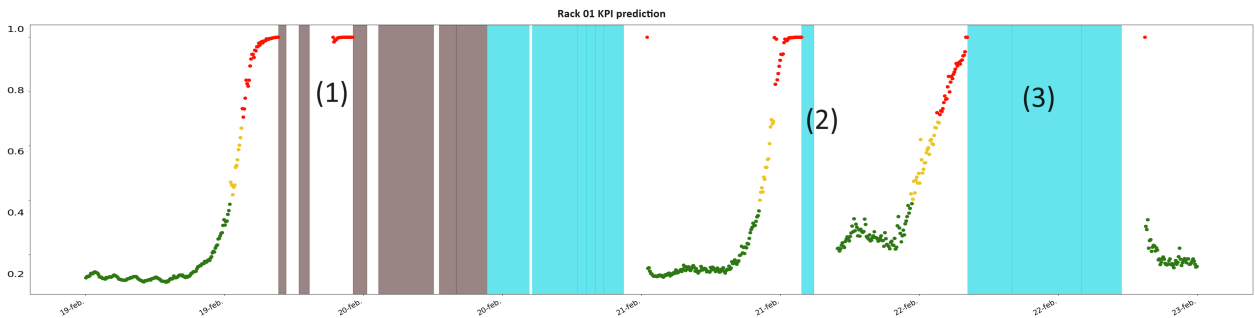
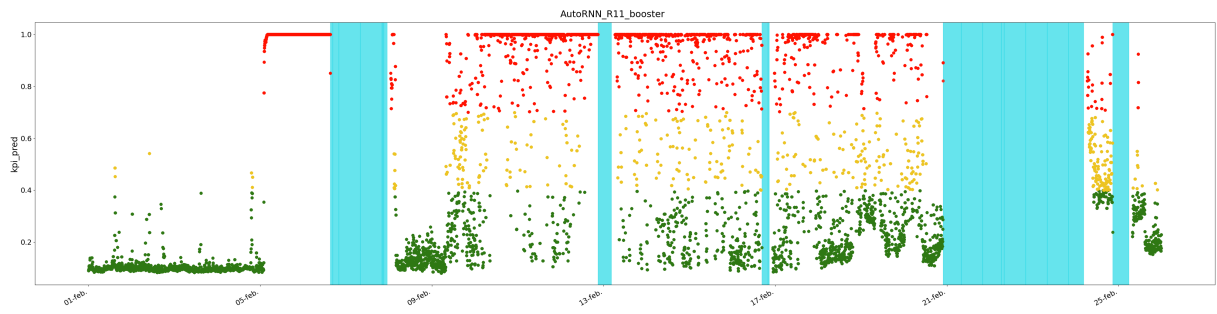


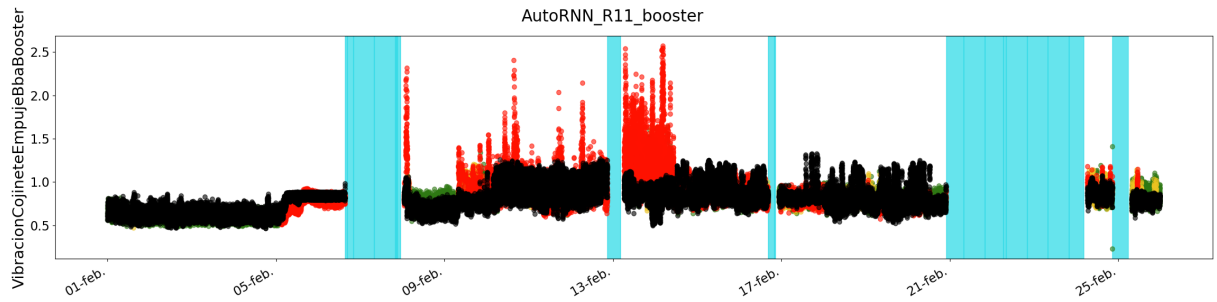
Figura 5.12: Caso de éxito bomba HP Rack 01. Descripción Fallas: (1) Rotación Inversa (2) Filtración válvula check (3) Alta presión de descarga . Anticipaciones (1) 4 hrs 10 minutos (2) 4 hrs 24 minutos (3) 3 hrs 46 minutos

El caso de éxito ilustrado en la Figura 4.7 destaca la eficacia del modelo LSTM autoencoder aplicado a la bomba booster en el Rack 11. En este ejemplo se logra una detección de una filtración en la línea de refrigeración, ya que el modelo logró anticipar la falla con 2 días de adelanto. Esta anticipación permitiría realizar las intervenciones necesarias para prevenir daños mayores y asegurar la continuidad operacional. En la Figura 4.8, se aprecia la utilidad del modelo en la detección temprana de un incremento en la presión de descarga en la bomba HP del Rack 08. Con una anticipación de 3 horas y 25 minutos. La Figura 4.9 presenta 3 fallas donde el modelo predijo anomalías con éxito previamente para cada caso en la bomba HP del Rack 01: rotación inversa, filtración en la válvula check y alta presión de descarga, con anticipaciones respectivas de 4 horas 10 minutos, 4 horas 24 minutos y 3 horas 46 minutos. Estos resultados demuestran la capacidad del modelo para adaptarse y responder a múltiples escenarios de falla, subrayando su versatilidad y eficacia. La detección temprana de estas anomalías evidencia el valor añadido del modelo en la reducción de tiempos de inactividad y en la promoción de una operación más segura y eficiente.

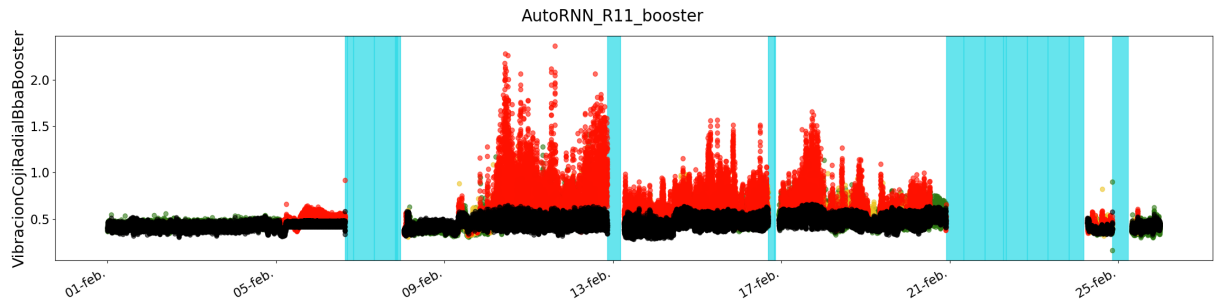
El último caso de éxito, ilustrado en la Figura 4.10, destaca una situación en donde en el equipo se realiza una detección el día 06/02/2024 a las 15:14 por filtraciones. Durante esta detección, se observó una contaminación de aceite con agua en los descansos del motor, lo que provocó una notable bajada de temperatura. Situación que el modelo identifica como anomalías severas (5.13.d) a partir de el 05/02/2024 a las 02:05, generando una anticipación de 1 día y 11 horas.



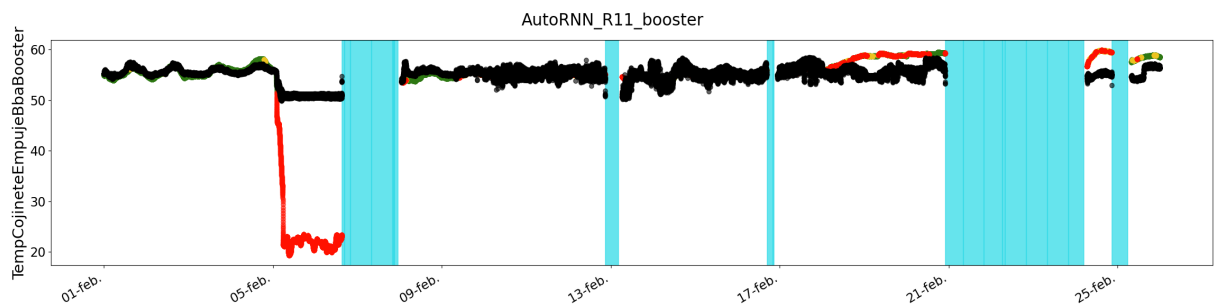
(a) KPI



(b) Vibración Cojinete Empuje Bba Booster



(c) Vibración Cojinete Radial Bba Booster



(d) Temperatura Cojinete Empuje Bba Booster

Figura 5.13: Las imágenes muestran la representación gráfica de los datos de sensores y las predicciones realizadas por los modelos de inteligencia artificial para la bomba booster del rack 11. En este gráfico, los puntos negros representan las predicciones efectuadas por los modelos de IA. Por otro lado, los puntos de colores reflejan los valores reales registrados por los sensores: los puntos rojos indican una anomalía severa, los amarillos señalan una anomalía leve, y los verdes representan un comportamiento normal del equipo. Además, las franjas celestes en el gráfico marcan los periodos de detención del equipo, proporcionando un contexto visual claro sobre los momentos en los que la maquinaria estuvo fuera de servicio.

Tras las reparaciones necesarias, que implicaron un tiempo de detención y pérdidas de producción, la bomba fue puesta nuevamente en funcionamiento. Sin embargo, el modelo continuó señalando anomalías severas, esta vez asociadas a un incremento notable de las vibraciones en los descansos en las direcciones radial y de empuje (5.13.b, 5.13.c), detectadas tras la reanudación operativa. Esta circunstancia condujo a tres detenciones de mantenimiento adicionales, hasta descubrir que la bomba no había sido correctamente alineada en la intervención previa.

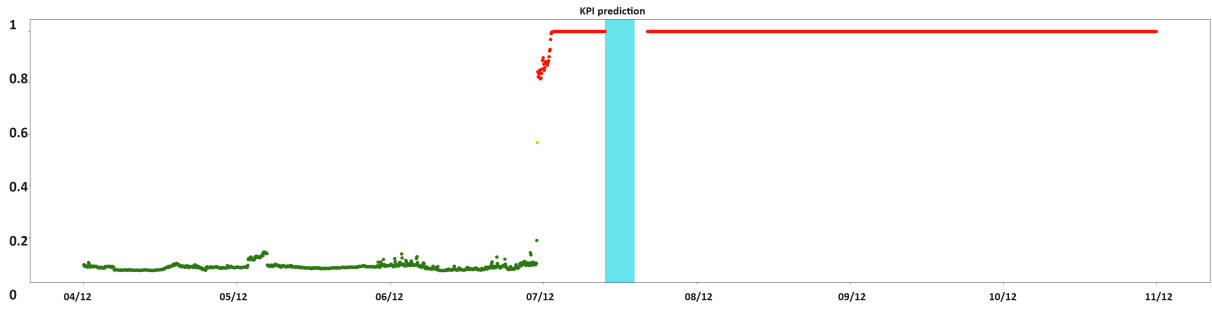
Este caso resalta la importancia crítica de la tecnología de deep learning en la detección anticipada de fallas, permitiendo generar alertas tempranas que facilitan intervenciones oportunas, minimizando así los tiempos de detención y mejorando la eficiencia en los procesos de mantenimiento. La capacidad del modelo para detectar y alertar sobre variaciones sutiles pero significativas en el comportamiento operativo de la maquinaria es una herramienta invaluable para la optimización de la disponibilidad y fiabilidad de los equipos industriales.

5.2.6.2. Falsos positivos

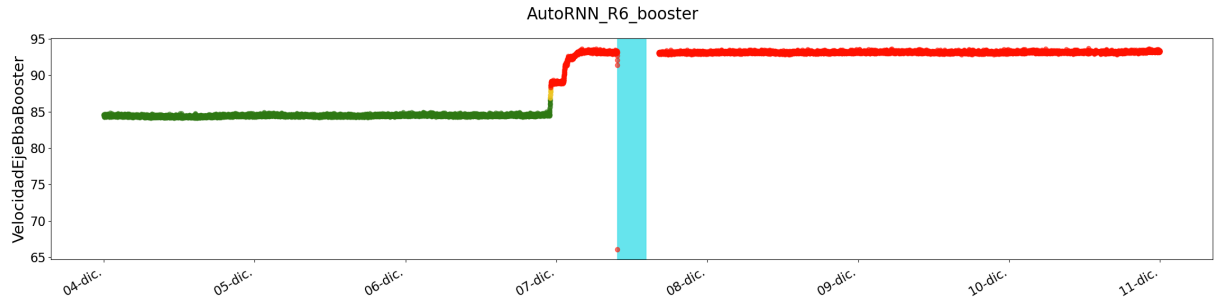
A continuación, se detalla el primer caso de falso positivo, identificado en la bomba booster del Rack 06 (5.14). El modelo de inteligencia artificial comenzó a generar alertas de anomalías severas a partir del 07/12/2023. No obstante, tras un análisis más detallado, se determinó que estas supuestas anomalías no estaban relacionadas con fallas o malfuncionamiento del equipo. En realidad, las alertas fueron desencadenadas por un incremento en la velocidad de operación de la bomba, un ajuste operacional previsto y no una condición adversa. Este incidente subraya la importancia de contextualizar adecuadamente los datos operacionales y las alertas generadas por el modelo, resaltando la necesidad de ajustar y afinar continuamente los algoritmos de IA para distinguir entre cambios operacionales legítimos y verdaderas anomalías que requieran atención.

Los casos aquí presentados se catalogan como falsos positivos debido a que el modelo emite alertas de anomalías cuando, en realidad, el equipo opera normalmente, aunque en un modo operacional distinto. Este fenómeno se observa comúnmente tanto en bombas HP como en bombas booster y se atribuye principalmente a la falta de diversidad en los rangos de operación dentro de la data de entrenamiento. Por ejemplo, si el modelo se entrena exclusivamente con datos correspondientes al motor operando al 80 % de su capacidad, aprende a reconocer este rango como el estándar de funcionamiento normal, interpretando cualquier variación significativa, como un incremento o decremento en la velocidad, como una anomalía.

Para mitigar este problema, es recomendable incorporar en la fase de entrenamiento datos que reflejen varios rangos de operación, permitiendo así que los modelos aprendan a identificar múltiples modos operativos como normales. No obstante, esta solución enfrenta el desafío de que cualquier nuevo modo o velocidad operacional, no representado en el entrenamiento, probablemente inducirá la generación de falsos positivos. Además, resulta complicado abarcar todos los posibles modos operativos durante el entrenamiento, dado que, como ilustra la figura 5.15, estos equipos pueden operar a velocidades que varían ampliamente, desde el 70 % hasta el 99 %, dependiendo de las necesidades operacionales y los requerimientos de producción de agua de la planta.

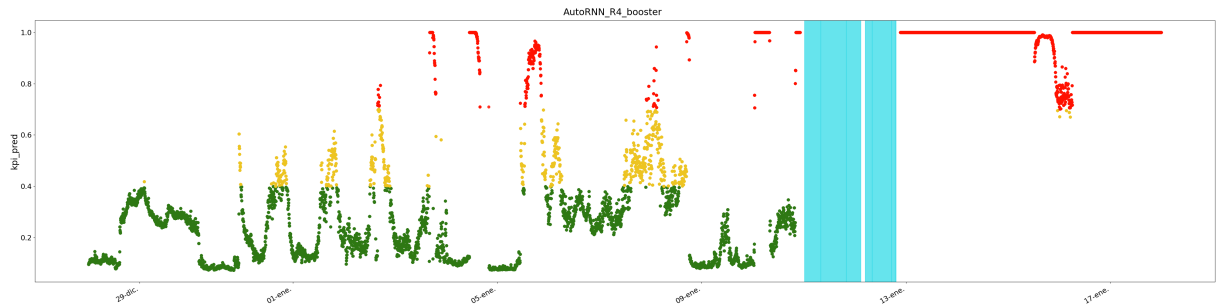


(a) KPI con anomalías durante funcionamiento normal.

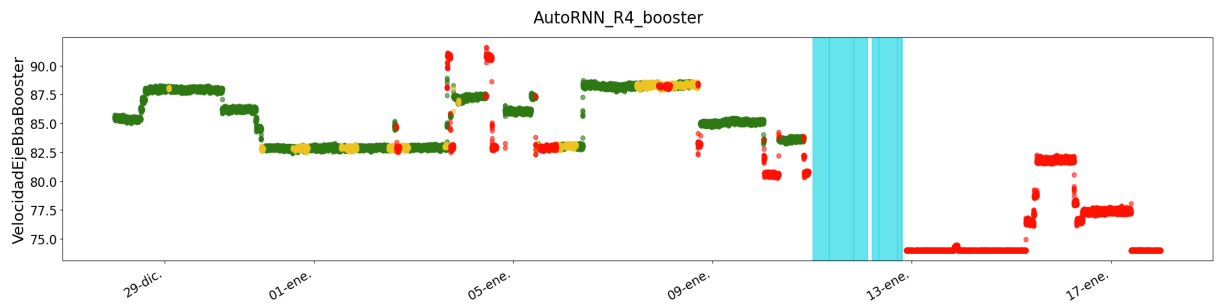


(b) Variable operacional de velocidad del motor

Figura 5.14: Caso de falsa alerta bomba booster Rack 06



(a) KPI con anomalías durante funcionamiento normal.



(b) Variable operacional de velocidad del motor

Figura 5.15: Caso de falsa alerta bomba booster Rack 04

5.2.7. Resumen Métricas

Para una evaluación detallada del desempeño de los modelos de inteligencia artificial implementados, las alertas generadas han sido clasificadas en cuatro categorías distintas, permitiendo una interpretación más granular y contextualizada de los resultados.

- **Alertas de fallas incipientes:** Esta categoría engloba las alertas que lograron anticiparse a situaciones críticas, es decir, fallas o detenciones necesarias para intervenciones preventivas. Estas alertas son indicativas del éxito del modelo en su función predictiva, demostrando su valor en la prevención de tiempos de inactividad no programados y en la facilitación de un mantenimiento más proactivo.
- **Alertas por anomalías temporales:** Refieren a las alertas desencadenadas por eventos anómalos que, aunque confirmados por expertos en terreno, poseen un carácter temporal y no conducen a una intervención inmediata del equipo. Ejemplos de esto incluyen incrementos momentáneos en la vibración que se resuelven sin acción correctiva. Aunque estos eventos no requieren detención, su detección es relevante para el monitoreo y seguimiento continuo.
- **Alertas falsas por cambios de operación:** Estas alertas se originan cuando el modelo interpreta incorrectamente variaciones en el modo operacional como anomalías del equipo. Distinguir estas alertas es crucial para afinar el rendimiento del modelo y reducir interrupciones innecesarias en la producción.
- **Fallas no detectadas:** Corresponden a las ocasiones en que el modelo falló en generar una alerta ante una falla real. Esta categoría es vital para identificar y abordar las limitaciones actuales del modelo, proporcionando áreas específicas para futuras mejoras y ajustes.

Tabla 5.5: Resumen alertas generadas por modelo IA

Alertas	Nº	Días	Horas	Minutos
Fallas Incipientes	30	5 días	15.84	9.6
Anomalías temporales	12	-	-	-
Cambios de Operación	9	-	-	-
Fallas no detectadas	2	-	-	-
TOTAL	53			

Gracias a esta clasificación de alertas, es factible construir una matriz de confusión para tener una visualización clara de las instancias donde el modelo ha tenido éxito en la detección de anomalías y fallas (verdaderos positivos) frente a los casos donde no lo ha hecho (falsos negativos), así como las situaciones de alertas innecesarias (falsos positivos). A partir de esta matriz, se pueden calcular métricas esenciales como la sensibilidad, que mide la capacidad del modelo para identificar correctamente las fallas y anomalías, y la precisión, que evalúa qué tan acertadas son las alertas emitidas.

Tabla 5.6: Matriz de confusión

	Predicción Positiva	Predicción Negativa
Real Positivo	42 (VP)	2 (FN)
Real Negativo	9 (FP)	-

Tabla 5.7: Métricas de sensibilidad y precisión

Métrica	Fórmula	Valor
Sensibilidad (Recall)	$\frac{VP}{VP+FN}$	0.95
Precisión (Precision)	$\frac{VP}{VP+FP}$	0.82

5.2.8. Umbral Adaptativo Basado en la Teoría del Valor Extremo

Una limitación observada en la implementación actual de modelos de detección de anomalías es la generación de alertas falsas desencadenadas por cambios en los modos operativos de los equipos, en lugar de por fallos reales. Estos falsos positivos disminuyen considerablemente la confianza en el sistema de monitoreo automatizado. Para superar este desafío, se propone la implementación de un método de umbral variable, basado en [38] que se adapte de manera inteligente a los patrones cambiantes de operación. La metodología propuesta se desglosa de la siguiente manera:

- **Recolección y Preparación de Datos:** Seleccionamos, recolectando y normalizando los datos operativos para el análisis.
- **Análisis de Valor Extremo (EVT):** Aplicamos EVT para modelar los máximos de los errores de reconstrucción, ajustando estos a una distribución de Valor Extremo Generalizado (GEV).
- **Estimación de Parámetros GEV:** A través del método de máxima verosimilitud, estimamos los parámetros de la distribución GEV.
 - Parámetro de forma (ξ)
 - Parámetro de ubicación (μ)
 - Parámetro de escala (σ)
- **Umbral Adaptativo:** Desarrollamos un umbral adaptativo definiendo la función $g(t)$ que representa la forma de los cambios operativos, probando varias formas funcionales como la logarítmica, la exponencial, la sinusoidal, y la polinomial. Finalmente se seleccionó la forma $g(t) = a \cdot \log(b + t + c)$.
- **Cálculo del Umbral:** Calculamos el umbral en cada instante de tiempo utilizando los parámetros GEV estimados y el nivel de confianza p según:
 - $\mu(t) = \mu$
 - $\sigma(t) = \exp(\beta_0 + \beta_t \cdot g(t))$
 - $\text{Threshold}(t) = \mu(t) - \frac{\sigma(t)}{\xi} \left[1 - \{-\ln(1 - p)\}^{-\xi} \right]$
- **Suavización del Umbral:** Aplicamos un suavizado con promedio móvil para evitar fluctuaciones bruscas y mejorar la estabilidad del umbral. Además se definen dos niveles de confianza, $p = 0.95$ y $p = 0.99$ para determinar si corresponde a una anomalía leve (amarilla) o severa (roja) respectivamente.

5.2.8.1. Comparación de casos con umbral variable

La Figura 5.16 demuestra la aplicación del umbral variable en el Rack 06 (5.14). Aquí se observa cómo el umbral adaptativo, marcado por las líneas amarilla y roja para dos niveles de confianza distintos (95 % y 99 % respectivamente), fluctúa en respuesta a los cambios en el error de reconstrucción (MAE), representado por los puntos azules. Los picos en el MAE que superan el umbral amarillo indican posibles anomalías leves, mientras que aquellos que cruzan el umbral rojo señalan anomalías más severas. El comportamiento del umbral muestra una clara adaptabilidad a los picos de error, ajustándose para reducir falsas alertas y destacar solo los eventos más críticos.

Se puede observar que en el momento del cambio operacional, el error si supera ambos umbrales ya que se trata de un aumento considerable en el vector de error. Sin embargo, al mantenerse estos valores del vector en el tiempo el umbral es capaz de ajustarse automáticamente para dejar de levantar alertas al notar de que se trataba de un cambio operacional.

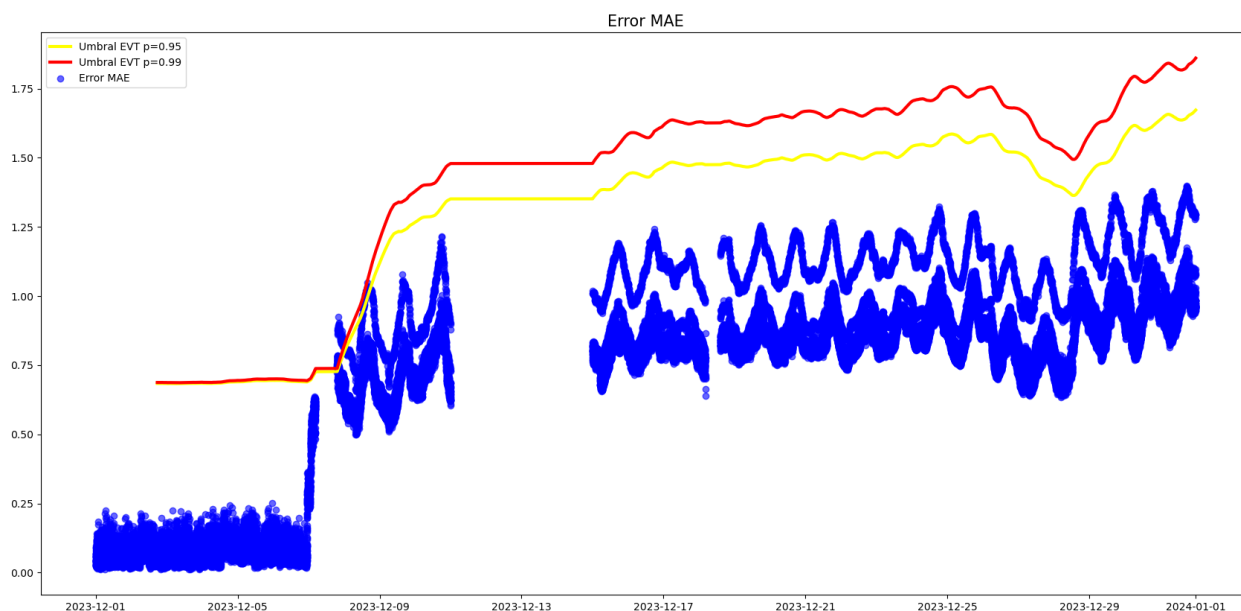


Figura 5.16: Umbral Variable Rack 06

En la Figura 5.17, vemos un comportamiento similar para Rack 04 (5.15). El umbral variable sigue de cerca los cambios en el MAE y se ajusta progresivamente. Este ajuste es particularmente evidente donde el MAE muestra una tendencia al alza, y el umbral se modifica en consecuencia para mantener una detección sensible y específica. La configuración de los umbrales ofrece una distinción clara entre las variaciones normales y las situaciones que requieren atención.

Finalmente, la Figura 5.18 destaca la capacidad del umbral variable, en un caso de anomalías reales y fallas (5.13), para adaptarse a un evento extremo, donde un pico significativo en el MAE es claramente identificado como una anomalía severa. Después del evento, el umbral se adapta rápidamente, demostrando su flexibilidad y la eficiencia del sistema para volver a la normalidad después de detectar condiciones críticas.

Estas visualizaciones confirman la robustez del umbral variable en diferentes situaciones, enfatizando su importancia para detectar y señalar anomalías de manera precisa, minimizando al mismo tiempo los falsos positivos y la posibilidad de interrupciones innecesarias en la producción.

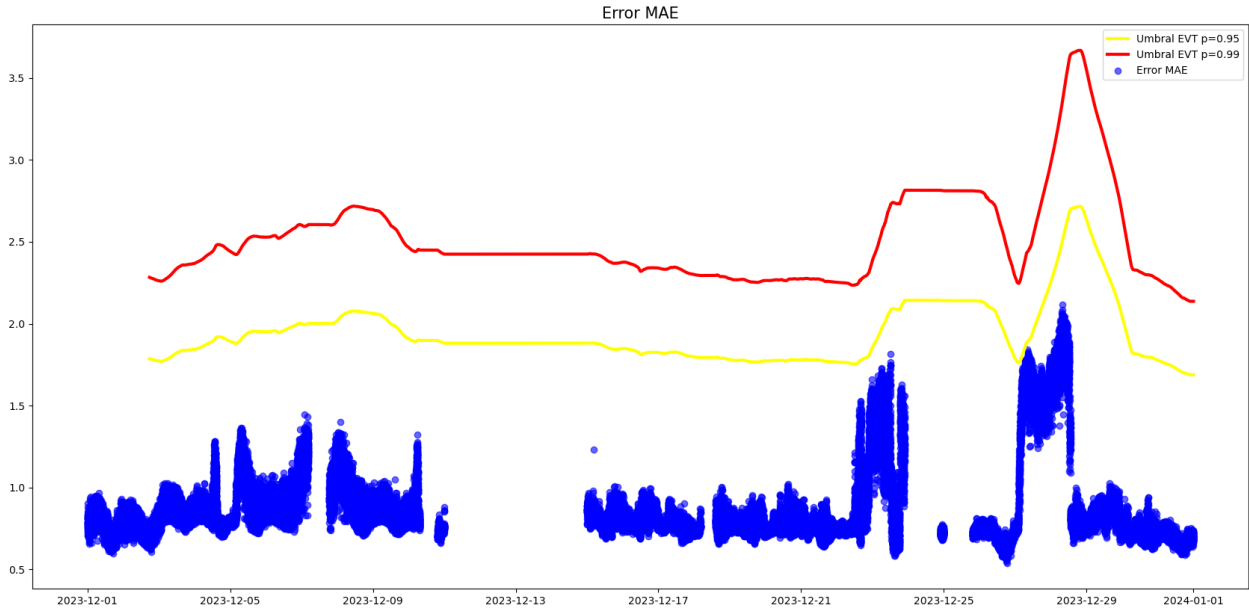


Figura 5.17: Umbral Variable Rack 04

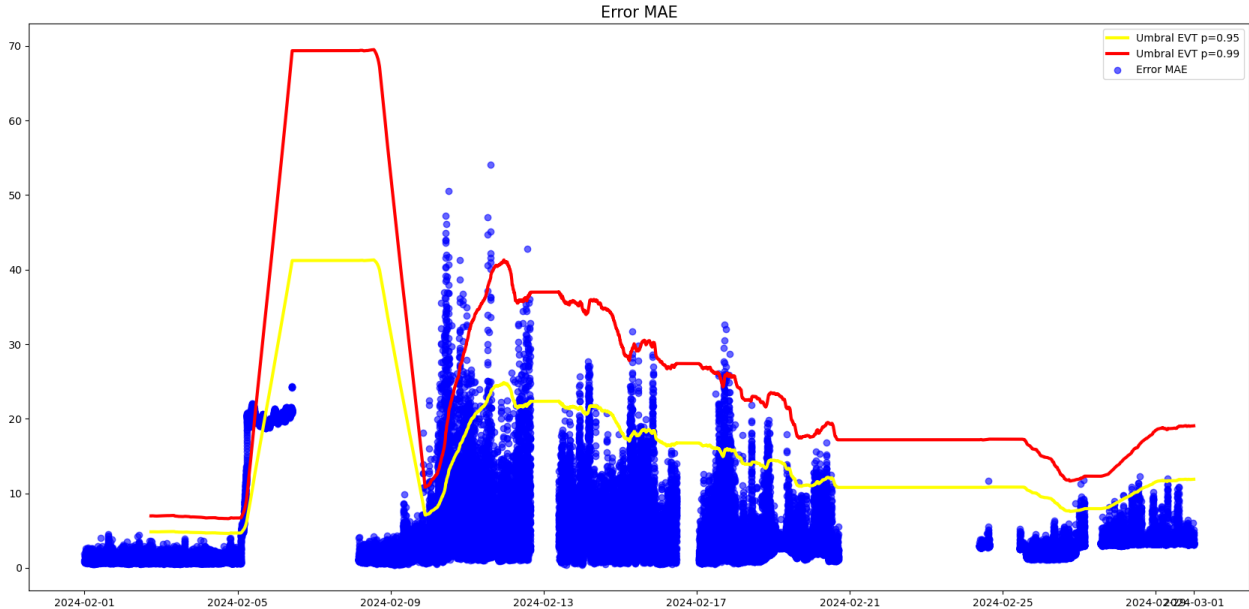


Figura 5.18: Umbral Variable Rack 11

5.2.8.2. Comparación de métricas con y sin umbral variable

Una de las mejoras más significativas observadas con la implementación del umbral variable concierne a la duración de las alertas generadas. Aunque el número total de alertas se mantuvo relativamente constante (de 9 falsos positivos se redujeron a 7) con y sin la implementación del umbral variable, se notó una reducción sustancial en la duración de las alertas, especialmente aquellas relacionadas con cambios operacionales. Con el umbral variable, estas alertas extendidas se reducen notablemente, limitándose a no más de 24 horas en la mayoría de los casos. Este ajuste dinámico es claramente evidente en las visualizaciones proporcionadas, donde el umbral responde y se adapta al vector de error. Cuando el error aumenta y se cruza el umbral, se activa la alerta correspondiente, pero gracias a la capacidad de adaptación del umbral, la condición de alerta no persiste indebidamente. Por contraste, en ausencia del umbral variable, se observó que las alertas podrían mantenerse activas durante periodos prolongados, abarcando varios días o incluso semanas, lo que potencialmente conduciría a interrupciones operativas innecesarias y a una respuesta de mantenimiento menos eficiente.

La implementación del umbral variable ha tenido un impacto significativo en la duración de las alertas por cambios de operaciones de nuestro sistema de detección de anomalías. Para cuantificar esta mejora, se compararon las métricas de rendimiento, específicamente la cantidad de falsos positivos, antes y después de la introducción del umbral variable. La Tabla 5.8 presenta esta comparativa.

Tabla 5.8: Comparación de métricas de rendimiento con y sin el uso del umbral variable.

Métrica	Sin Umbral Variable	Con Umbral Variable
Falsos Positivos	9	7
Duración promedio por alarma	14 días	24 hrs.
Precisión	0.82	0.85

Este análisis indica claramente que la adopción de umbrales variables disminuye sustancialmente la duración de falsas alertas y mejora la precisión del sistema de detección de anomalías. Al ajustar los umbrales de alerta en tiempo real, los modelos son capaces de discriminar con mayor efectividad si un aumento en el vector de error se debe a cambios operacionales o a comportamientos verdaderamente anómalos.

Capítulo 6

Discusiones

Este capítulo discute las implicaciones de los resultados obtenidos a partir de los casos de estudio presentados. Se discutirán los hallazgos obtenidos en el contexto de la literatura existente, se analizará su relevancia para las operaciones industriales y se identificarán las áreas de mejora y oportunidades para futuras investigaciones.

6.1. Discusión de Resultados

Los resultados obtenidos en los casos de estudio reflejan una capacidad significativa de los modelos de deep learning no supervisado para reconstruir las señales de los sensores de los equipos. Esto permite identificar precisamente la presencia de anomalías cuando los datos reales difieren considerablemente de las predicciones generadas por los modelos. En particular, la aplicación de estos modelos en el segundo caso de estudio, utilizando datos reales, demostró la posibilidad de anticiparse a fallas con suficiente antelación para permitir una intervención proactiva del personal de mantenimiento. Esta anticipación es un claro indicativo de la viabilidad del mantenimiento predictivo apoyado por estas técnicas.

Las métricas de rendimiento obtenidas (5.7) subrayan esto y demuestran la gran oportunidad de mejora que existe en los procesos de mantenimiento preventivo actuales dentro de la industria y como se podría mejorar considerablemente la toma de decisiones y las acciones correctivas de maquinaria productiva.

La capacidad de anticiparse a fallas antes de que ocurran, y con un margen de tiempo suficiente para la intervención, puede revolucionar la manera en que se realizan las operaciones de mantenimiento, moviéndose hacia un enfoque más proactivo y menos reactivo. Este estudio aporta una base sólida para la futura implementación práctica de estas tecnologías, promoviendo una transición hacia sistemas de mantenimiento más inteligentes y automatizados, los cuales podrían resultar en un aumento significativo de la eficiencia operacional y en una reducción de costos de mantenimiento.

Además, la investigación presentada proporciona evidencia empírica de que el mantenimiento predictivo basado en IA no es meramente conceptual sino factible y efectivo. Con los modelos y enfoques desarrollados aquí, la industria tiene a su disposición herramientas potentes para mejorar el cuidado y la gestión de su maquinaria, anticipándose a las fallas

en lugar de simplemente reaccionar a ellas. Este enfoque no solo mejora la longevidad y el rendimiento de los equipos sino que también apoya una cultura de mantenimiento más sostenible y eficiente.

La implementación del umbral variable ha demostrado ser un componente crucial para la mejora de la eficiencia en la detección de anomalías en nuestros modelos de deep learning no supervisado. A diferencia de los umbrales estáticos, que no se adaptan a las fluctuaciones normales del funcionamiento del equipo, el umbral variable ha permitido una distinción más precisa entre los cambios operacionales normales y las verdaderas condiciones anómalas.

En la fase de análisis de los resultados obtenidos a través de la implementación del umbral variable, se observó que el número de falsas alertas experimentó una reducción marginal, pasando de nueve a siete. Este descenso, aunque no tan pronunciado como se podría esperar, es aún indicativo de una tendencia positiva hacia la reducción de las interrupciones operativas innecesarias. Resulta imperativo destacar que la adopción del umbral variable no comprometió la capacidad del sistema para enviar alertas verdaderas; todas las alertas legítimas se mantuvieron intactas. Sin embargo, el impacto más notable se manifestó en la duración de las alertas, especialmente aquellas generadas por cambios operativos, las cuales se vieron significativamente acortadas. Anteriormente, estas alertas podían persistir por días o incluso semanas, pero con la integración del umbral variable, la duración de estas no excedió las 24 horas. Este hallazgo subraya la relevancia del umbral variable como una herramienta dinámica que se adapta eficientemente a las fluctuaciones normales del proceso, permitiendo una distinción más precisa entre los cambios operativos normales y los eventos que efectivamente ameritan una alerta, lo cual es crucial para la optimización del mantenimiento preventivo y la confianza en los sistemas automatizados de monitoreo.

6.2. Limitaciones e implicaciones prácticas

Sin embargo, es fundamental abordar la incidencia de falsos positivos, especialmente en el entorno de datos reales, donde su número elevado puede comprometer la confiabilidad de los sistemas de mantenimiento predictivo. Las detenciones innecesarias, provocadas por alertas erróneas, implican considerables pérdidas en la producción y, por tanto, deben minimizarse y reducirse ojalá a cero.

La implementación de un umbral variable mostró ser una estrategia efectiva para reducir los tiempos de alertas generadas por cambios operacionales, contribuyendo así a la confiabilidad de los sistemas de alerta automática. Esta técnica demostró una notable adaptabilidad, ajustándose automáticamente a los cambios operativos en los datos sin comprometer la sensibilidad en la identificación de anomalías reales. Esta adaptabilidad es esencial para el funcionamiento autónomo de los sistemas de mantenimiento predictivo, permitiendo que los modelos permanezcan efectivos y relevantes ante la evolución de las condiciones operativas y el envejecimiento natural de las máquinas. A pesar de esta mejora, la posibilidad de que se generen alertas debido a variaciones operativas sigue presente; no obstante, la duración de estas alertas es breve, ya que el umbral se adapta y se ajusta al nivel de error proporcionado por el Autoencoder LSTM.

Otro aspecto relevante de estos modelos es su habilidad para distinguir entre diferentes

modos de operación y adaptar sus reconstrucciones acordemente. Esta capacidad depende críticamente de que dichos modos sean introducidos durante la fase de entrenamiento, lo que conlleva limitaciones en términos de la versatilidad operativa de la maquinaria. Se recomienda, por lo tanto, que la data de entrenamiento incluya los modos operativos más predominantes o comunes según el historial de funcionamiento del equipo. Complementariamente, el uso del umbral variable puede mitigar la generación de falsas alertas en situaciones que requieran nuevos modos operativos por necesidades operacionales.

Es crucial mencionar que para mantener la eficacia de los modelos de detección de anomalías, se requieren reentrenamientos periódicos, especialmente después de cambios significativos en la maquinaria, como la sustitución de piezas claves o motores. Ya que, este tipo de cambios significan en la mayoría de los casos nuevas formas de vibración o disminución en temperaturas de descansos y rodamientos de los motores. La frecuencia y el volumen de datos necesarios para el reentrenamiento deben equilibrarse para asegurar que los modelos sean suficientemente robustos. Se aconseja usar datos de entre uno y tres meses para abarcar una gama amplia de condiciones operativas y asegurar la robustez del modelo. Entrenamientos basados en periodos más cortos pueden resultar en modelos que reaccionan excesivamente a fluctuaciones menores en los datos de los sensores, generando una cantidad inaceptable de falsas alertas.

Además de las estrategias técnicas implementadas, es esencial destacar la importancia de la validación de las alertas generadas por los modelos a través de observaciones en terreno. La interacción entre los sistemas de detección automatizada y el juicio experto de los operarios es crucial para la confirmación de anomalías reales. Esto subraya que, aunque los modelos de deep learning no supervisado son herramientas poderosas y se deben tener en consideración, no se debe confiar en ellos ciegamente sin una previa verificación.

En el contexto de una implementación industrial real, la participación activa del personal en terreno para validar las alertas se vuelve indispensable. Esta colaboración humana no solo sirve para confirmar la presencia de condiciones anómalas, sino también para enriquecer el entendimiento contextual de las alertas que podrían ser erróneamente interpretadas como anomalías cuando en realidad corresponden a cambios operacionales legítimos.

Al incorporar el conocimiento y la experiencia del personal técnico en la validación de las alertas, es posible construir un registro histórico robusto y confiable. Este registro puede ser utilizado para realizar análisis retrospectivos de funcionamiento histórico y para el entrenamiento continuo de los modelos, mejorando así su precisión y fiabilidad. La creación de este tipo de datasets enriquecidos y validados por humanos es un paso esencial hacia la maduración de los sistemas de mantenimiento predictivo basados en IA, facilitando la transición hacia una mayor autonomía de diagnóstico y una confianza reforzada en las decisiones automatizadas.

6.3. Recomendaciones y Trabajo Futuro

Mirando hacia el futuro, se recomienda explorar soluciones complementarias al umbral variable para mejorar la diferenciación entre los diferentes modos de operación. Una posibilidad

prometedora es la aplicación de técnicas de clustering para identificar y clasificar los modos de operación de manera automática. Gracias al EDA, ya fue posible detectar dos modos de operación predominantes por lo que es interesante seguir explorando este tipo de técnicas. Esto podría facilitar la creación de modelos específicos entrenados para cada modo de operación identificado, permitiendo una detección de anomalías más afinada y adaptada a las particularidades de cada estado operativo. Este enfoque podría no solo mejorar la precisión de las alertas, sino también proporcionar una mayor comprensión de las dinámicas operacionales de la maquinaria. Este tipo de investigación podría abrir nuevas vías para el mantenimiento predictivo, llevando a una implementación más efectiva y eficiente en entornos industriales complejos.

La presente discusión enfatiza que, aunque los modelos de deep learning no supervisado ofrecen un enfoque prometedor para el mantenimiento predictivo, su integración en entornos industriales exige una consideración cuidadosa de las dinámicas operativas y los cambios en los equipos. Estos factores, junto a la participación activa de personal en terreno deben ser contemplados para lograr sistemas de detección de anomalías fiables y efectivos.

Capítulo 7

Conclusiones

Esta investigación ha realizado un aporte significativo en el campo del mantenimiento predictivo a través de la implementación y evaluación de modelos de deep learning no supervisados. A lo largo de este estudio, se ha validado la aplicabilidad de dichos modelos para la detección temprana de anomalías y la predicción de fallas en equipos industriales. Los resultados obtenidos sugieren que su uso puede conducir a ahorros sustanciales tanto en tiempo como en recursos, gracias a la anticipación estratégica que ofrecen.

Inicialmente, con la base de datos SKAB —un entorno controlado con anomalías inducidas artificialmente—, se realizó un análisis comparativo entre algoritmos clásicos de machine learning y algoritmos de deep learning basados en autoencoders. Los resultados mostraron que los autoencoders tienen una capacidad superior y notable para la detección de anomalías, evidenciando su habilidad para generar predicciones precisas a través de la reconstrucción de señales. Al extender estos modelos a un conjunto de datos industriales real, se mantuvo un rendimiento satisfactorio, con métricas robustas en la detección de anomalías y la predicción de fallas. Se lograron casos de éxito donde se anticiparon fallas críticas con días de margen, lo que confirma el valor práctico de estas soluciones en el ámbito del mantenimiento predictivo.

Por otro lado, la adopción del umbral variable ha surgido como un método prometedor, mejorando la detección de anomalías al acortar la duración de las alertas falsas y conservar la precisión en las alertas legítimas. Este avance representa un progreso considerable en la minimización de interrupciones operativas y en el fortalecimiento de la confiabilidad de los modelos.

No obstante, se debe hacer mención de las limitaciones inherentes a la metodología actual, como la necesidad de reentrenamiento constante y la dependencia de la representatividad de modos operativos en los datos utilizados para el entrenamiento. Estos desafíos deben ser cuidadosamente abordados para potenciar aún más la eficiencia de los sistemas predictivos.

El impacto de este trabajo en el mantenimiento predictivo industrial es considerable, proponiendo un cambio hacia sistemas de detección de anomalías más inteligentes y automatizados, que fomentan una estrategia de mantenimiento proactiva y preventiva.

Para futuros trabajos, se sugiere la exploración de modelos adaptativos y dinámicos que puedan incorporar nuevos modos de operación sin la necesidad de reentrenamiento constante.

Además, el uso de técnicas complementarias como el clustering para la identificación automática de modos operativos podría ofrecer una diferenciación aún más precisa y reducir la carga operativa en el mantenimiento predictivo.

En conclusión, la investigación ha proporcionado insights valiosos y ha desarrollado metodologías que pueden aplicarse efectivamente en entornos industriales para mejorar la precisión y la confiabilidad del mantenimiento predictivo. Se espera que los hallazgos de este estudio inspiren y guíen futuras investigaciones, contribuyendo al avance y la adopción de soluciones inteligentes en la industria.

Bibliografía

- [1] McKinsey and Company, “Industry 4.0: How to navigate digitization of the manufacturing sector”. <https://www.mckinsey.com/business-functions/operations/our-insights/industry-40-how-to-navigate-digitization-of-the-manufacturing-sector>, 2015.
- [2] Lei, Y., Yang, B., y Jiang, X., “Applications of machine learning to machine fault diagnosis: A review and roadmap”, *Mechanical Systems and Signal Processing*, 2019, [doi:10.1016/j.ymssp.2019.106587](https://doi.org/10.1016/j.ymssp.2019.106587).
- [3] Ture, B. A., Akbulut, A., y Zaim, A. H., “Techniques for apply predictive maintenance and remaining useful life: A systematic mapping study”, *Journal Name*, 2020, [doi:10.35193/bseufbd.900214](https://doi.org/10.35193/bseufbd.900214).
- [4] Carvalho, T., Soares, F., Vita, R., Francisco, R., Basto, J., y Alcalá, S., “A systematic literature review of machine learning methods applied to predictive maintenance”, *Computers & Industrial Engineering*, vol. 137, p. 106024, 2019, [doi:10.1016/j.cie.2019.106024](https://doi.org/10.1016/j.cie.2019.106024).
- [5] Amruthnath, N. y Gupta, T., “A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance”, en *2018 5th International Conference on Industrial Engineering and Applications (ICIEA)*, IEEE, 2018, [doi:10.1109/iea.2018.8387124](https://doi.org/10.1109/iea.2018.8387124).
- [6] Balakrishnan, G., Zhao, A., Sabuncu, M. R., Gutttag, J., y Dalca, A. V., “Unsupervised deep learning in medical image analysis”, *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 7, pp. 1559–1572, 2018.
- [7] Wu, J., Zhang, C., Xue, T., Freeman, B., y Tenenbaum, J., “Variational autoencoder for deep learning of images, labels and captions”, en *Advances in neural information processing systems*, pp. 2352–2360, 2016.
- [8] Dalca, A. V., Gutttag, J., y Sabuncu, M. R., “Unsupervised deep learning for bayesian brain mri segmentation”, en *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 649–657, Springer, Cham, 2018.
- [9] Bampoula, X., Siaterlis, G., Nikolakis, N., y Alexopoulos, K., “A deep learning model for predictive maintenance in cyber-physical production systems using lstm autoencoders”, *Sensors*, vol. 21, no. 3, p. 972, 2021, [doi:10.3390/s21030972](https://doi.org/10.3390/s21030972).
- [10] Di Mauro, M. P., Muscato, G., Pilato, G., y Sottile, G., “Unsupervised, deep learning-based detection of failures in industrial equipments: The future of predictive maintenance”, *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2233–2241, 2020, [doi:10.1109/TII.2019.2952664](https://doi.org/10.1109/TII.2019.2952664).
- [11] Jezzini, A., Ayache, M., Elkhansa, L., Makki, B., y Zein, M., “Effects of predictive maintenance(pdm), proactive maintenace(pom) & preventive maintenance(pm) on minimizing

- the faults in medical instruments”, en 2013 2nd International Conference on Advances in Biomedical Engineering, pp. 53–56, IEEE, 2013, [doi:10.1109/ICABME.2013.6648845](https://doi.org/10.1109/ICABME.2013.6648845).
- [12] Baptista, M., Sankararaman, S., de Medeiros, I. P., Nascimento, C., Prendinger, H., y Henriques, E. M., “Forecasting fault events for predictive maintenance using data-driven techniques and arma modeling”, *Computers and Industrial Engineering*, 2018, [doi:10.1016/j.cie.2017.10.033](https://doi.org/10.1016/j.cie.2017.10.033).
- [13] Goodfellow, I., Bengio, Y., y Courville, A., *Deep learning*, vol. 1. MIT press Cambridge, 2016.
- [14] Zhao, Q., Sun, J., Yu, C., Cui, C., Lv, L., y Xiao, J., “A paralleled large-scale astronomical cross-matching function”, en *International Conference on Algorithms and Architectures for Parallel Processing*, pp. 604–614, Springer, 2009.
- [15] Ioffe, S. y Szegedy, C., “Batch normalization: Accelerating deep network training by reducing internal covariate shift”, en *International conference on machine learning*, pp. 448–456, PMLR, 2015.
- [16] LeCun, Y., Bottou, L., Bengio, Y., y Haffner, P., “Gradient-based learning applied to document recognition”, *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [17] Krizhevsky, A., Sutskever, I., y Hinton, G. E., “Imagenet classification with deep convolutional neural networks”, en *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- [18] Chollet, F., “Xception: Deep learning with depthwise separable convolutions”, en *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1251–1258, 2017.
- [19] He, K., Zhang, X., Ren, S., y Sun, J., “Spatial pyramid pooling in deep convolutional networks for visual recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.
- [20] He, K., Zhang, X., Ren, S., y Sun, J., “Deep residual learning for image recognition”, en *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
- [21] Sarker, I. H., “Deep cybersecurity: A comprehensive overview from neural network and deep learning perspective”, *SN Computer Science*, vol. 2, no. 3, pp. 1–16, 2021.
- [22] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., y Rabinovich, A., “Going deeper with convolutions”, en *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.
- [23] Dupond, S., “A thorough review on the current advance of neural network structures”, *Annual Review of Control*, vol. 14, pp. 200–230, 2019.
- [24] Hochreiter, S. y Schmidhuber, J., “Long short-term memory”, *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [25] Siami-Namini, S., Tavakoli, N., y Namin, A. S., “The performance of lstm and bilstm in forecasting time series”, en *2019 IEEE International Conference on Big Data (Big Data)*, pp. 3285–3292, IEEE, 2019.
- [26] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., y Bengio, Y., “Learning phrase representations using rnn encoder-decoder for statistical

- machine translation”, en arXiv preprint arXiv:1406.1078, 2014.
- [27] Chung, J., Gulcehre, C., Cho, K. H., y Bengio, Y., “Empirical evaluation of gated recurrent neural networks on sequence modeling”, arXiv preprint arXiv:1412.3555, 2014.
- [28] Kingma, D. y Ba, J., “Adam: A method for stochastic optimization”, arXiv preprint arXiv:1412.6980, 2014.
- [29] Chandola, V., Banerjee, A., y Kumar, V., “Anomaly detection: A survey”, ACM computing surveys (CSUR), vol. 41, no. 3, pp. 1–58, 2009.
- [30] He, H. y Garcia, E. A., “Learning from imbalanced data”, IEEE Transactions on knowledge and data engineering, vol. 21, no. 9, pp. 1263–1284, 2009.
- [31] He, H. y Ma, Y., eds., Imbalanced learning: foundations, algorithms, and applications. John Wiley & Sons, 2013.
- [32] Schlüter, H. M., Tan, J., Hou, B., y Kainz, B., “Natural synthetic anomalies for self-supervised anomaly detection and localization”, en European Conference on Computer Vision, pp. 474–489, Springer Nature Switzerland, 2022.
- [33] Shen, L., Li, Z., y Kwok, J., “Timeseries anomaly detection using temporal hierarchical one-class network”, Advances in Neural Information Processing Systems, vol. 33, pp. 13016–13026, 2020.
- [34] Chalapathy, R., Menon, A. K., y Chawla, S., “Anomaly detection using one-class neural networks”, arXiv preprint arXiv:1802.06360, 2018.
- [35] Wang, H., Bah, M., y Hammad, M., “Progress in outlier detection techniques: A survey”, IEEE Access, vol. 7, pp. 107964–108000, 2019.
- [36] Cui, Y., Liu, Z., y Lian, S., “A survey on unsupervised industrial anomaly detection algorithms”, arXiv preprint arXiv:2204.11161, 2022.
- [37] Aggarwal, C., Outlier analysis. Springer, 2015.
- [38] 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, pp. 825–834, 2018, doi:10.1016/j.renene.2018.05.024.
- [39] Clark, J., Liu, Z., y Japkowicz, N., “Adaptive threshold for outlier detection on data streams”, en 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA), (Turin, Italy), pp. 41–49, IEEE, 2018, doi:10.1109/DSAA.2018.00014.
- [40] Aminikhanghahi, S. y Cook, D., “A survey of methods for time series change point detection”, Knowledge and Information Systems, vol. 51, pp. 339–367, 2017, doi:10.1007/s10115-016-0987-z.
- [41] Lavin, A. y Ahmad, S., “Evaluating real-time anomaly detection algorithms – the numenta anomaly benchmark”, en 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pp. 38–44, IEEE, 2015, doi:10.1109/ICMLA.2015.141.
- [42] Truong, C., Oudre, L., y Vayatis, N., “Selective review of offline change point detection methods”, Signal Processing, vol. 167, p. 107299, 2020, doi:10.1016/j.sigpro.2019.107299.
- [43] Katser, I. D. y Kozitsin, V. O., “Skoltech anomaly benchmark (skab)”. <https://www.kaggle.com/dsv/1693952>, 2020, doi:10.34740/KAGGLE/DSV/1693952.

- [44] Kingma, D. P. y Ba, J., “Adam: A method for stochastic optimization”, arXiv preprint arXiv:1412.6980, 2014, [doi:10.48550/arXiv.1412.6980](https://doi.org/10.48550/arXiv.1412.6980).

Anexos

Anexo A. Resultados Base de Datos SKAB

A.1. EDA

Tabla A.1: Resumen estadístico de los experimentos

Metric	anomaly free	valve1[1.csv]	valve2[1.csv]	other[1.csv]
Accelerometer1RMS mean	0.212604	0.026868	0.027663	0.080549
Accelerometer2RMS mean	0.268407	0.040045	0.040037	0.131089
Current mean	2.403010	0.979026	0.976845	1.569378
Pressure mean	0.111266	0.074473	0.048541	0.054711
Temperature mean	89.472308	73.713523	67.895174	92.567708
Thermocouple mean	28.474310	25.740027	24.265096	22.237863
Voltage mean	228.647173	231.413459	230.143218	230.246807
Volume Flow RateRMS mean	125.237471	31.566819	31.472275	76.157459
Accelerometer1RMS std	0.004740	0.000313	0.000333	0.001167
Accelerometer2RMS std	0.003989	0.000988	0.001086	0.003030
Current std	0.486469	0.277454	0.269748	0.386894
Pressure std	0.253332	0.249990	0.270689	0.270437
Temperature std	0.667109	1.333144	0.695198	1.933215
Thermocouple std	0.730734	0.063399	0.021573	0.018487
Voltage std	10.980578	10.764134	11.235994	11.037523
Volume Flow RateRMS std	1.605327	0.760021	1.314535	1.155151

Tabla A.3: Tabla resumen para los 34 experimentos

	Tipo de Anomalia	Experimento	Descripción	Duración	Filas	% Anomalia	Puntos de Cambio	NaNs
0	anomaly_free	0	Modo normal	02:46:00	9405	0	0	0
1	valve1	0.csv	Cierre de válvula aguas abajo	00:19:59	1147	34.96	4.0	0
2	valve1	1.csv	Cierre de válvula aguas abajo	00:20:00	1145	35.11	4.0	0
3	valve1	10.csv	Cierre de válvula aguas abajo	00:19:59	1146	34.99	4.0	0
4	valve1	11.csv	Cierre de válvula aguas abajo	00:20:00	1141	34.97	4.0	0
5	valve1	12.csv	Cierre de válvula aguas abajo	00:20:00	1140	35.0	4.0	0
6	valve1	13.csv	Cierre de válvula aguas abajo	00:19:59	1140	35.0	4.0	0
7	valve1	14.csv	Cierre de válvula aguas abajo	00:19:59	1139	35.03	4.0	0
8	valve1	15.csv	Cierre de válvula aguas abajo	00:20:00	1150	35.13	4.0	0
9	valve1	2.csv	Cierre de válvula aguas abajo	00:19:59	1075	31.35	3.0	0
10	valve1	3.csv	Cierre de válvula aguas abajo	00:20:01	1148	35.19	4.0	0
11	valve1	4.csv	Cierre de válvula aguas abajo	00:19:59	1095	31.87	4.0	0
12	valve1	5.csv	Cierre de válvula aguas abajo	00:19:59	1154	34.92	4.0	0
13	valve1	6.csv	Cierre de válvula aguas abajo	00:19:59	1154	35.1	4.0	0
14	valve1	7.csv	Cierre de válvula aguas abajo	00:20:00	1094	37.02	4.0	0
15	valve1	8.csv	Cierre de válvula aguas abajo	00:19:58	1144	34.97	4.0	0
16	valve1	9.csv	Cierre de válvula aguas abajo	00:20:00	1148	35.02	4.0	0
17	valve2	0.csv	Cierre de válvula en la entrada	00:19:59	1125	35.02	4.0	0
18	valve2	1.csv	Cierre de válvula en la entrada	00:20:00	1063	31.33	4.0	0
19	valve2	2.csv	Cierre de válvula en la entrada	00:19:59	1129	34.99	4.0	0
20	valve2	3.csv	Cierre de válvula en la entrada	00:17:38	995	39.7	4.0	0
21	other	1.csv	Simulación de fugas y adiciones de fluido	00:13:00	745	25.23	2.0	0
22	other	10.csv	Aumento lento del agua en el circuito	00:23:18	1327	44.16	4.0	0
23	other	11.csv	Aumento repentino del agua en el circuito	00:20:54	1190	37.9	4.0	0
24	other	12.csv	Drenaje de agua del tanque hasta cavitación	00:20:03	1048	29.48	4.0	0
25	other	13.csv	Suministro bifásico a la entrada de la bomba	00:18:54	923	28.71	4.0	0
26	other	14.csv	Suministro de agua a temperatura elevada	00:15:51	905	33.37	3.0	0
27	other	2.csv	Simulación de fugas y adiciones de fluido	00:17:43	780	49.23	4.0	0
28	other	3.csv	Simulación de fugas y adiciones de fluido	00:19:58	1137	35.0	4.0	0
29	other	4.csv	Simulación de fugas y adiciones de fluido	00:20:56	1191	33.17	3.0	0
30	other	5.csv	Comportamiento abrupto del rotor	00:20:08	1155	35.5	2.0	0
31	other	6.csv	Comportamiento lineal del rotor	00:20:00	1147	35.05	4.0	0
32	other	7.csv	Comportamiento escalonado del rotor	00:19:01	1090	31.83	4.0	0
33	other	8.csv	Función delta de Dirac en rotor	00:20:02	1147	35.14	4.0	0
34	other	9.csv	Comportamiento exponencial del rotor	00:19:59	1144	35.05	4.0	0

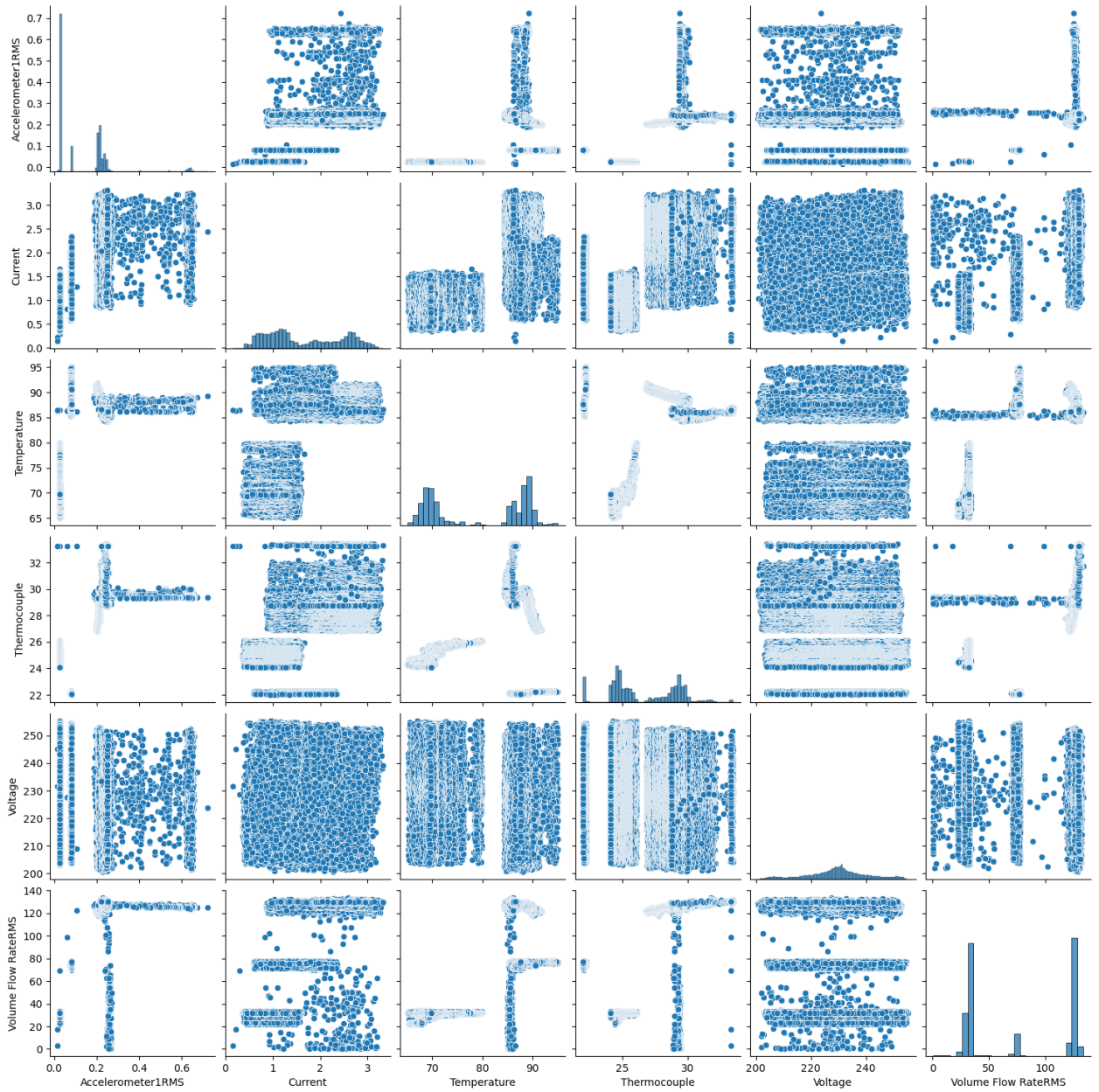


Figura A.1: Pairplot BBDD SKAB

A.2. Ejemplo SKAB

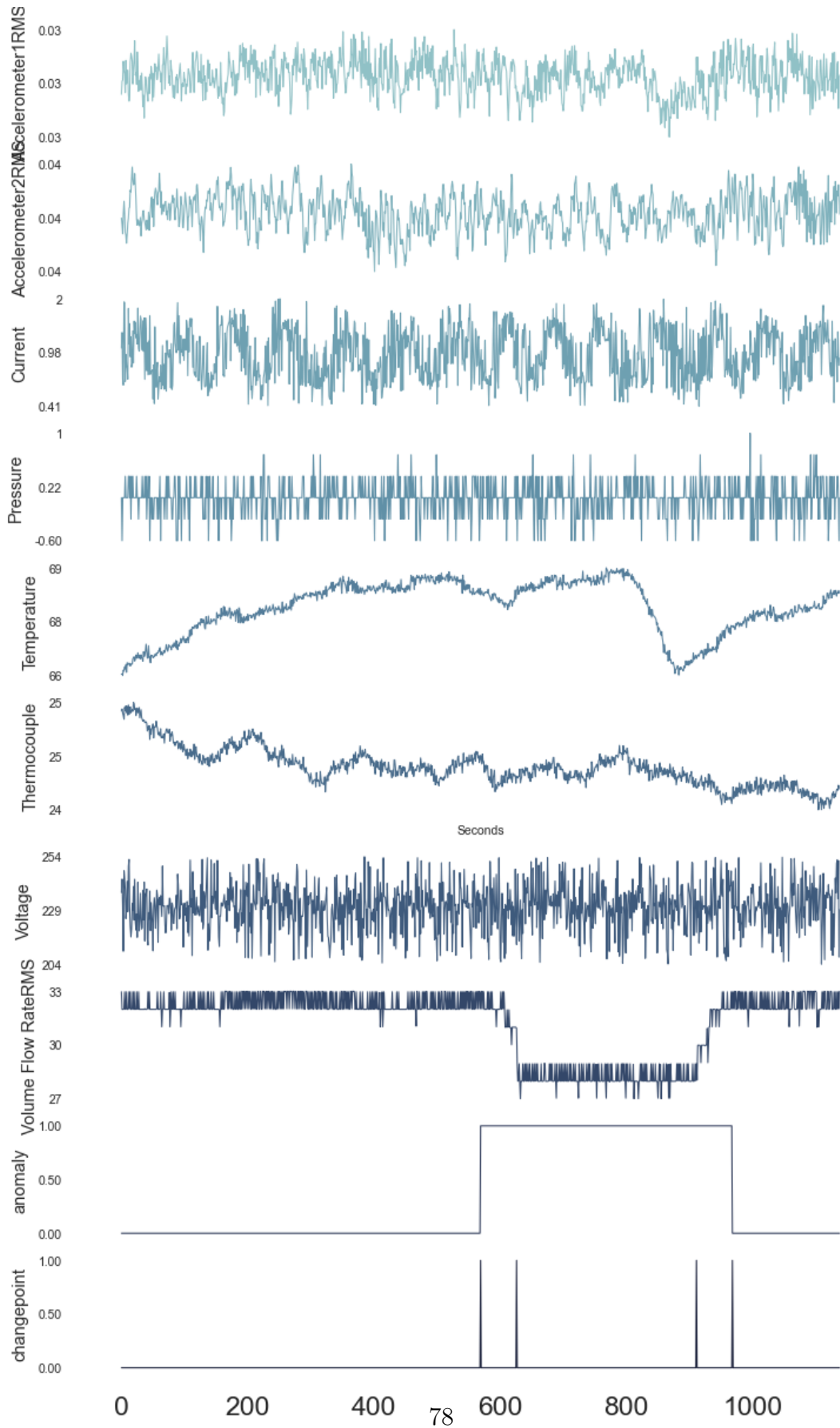
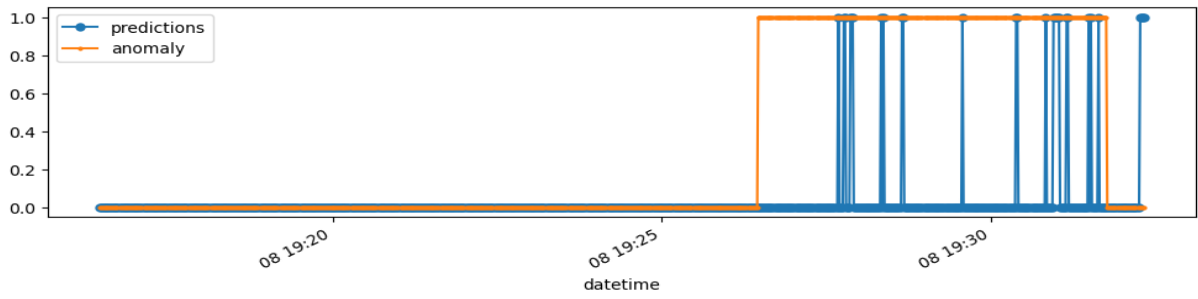
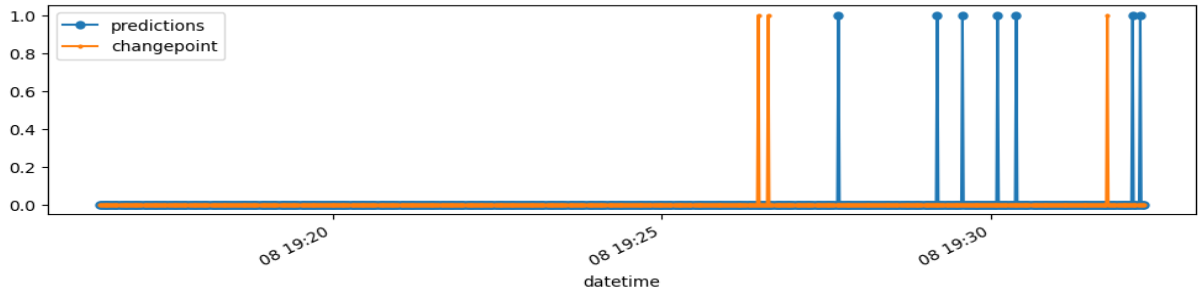


Figura A.2: Señales para experimento 20, BBDD SKAB

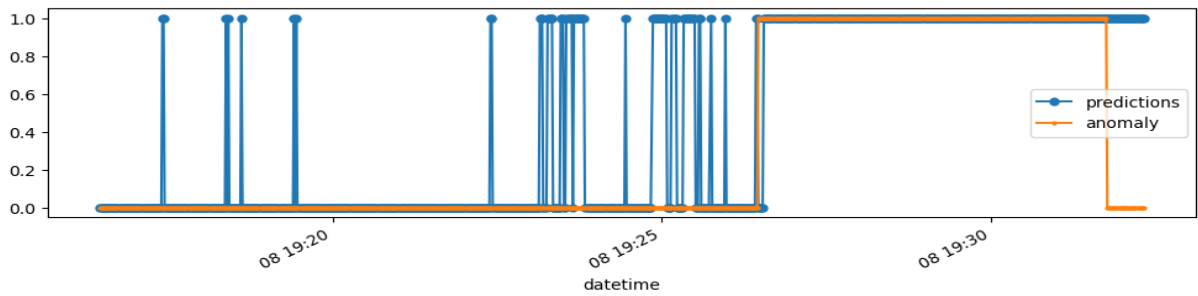


(a) Detección de Anomalías

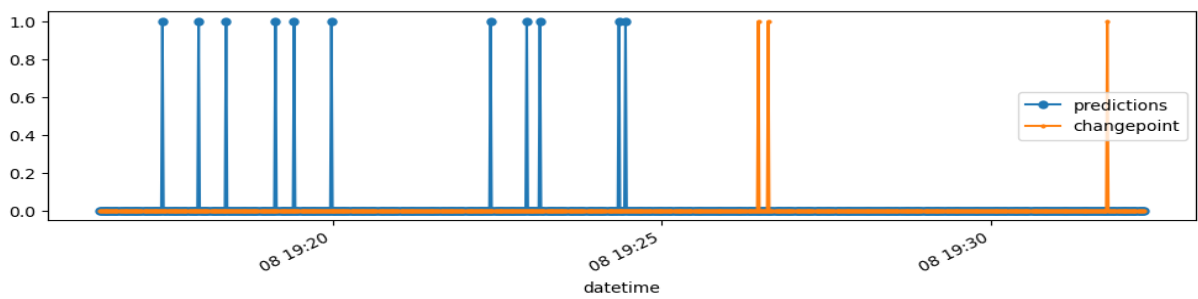


(b) Detección Puntos de Cambios

Figura A.3: Ejemplo Predicciones Isolation Forest



(a) Detección de Anomalías



(b) Detección Puntos de Cambios

Figura A.4: Ejemplo Predicciones Elliptic Envelope

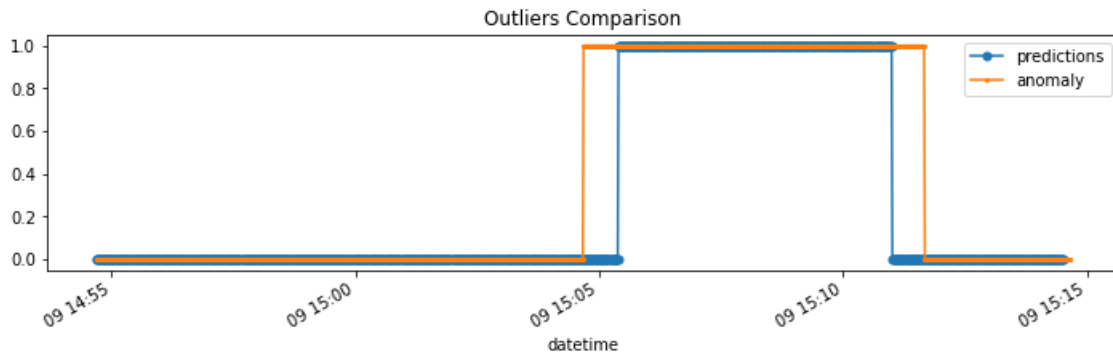


Figura A.5: Ejemplo detección de anomalías CNN Autoencoder. Se realiza comparación entre las etiquetas y las predicciones del modelo.

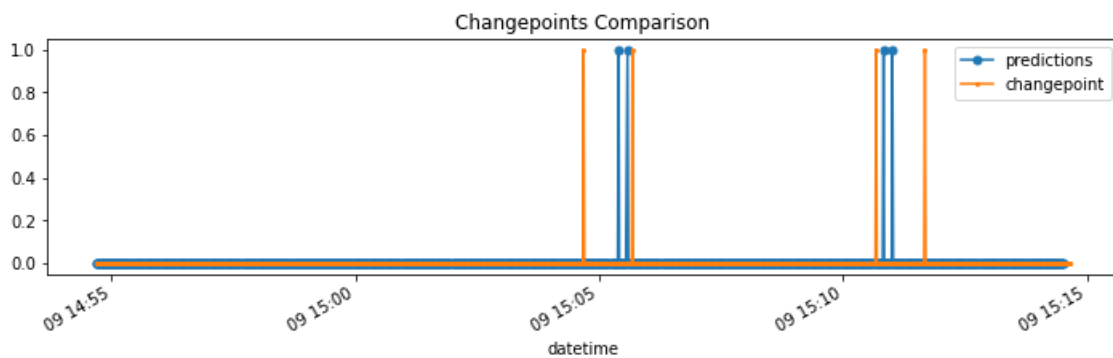


Figura A.6: Ejemplo detección de puntos de cambio CNN Autoencoder

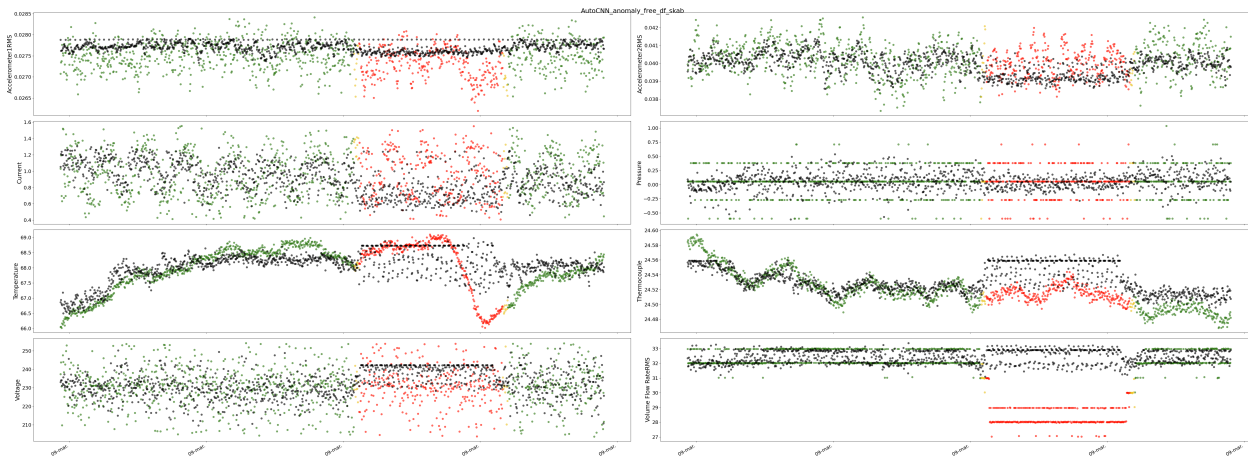


Figura A.7: Reconstrucciones CNN Autoencoder. Los puntos negros corresponden a las reconstrucciones que realiza el modelo CNN AE en base a la señal original. Los colores verde indican que es un estado sin anomalías, mientras que amarillo y rojo corresponden a anomalías leves y graves respectivamente.

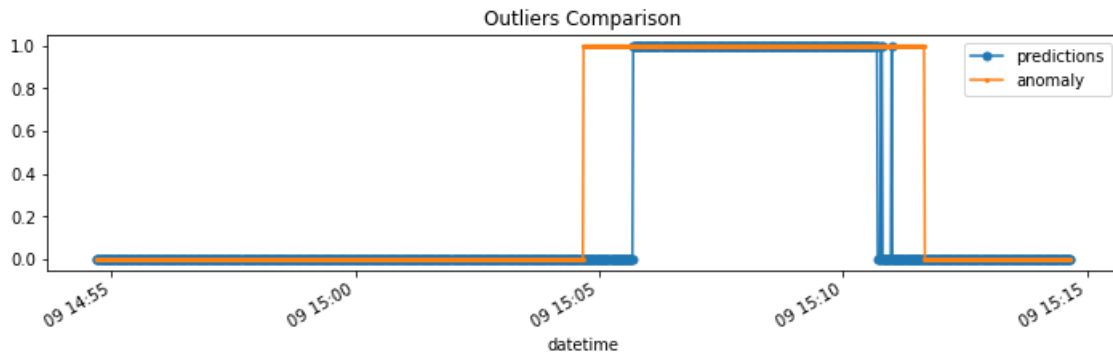


Figura A.8: Ejemplo detección de anomalías MLP Autoencoder. Se realiza comparación entre las etiquetas y las predicciones del modelo.

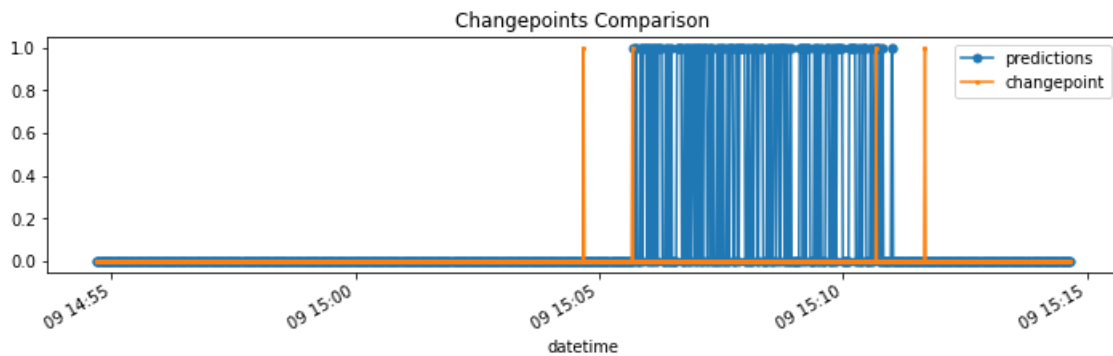


Figura A.9: Ejemplo detección de puntos de cambio MLP Autoencoder



Figura A.10: Reconstrucciones MLP Autoencoder. Los puntos negros corresponden a las reconstrucciones que realiza el modelo MLP AE en base a la señal original. Los colores verde indican que es un estado sin anomalías, mientras que amarillo y rojo corresponden a anomalías leves y graves respectivamente.

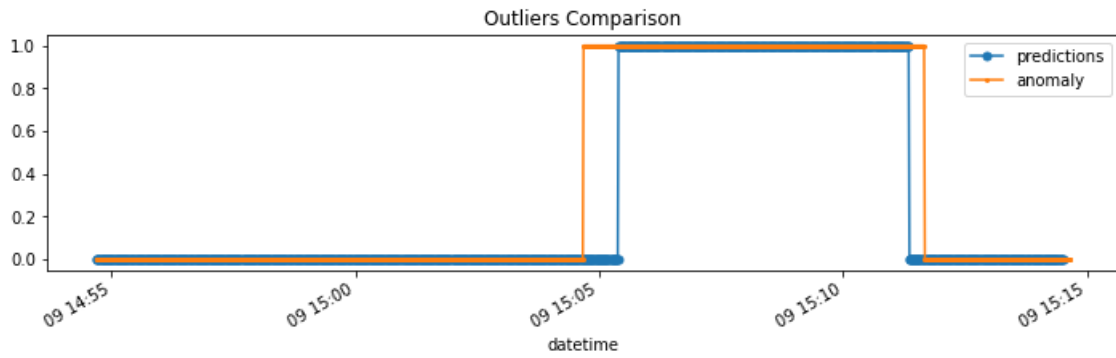


Figura A.11: Ejemplo detección de anomalías GRU Autoencoder. Se realiza comparación entre las etiquetas y las predicciones del modelo.

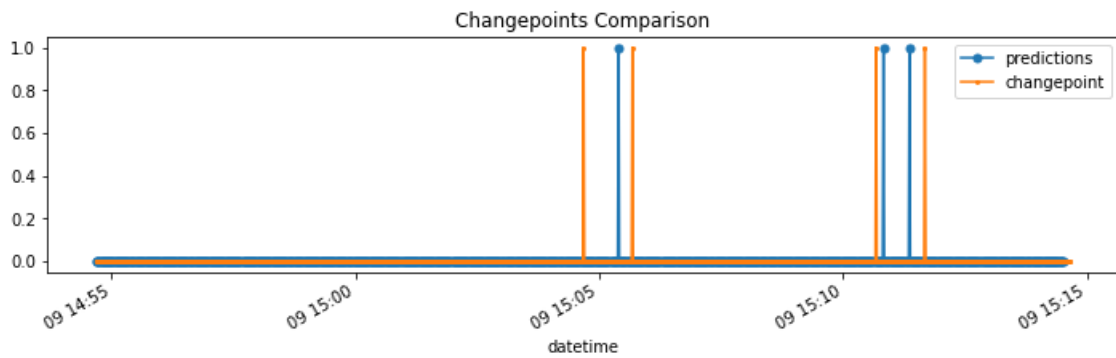


Figura A.12: Ejemplo detección de puntos de cambio GRU Autoencoder

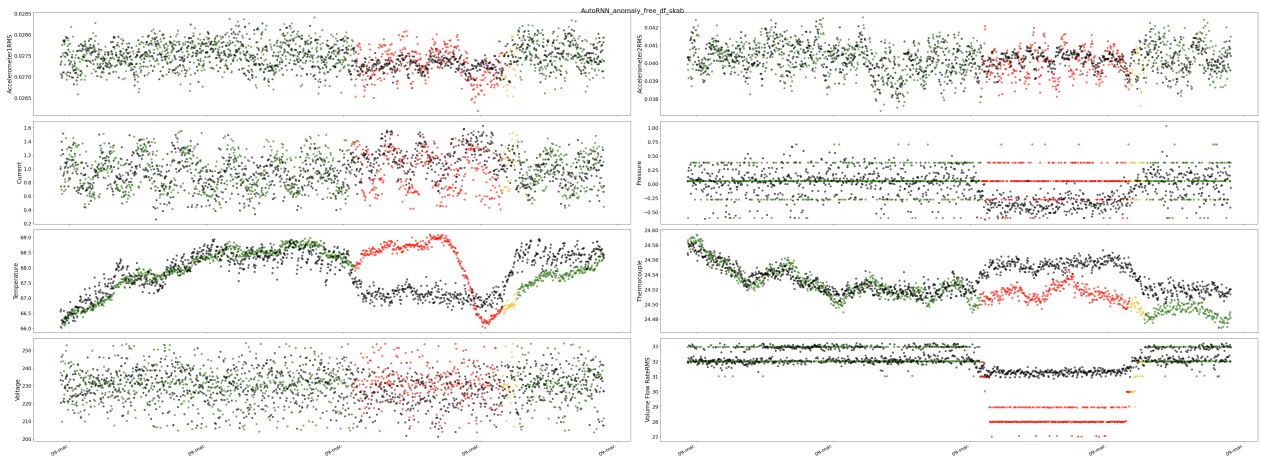


Figura A.13: Reconstrucciones GRU Autoencoder. Los puntos negros corresponden a las reconstrucciones que realiza el modelo GRU AE en base a la señal original. Los colores verde indican que es un estado sin anomalías, mientras que amarillo y rojo corresponden a anomalías leves y graves respectivamente.

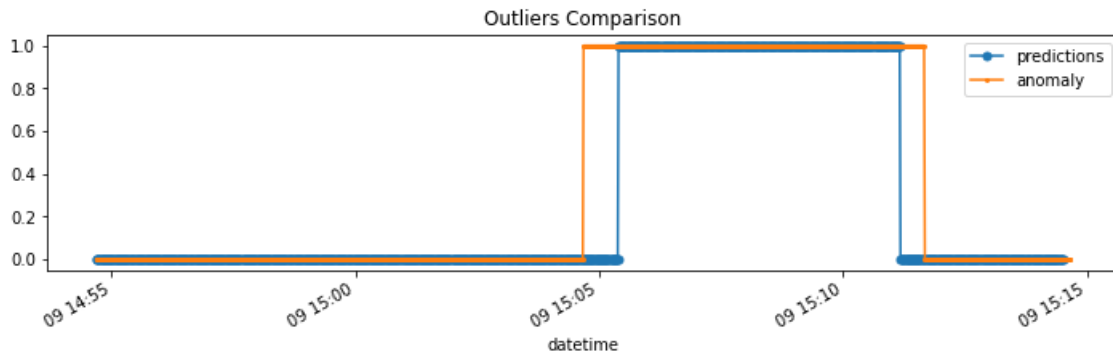


Figura A.14: Ejemplo detección de anomalías LSTM Autoencoder. Se realiza comparación entre las etiquetas y las predicciones del modelo.

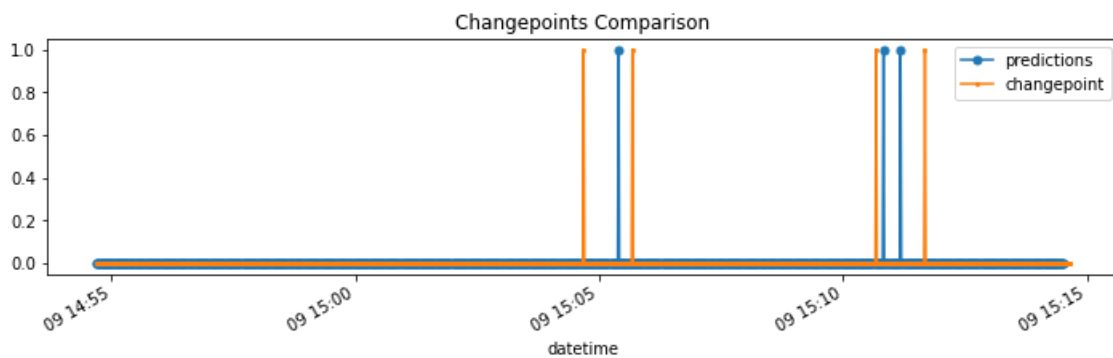


Figura A.15: Ejemplo detección de puntos de cambio LSTM Autoencoder

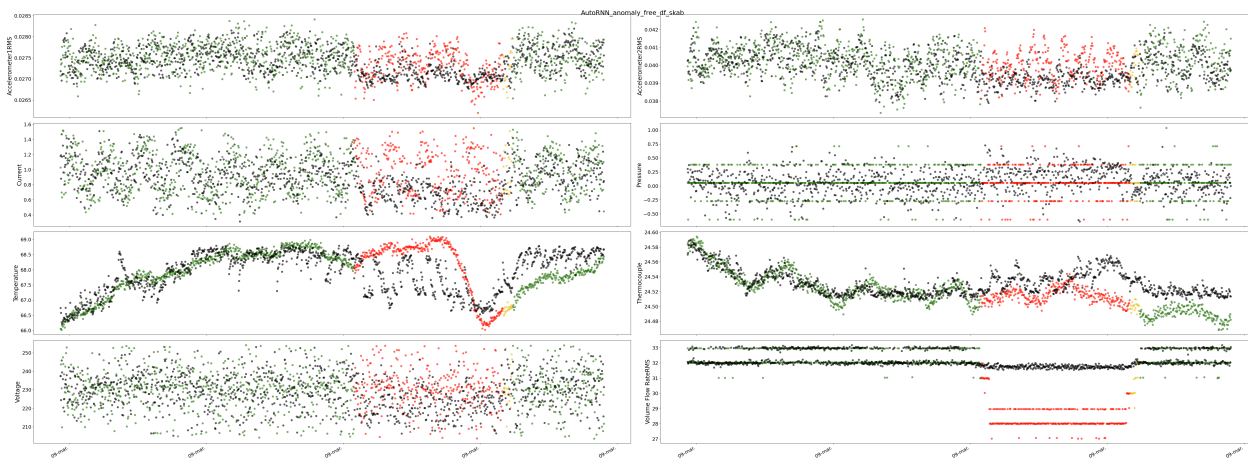


Figura A.16: Reconstrucciones LSTM Autoencoder. Los puntos negros corresponden a las reconstrucciones que realiza el modelo LSTM AE en base a la señal original. Los colores verde indican que es un estado sin anomalías, mientras que amarillo y rojo corresponden a anomalías leves y graves respectivamente.

Anexo B. Resultados Base de Datos Bombas de Alta Presión y Booster

B.1. EDA

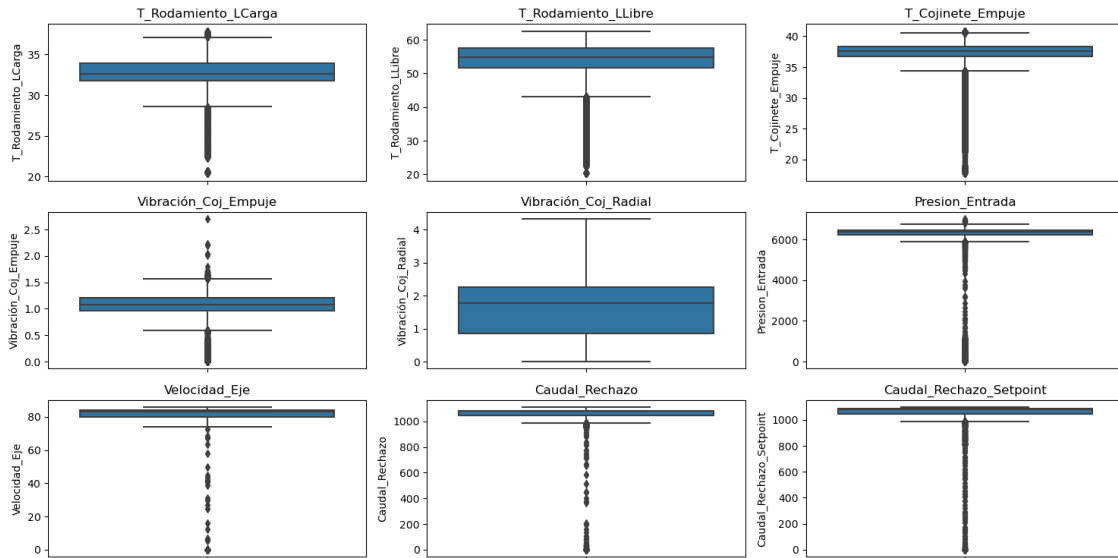


Figura B.1: Gráficos Boxplot Bomba Booster R6

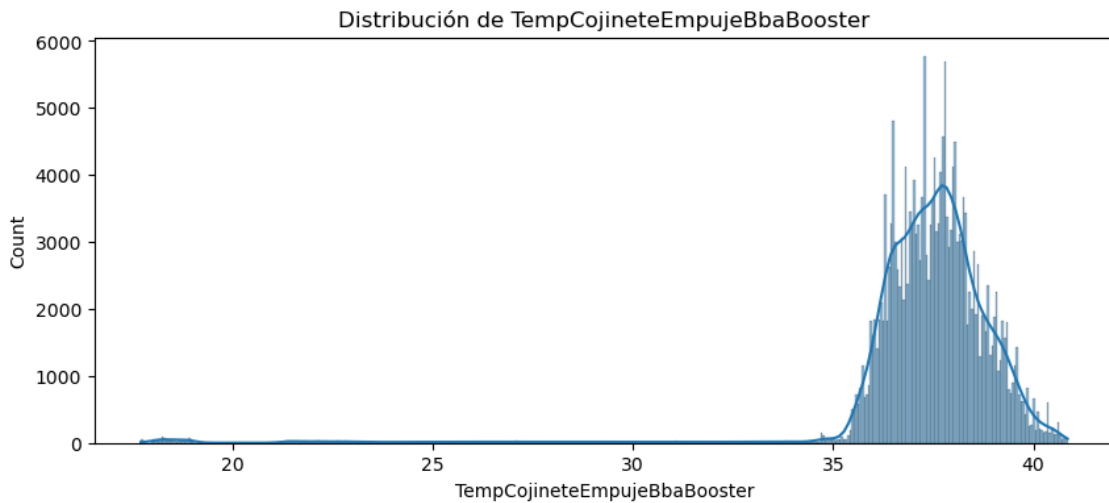


Figura B.2: Histograma Temperatura Cojinete Empuje Bomba Booster R6

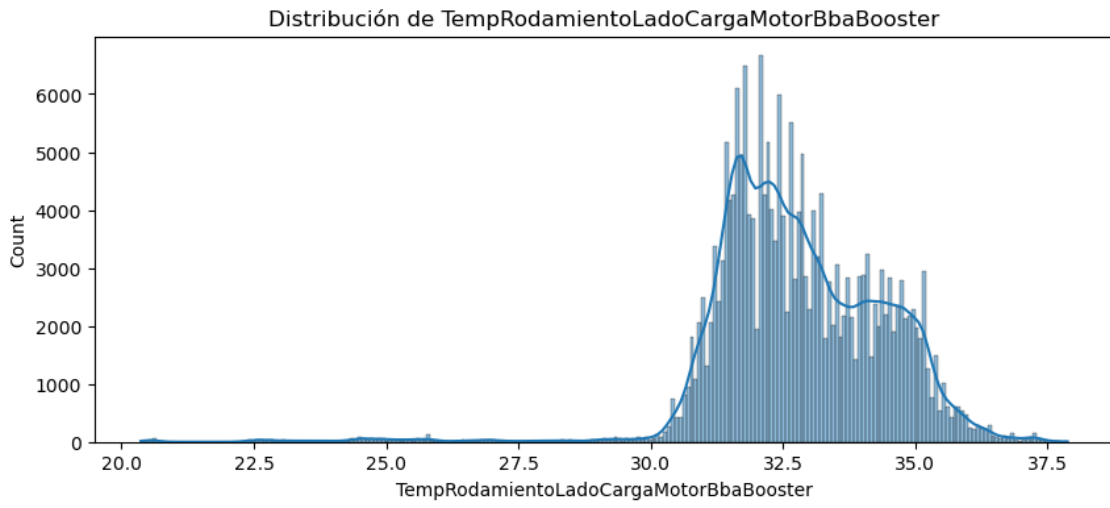


Figura B.3: Histograma Temperatura Rodamiento Lado Carga Motor Bomba Booster R6

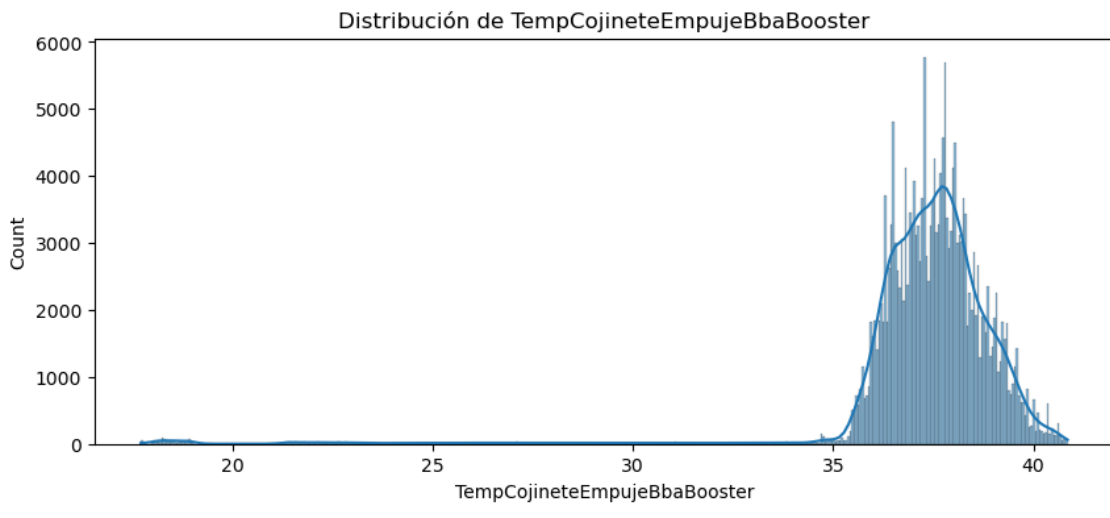


Figura B.4: Histograma Temperatura Rodamiento Lado Libre Motor Bomba Booster R6

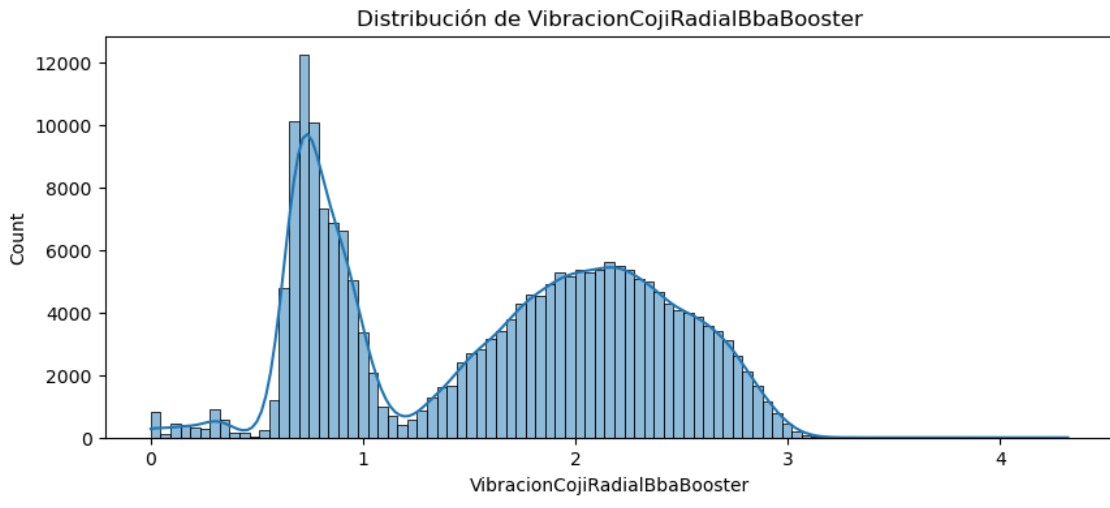


Figura B.5: Vibración Cojinete Radial Bomba Booster R6

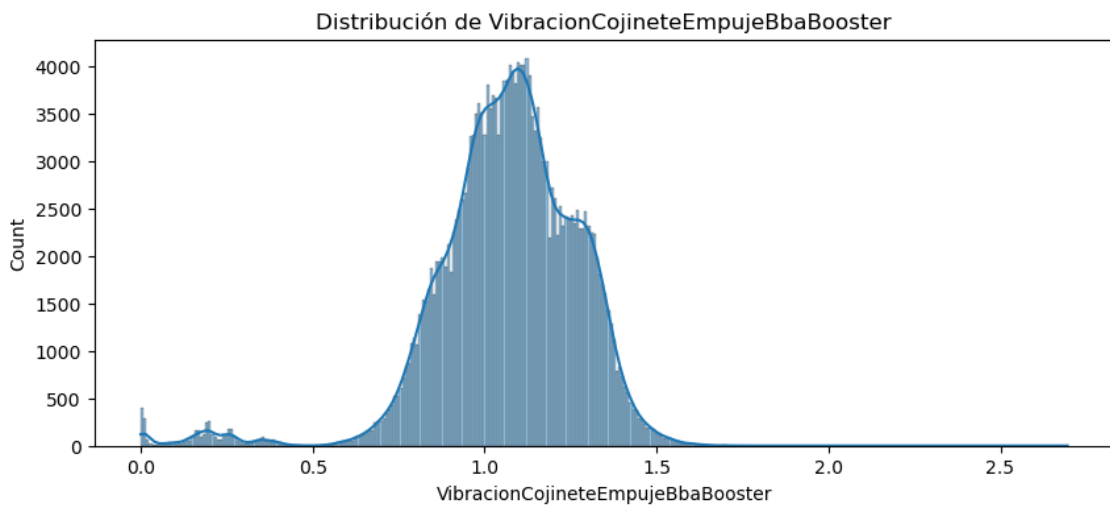


Figura B.6: Histograma Vibración Cojinete Empuje Bomba Booster R6

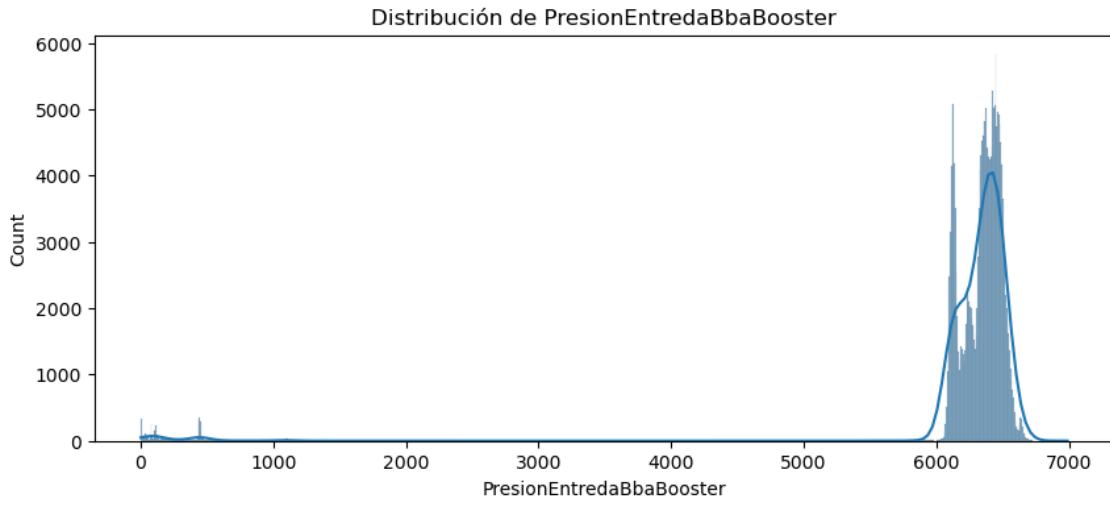


Figura B.7: Histograma Presión Entrada Bomba Booster R6

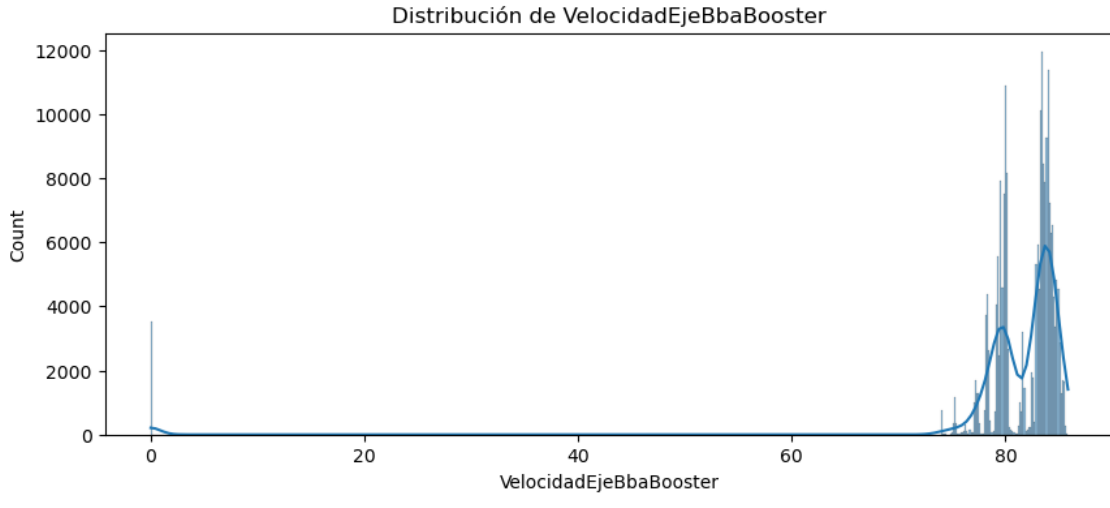


Figura B.8: Histograma Velocidad Eje Bomba Booster R6

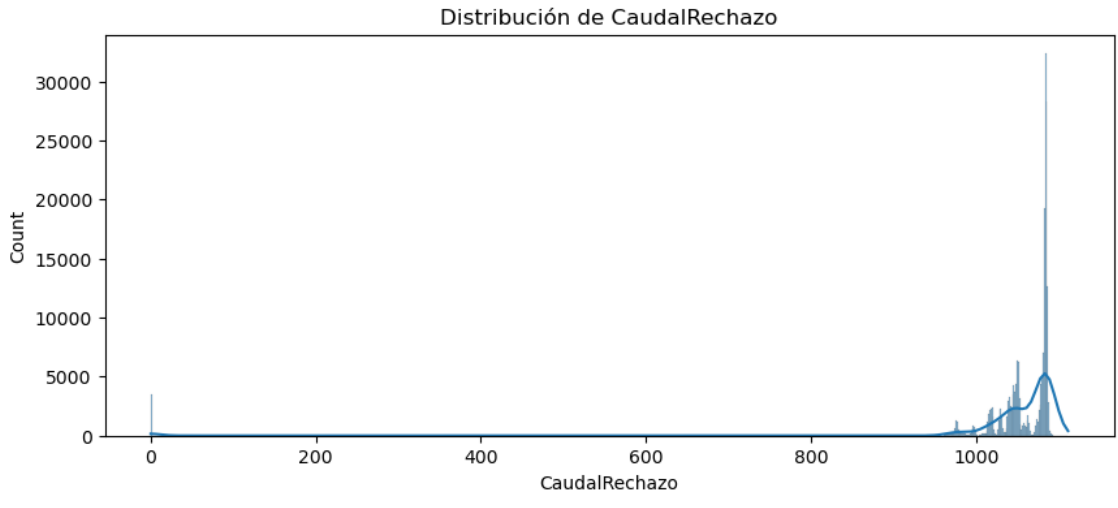


Figura B.9: Histograma Caudal Rechazo Bomba Booster R6

Tabla B.2: Estadísticas Descriptivas de los Sensores de Bombas Booster

	T_Rod_LC	T_Rod_LL	T_CojEmpuje	VibrCojEmpuje	VibrCojRadial	PresionEntrada	VelocidadEje	CaudalRechazo
count	378436.000000	378433.000000	378421.000000	378436.000000	378436.000000	378457.000000	369170.000000	378455.000000
mean	32.601197	53.190699	36.813777	1.048837	1.514282	6042.245466	80.339330	1017.739413
std	1.815227	6.046525	3.406404	0.267072	0.753708	1299.918926	11.282796	216.155139
min	20.361639	20.180819	17.712063	0.000000	0.000000	0.000000	0.000000	0.000000
25 %	31.663996	51.071251	36.572212	0.946441	0.792975	6177.198242	79.797867	1043.698669
50 %	32.499611	54.058136	37.430992	1.081814	1.603723	6345.490234	82.967564	1078.012939
75 %	33.758297	56.926903	38.156101	1.220912	2.153620	6427.960938	83.782530	1084.482666
max	37.878235	62.632183	40.853283	2.695125	4.319718	6989.930664	85.843346	1111.444214