



UNIVERSIDAD
DE PIURA

FACULTAD DE INGENIERÍA

**Diseño de modelo basado en redes neuronales para la
detección de anomalías en motorreductor de molino de secado
en empresas agroindustriales**

Tesis para optar el Título de
Ingeniero Mecánico-Eléctrico

Luis David Arizola Rangel

Asesor:
Dr. Ing. José Manrique Silupu

Piura, diciembre de 2024



Declaración Jurada de Originalidad del Trabajo Final

Yo, Luis David Arizola Rangel, egresado del Programa Académico de Ingeniería Mecánico-Eléctrica de la Facultad de Ingeniería de la Universidad de Piura, identificado(a) con DNI: 73421987, declaro que:

Soy autor del trabajo final titulado:

“Diseño de modelo basado en redes neuronales para la detección de anomalías en motorreductor de molino de secado en empresas agroindustriales”

El mismo que presento bajo la modalidad de Tesis para optar el Título profesional de Ingeniero Mecánico-Eléctrico.

Que el trabajo se realizó en coautoría con los siguientes alumnos de la Universidad de Piura.

- Haga clic o pulse aquí para escribir texto, identificado con Elija un elemento: Escribir número
- Haga clic o pulse aquí para escribir texto, identificado con Elija un elemento: Escribir número
- Haga clic o pulse aquí para escribir texto, identificado con Elija un elemento: Escribir número
- Haga clic o pulse aquí para escribir texto, identificado con Elija un elemento: Escribir número

El texto de mi trabajo final es original y no vulnera los derechos de terceros o, de ser el caso, derechos de los coautores, incluidos los derechos de propiedad intelectual, datos personales, entre otros. En tal sentido, el texto de mi trabajo final no ha sido plagiado total ni parcialmente, para lo cual, he respetado las normas internacionales de citas y referencias de las fuentes consultadas. Asimismo, el texto del trabajo final que presento no ha sido publicado ni presentado antes en cualquier medio electrónico o físico; y que la investigación, los resultados, datos, conclusiones y demás información presentada que atribuyo a mi autoría son veraces.

En caso de detectarse el incumplimiento de lo declarado asumo frente a terceros, la Universidad de Piura y/o la Administración Pública toda responsabilidad que pueda derivarse por el trabajo final presentado. Lo señalado incluye responsabilidad pecuniaria incluido el pago de multas u otros por los daños y perjuicios que se ocasionen.

La asesoría del trabajo estuvo a cargo de los siguientes docentes de la Universidad de Piura:

- Dr. Ing. José Manrique Silupu, identificado con DNI: 44235652
- Haga clic o pulse aquí para escribir texto, identificado con Elija un elemento: Escribir número
- Haga clic o pulse aquí para escribir texto, identificado con Elija un elemento: Escribir número

Declaro (declaramos) que:

Luego de haber empleado el software de coincidencia Turnitin, revisado las fuentes de información señaladas por el autor, y en razón de mi (nuestra) experiencia como investigador(es), declaro (declaramos) que las ideas expuestas en el trabajo final alcanzan las condiciones de calidad, integridad y originalidad acorde a los objetivos institucionales y estándares en materia de investigación. Finalmente, no asumo (asumimos) responsabilidad por la posible vulneración de derechos de autor en el trabajo final referido, pues tal responsabilidad es exclusiva del autor.

Fecha: 19/12/2024.



.....
*Firma del autor*¹



.....
*Firma del asesor*¹

.....
*Firma del co-asesor*¹

.....
*Firma del co-asesor*¹

¹ Firma idéntica al DNI. No se admite digital, salvo certificado.

The watermark is a circular seal of the University of Burgos. It features a central figure holding a sword and scales, surrounded by the Latin text 'UNIVERSITAS • STUDIO • BURGENSE' in a circular arrangement.

Dedicatoria

Dedico este trabajo de tesis a Sarita y Luis, mis padres por ser siempre esa guía que me ha permitido ser una buena persona y profesional. Su apoyo y presencia ha sido muy importante para mí, este logro también es de ustedes.

A mi abuela Elvira quien partió el 2020, sé que hubiese querido estar aquí y donde esté sé que comparte con mucha alegría este logro.

A mi hermana Marife, con quien comparto desde el 2003. Por todo su apoyo emocional y apoyo en no rendirme en este tramo de mi vida.

Al David de 17 años, lleno de sueños y miedos de empezar en ese entonces esta etapa que hoy culmina.

A todos mi familiares y seres queridos que de alguna manera han estado conmigo en esta etapa.

Agradecimientos

En primer lugar, agradezco a Dios por brindarme la fortaleza, salud y oportunidad de alcanzar este logro importante en mi vida.

A Sara y Luis, mis padres por su apoyo incondicional, amor y confianza en cada parte de esta etapa. Sus consejos y empuje han sido fundamentales en mi desarrollo personal y profesional.

A Marife, mi hermana, de quien he aprendido mucho. Su aliento compañía ha sido muy importante en esta etapa. Gracias por la motivación y por recordarme que todo se puede lograr con esfuerzo y dedicación.

Al Ing. José Manrique por su guía y conocimientos compartidos. Su orientación ha sido importante para la realización de este trabajo.

A cada uno de mis compañeros y amigos con los que compartí clases, por compartir diversos consejos, conocimientos y sobre todo experiencias en esta etapa universitaria. Ha sido una experiencia invaluable que siempre recordaré con cariño.

Resumen

Dentro de la industria se suelen trabajar con diversos equipos que realicen algún tipo de accionamiento en específico para algún proceso, dentro de este grupo se encuentran los motores eléctricos de inducción y los reductores de velocidad, que se encargan de transformar la energía eléctrica en mecánica y de aumentar el torque respectivamente. Ambos debido a su robustez y facilidad de mantenimiento cuentan con variedad de aplicación en la industria. Bajo este contexto se tiene como objetivo diseñar un modelo de red neuronal basado en un autoencoder capaz de detectar anomalías en motorreductores utilizados en procesos agroindustriales que garantice una operación continua de estos equipos, además de poder realizar una intervención temprana, evitando mantenimientos correctivos.

Para el diseño del modelo y entrenamiento se parte de la recolección y preprocesamiento de los datos del motorreductor analizados donde se incluyen variables como temperaturas, vibraciones, velocidad, corriente y torque. Se realiza un preprocesamiento de los datos para encontrar valores atípicos, imputar datos faltantes de ser necesario. A continuación, se desarrolla la arquitectura de la red neuronal con los datos normalizados donde este algoritmo logre aprender del comportamiento normal del equipo y replique los datos en su salida, aquí serán identificados los patrones anómalos cuando los datos difieran de manera significativa. Luego se realiza el proceso de evaluación y validación a fin de encontrar la tasa de anomalías detectadas y la precisión de estas detecciones. Finalmente, se analizan los resultados para observar la capacidad del modelo de detectar fallas, así como identificar que tipos de fallas se encontraron. Con esto se logrará demostrar como se puede poner en práctica la red neuronal.

Queda demostrado que el modelo de red neuronal con autoencoder tiene un buen desempeño en la detección de anomalías ya que se obtuvo un error de reconstrucción de 0.023, lo que quiere decir que el modelo es capaz de reconstruir el dato de entrada con un error bajo y por lo tanto ha conseguido aprender adecuadamente de los patrones de operación normal del equipo. Con estos resultados es posible llevar este diseño a una implementación en entornos agroindustriales que permita identificar con antelación eventos de fallos.

Tabla de contenido

Introducción	10
Capítulo 1 Redes Neuronales LSTM	11
1.1 Fundamentos teóricos.....	11
1.1.1 <i>Según el número de capas</i>	11
1.1.2 <i>Según la realimentación</i>	12
1.2 Redes neuronales convolucionales	14
1.3 Motores eléctricos trifásicos	15
1.3.1 <i>Características y funcionamiento</i>	15
1.3.2 <i>Tipos de fallas en motores</i>	18
1.4 Mantenimiento predictivo	22
1.4.1 <i>Análisis de vibraciones</i>	22
1.4.2 <i>Termografía</i>	24
1.4.3 <i>Análisis por ultrasonido</i>	25
1.4.4 <i>Análisis de aceite</i>	25
1.5 Redes neuronales en la detección de anomalías	26
Capítulo 2 Desarrollo e implementación de modelo	30
2.1 Recolección y preprocesamiento de datos	30
2.1.1 <i>Características de motorreductor</i>	32
2.1.2 <i>Preprocesamiento de los datos</i>	34
2.2 Diseño de red neuronal autoencoder	41
2.2.1 <i>Escalado y normalización de datos</i>	41
2.3 Entrenamiento y validación del modelo	44
2.4 Resultados y análisis	47
2.4.1 <i>Métricas evaluadas</i>	57
Conclusiones.....	60
Referencias.....	62
Apéndices	64
Apéndice A. Modelo de Red Neuronal Autoencoder para la detección de anomalías en motorreductores de molinos de secado	65
Anexos	78
Anexo A. Ficha Técnica del motor eléctrico	79

Lista de tablas

Tabla 1 Parámetros de motorreductor	32
Tabla 2 Señales recolectadas de motorreductor	33
Tabla 3 Información del set de datos.....	35
Tabla 4 Información de la división de Train y Test	41
Tabla 5 Identificación de anomalías del modelo autoencoder	48
Tabla 6 Identificación de anomalías del modelo autoencoder – Prueba 2	51
Tabla 7 Importancia SHAP de predicción positiva	55
Tabla 8 Importancia SHAP de predicción anomalía negativa	56
Tabla 9 Percentiles del error medido para modelo.....	59



Lista de figuras

Figura 1 Estructura red neuronal monocapa	11
Figura 2 Estructura de red neuronal multicapa	12
Figura 3 Estructura de red neuronal recurrente	13
Figura 4 Motor eléctrico de inducción	15
Figura 5 Rotor de jaula de ardilla	16
Figura 6 Conexiones en estrella y triangulo en la caja de bornes	17
Figura 7 Desbalanceo estático del rotor	21
Figura 8 Desbalanceo par o acoplado	21
Figura 9 Desbalanceo dinámico	22
Figura 10 Analizador de vibraciones	23
Figura 11 Cámara termográfica	24
Figura 12 Medidor ultrasónico	25
Figura 13 Anomalías dentro de un espacio bidimensional	26
Figura 14 Ejemplo de anomalía contextual	27
Figura 15 Estructura de red neuronal autoencoder	29
Figura 16 Pérdida del poder calorífico de bagazo en relación con su humedad y cenizas	31
Figura 17 Histograma de sensores de motorreductor	37
Figura 18 Tendencias preprocesadas de señales	38
Figura 19 Análisis de correlaciones entre variables	40
Figura 20 Código para normalización y escalamiento de los datos	42
Figura 21 Código para redimensionamiento de los datos	43
Figura 22 Evaluación de función de perdida en entrenamiento y validación	45
Figura 23 Grafico de función de perdida de modelo	45
Figura 24 Histograma de la distribución de función de pérdida	46
Figura 25 Creación de señal binaria de anomalía	48
Figura 26 Importancias SHAP sobre modelo autoencoder	50
Figura 27 Importancias SHAP sobre modelo autoencoder – Prueba 2	52
Figura 28 Gráfico resumen de SHAP – Modelo autoencoder	53
Figura 29 Gráfico SHAP Waterfall – Anomalía positiva	54

Figura 30 <i>Gráfico SHAP Waterfall – Anomalía negativa</i>	55
Figura 31 <i>Cálculo de error cuadrático medio</i>	58
Figura 32 <i>Cálculo de la desviación estándar</i>	58



Introducción

En el ámbito agroindustrial, los procesos productivos destacan por su alta complejidad, ya que implican la interacción de múltiples variables que deben ser controladas con precisión para garantizar tanto la eficiencia operativa como la calidad del producto final. En el Perú, la agroindustria es un sector clave de la economía nacional. Según el Ministerio de Economía y Finanzas (MEF), en 2021, este sector representó el segundo mayor contribuyente a las exportaciones, con envíos que superaron los US\$ 8 mil millones. Además, genera cerca de dos millones de empleos, entre directos, indirectos e inducidos, consolidándose como un pilar económico y social del país.

Teniendo en cuenta este contexto, es importante que se garantice la continuidad operativa de los equipos industriales de una planta en donde se busque evitar las interrupciones que afecten a la producción. Por su amplia aplicación es común encontrar motores eléctricos y cajas reductoras ya sea planetario o de ejes paralelos, los cuales son activos críticos ya que por su conversión de energía eléctrica en energía mecánica impulsan procesos como transporte, procesamiento y almacenamiento de materia prima. La detección temprana de anomalías en estos equipos se posiciona como una estrategia efectiva para prevenir fallas inesperadas, reducir los costos asociados a reparaciones correctivas, asegurar la calidad del producto y evitar paradas no programadas.

La aplicación de tecnologías avanzadas como el machine learning y la inteligencia artificial son herramientas capaces de estudiar los algoritmos o modelos posibles dentro de un conjunto de datos imitando la forma en como aprenden los seres humanos, tratando de mejorar continuamente su precisión. Cabe resaltar que existen diversos tipos de algoritmos de machine learning entre los que se encuentran los algoritmos de regresión, supervisados, árboles de decisión, etc. Para el presente trabajo se propone un algoritmo de Redes Neuronales LSTM Autoencoder, que es un tipo de red no supervisado basado en la compresión y descompresión de los datos de entrada, generando una reconstrucción en la salida.

Uno de los principales retos que se presentan corresponde a la recolección de los datos, especialmente con las etiquetas que para un modelo de red neuronal son las que brindan información de eventos de fallos ocurridos en alguna planta industrial, ya que sucede en muchos casos que no se cuenta precisamente con esta información por parte del equipo de mantenimiento. Equipos como los PLC's o sistemas SCADA pueden dar alcance de esto. Según la revista MIT Technology Review Insights (2023) estos controladores y sistemas contienen grandes volúmenes de datos de máquinas que beneficiarían a los modelos de inteligencia artificial. Ante esta problemática, el autoencoder propone una solución interesante con buenos resultados mediante la fijación de umbrales para la detección de anomalías.

Capítulo 1

Redes Neuronales LSTM

En los últimos años, las redes neuronales artificiales con el avance de la tecnología han ido recibiendo cada vez un mayor interés respecto a la minería de datos o también llamado “Big Data” ya que, esta se presenta con los medios necesarios para modelar problemas complejos de una manera efectiva y eficiente. Estos modelos al ser empleados con datos son capaces de encontrar relaciones o patrones de manera inductiva a través de algoritmos de aprendizaje que precisamente se basan en los datos de entrada.

Dentro de este primer capítulo se abordan los fundamentos teóricos para tener en cuenta de redes neuronales y su aplicación en la detección de fallas, dando un enfoque a las fallas que se puedan detectar en un motorreductor utilizado en un molino de secado del sector agroindustrial.

1.1 Fundamentos teóricos

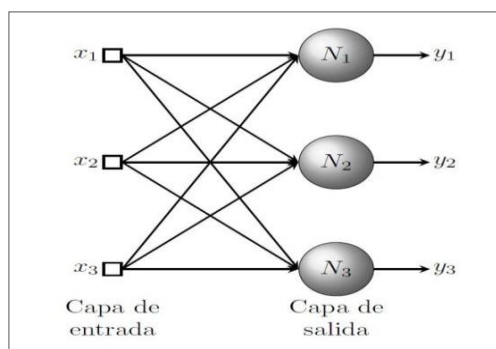
Las Redes Neuronales Artificiales (RNA) se presentan como alternativa en la resolución de problemas como los de clasificación, identificación, optimización, diagnóstico y predicción especialmente cuando se cuenta con una gran cantidad de datos. Estos sistemas tratan de emular el comportamiento del cerebro humano a través del modelamiento de estructuras de procesamiento o mediante la construcción de arquitecturas físicas aproximándose a una red neuronal biológica.

Las redes neuronales se clasifican de la siguiente manera:

1.1.1 Según el número de capas

Según este tipo de clasificación las redes neuronales pueden ser monocapas, es decir, que consta de una sola capa donde la capa de salida proyecta a la capa de entrada, esta suele utilizarse para eliminar distorsiones en señales.

Figura 1
Estructura red neuronal monocapa

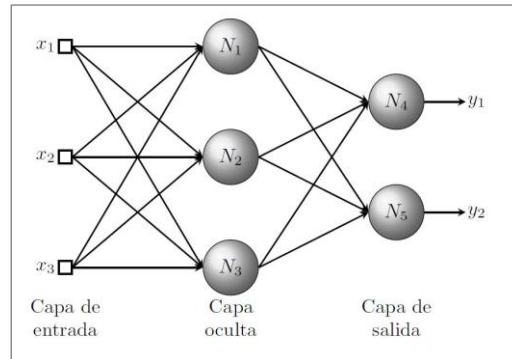


Nota. De Inteligencia Artificial (s.f.).
Clasificación de las Redes Neuronales”

También dentro de esta clasificación se encuentran las redes neuronales multicapa, que, a diferencia del primer caso, estas cuentan con una capa intermedia u oculta entre la entrada y la salida. Este tipo de red puede estar parcial o totalmente conectada.

Figura 2

Estructura de red neuronal multicapa

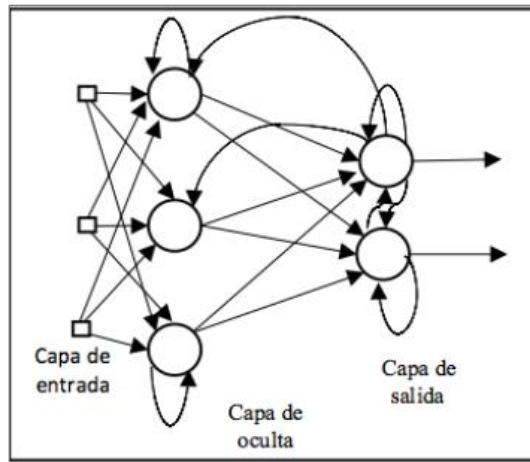


Nota. De Inteligencia Artificial (s.f.).
Clasificación de las Redes Neuronales”

1.1.2 Según la realimentación

Teniendo en cuenta esta clasificación, existen las redes neuronales no recurrentes que se caracterizan por no existir algún tipo de realimentación, esto porque se mueven siempre hacia adelante.

En contraste, se encuentran las **redes neuronales recurrentes (RNN)**, que se caracterizan por permitir la realimentación mediante la utilización de lazos. Esto significa que el cálculo de una entrada puede depender del paso anterior e, incluso, en algunas ocasiones, del paso futuro. Dicho de otra manera, en cada instante de tiempo t esta neurona recurrente recibe una entrada x que a su vez ha sido una salida \hat{y} en el paso anterior, resaltar que, para la primera neurona o primer paso, la entrada generalmente se establece como 0. Estos lazos pueden conectar neuronas dentro de distintas capas ocultas o, incluso, dentro de la misma capa, como se muestra en la figura 3.

Figura 3*Estructura de red neuronal recurrente*

Nota. Tomado de Redes Neuronales Artificiales Aplicadas al Reconocimiento de Patrones (p. 20), por Soria & Blanco, 2007

Se crea una capa de neuronas recurrentes, en donde cada paso de tiempo, cada neurona recibe tanto el vector de entrada como el vector de salida del paso anterior de tiempo. De esta manera, se tiene en cuenta que tanto las entradas como las salidas ahora son vectores y cuando solo hay una neurona, la salida será un escalar.

Cada neurona recurrente cuenta con dos conjuntos de pesos: uno para las entradas x y otro destinado para las salidas del paso anterior \hat{y} . Existen vectores de peso w_x y w_y que consideran toda la capa recurrente en lugar de solo tomar una neurona. Esto se puede organizar mediante matrices de peso W_x y W_y que consideren todos los vectores de peso de la red.

Una vez realizado lo especificado anteriormente, se calcula el vector de salida de toda la capa recurrente, este se puede representar mediante la siguiente ecuación:

$$\hat{y}_t = W_x * x(t) + W_y * \hat{y}(t - 1) + b$$

Estas redes tienen la capacidad de realizar una amplia variedad de funciones, como el procesamiento de secuencias temporales, la proyección de una trayectoria, predicciones no lineales y la representación de sistemas dinámicos complejos. Debido a su capacidad para procesar secuencias temporales, también se les conoce como redes espacio – temporales o redes dinámicas. Su principal propósito es relacionar tanto secuencias de entrada como de salida, procesándolas como patrones que varían a lo largo del tiempo.

Dentro de las tareas esenciales que pueden ser realizadas por las RNN está el reconocimiento de secuencias, es decir, que la red neuronal es capaz de identificar un patrón de salida específico cuando se le proporciona una secuencia de entrada. Esto suele ser aplicado en la clasificación de secuencias de series temporales o el reconocimiento de patrones activos. También se encuentra la aplicación en la reproducción de secuencias, en donde la red neuronal es capaz de continuar una secuencia con tan solo observar parte de ella, generando el resto de la serie. Esto es útil en aplicaciones como la generación de texto o la predicción de eventos futuros en una serie temporal.

Finalmente, otra tarea realizada es la asociación temporal, en donde se genera una secuencia de salida específica como respuesta a una secuencia de entrada determinada, esta tarea es clave en aplicaciones donde es importante generar una salida que varíe de acuerdo a la dinámica temporal de la entrada, al control de procesos o la respuesta de señales complejas.

Para el entrenamiento de una RNN es necesario un desarrollo en el tiempo que utilice la propagación hacia atrás regular. A esta estrategia de propagación se le conoce como retropropagación a través del tiempo (BPTT). Esta estrategia consiste en dar un paso hacia adelante a través de la red desarrollada, a continuación, se hace la evaluación de la secuencia de salida utilizando una función de pérdida. Se debe de tener en consideración que la función de pérdida puede llegar a ignorar algunas salidas, como el caso de una secuencia de vector RNN donde se ignoran todas las salidas a excepción de la última.

1.2 Redes neuronales convolucionales

Las redes neuronales convolucionales (CNN) son un tipo de redes neuronales artificiales dentro del campo del machine learning. Estas redes son capaces de identificar diversas características de una entrada mediante el uso de capas ocultas especializadas. Las CNN emplean varias capas de convolución organizadas jerárquicamente. Las primeras capas suelen identificar características simples o patrones básicos, como bordes o texturas. A medida que se avanza hacia capas más profundas, estas capas son capaces de reconocer patrones más complejos, como la estructura de un rostro o la forma de una silueta.

Núñez, F. (2016) en su tesis de maestría titulada "*Diseño de un sistema de reconocimiento automático de matrículas de vehículos mediante una red neuronal convolucional*" menciona que, la arquitectura de una red neuronal convolucional (CNN) esta compuesta de varias capas, donde cada una de estas cumple una un rol en específico respecto al procesamiento de datos de entrada como las imágenes. Dentro de estas capas incluye las capas convolucionales, *pooling* y completamente conectadas.

La primera capa se encarga de realizar la operación de convolución dentro de un conjunto de filtros o kernels que se desplazan sobre la imagen de entrada para la extracción de características específicas como bordes o texturas. Cada uno de estos filtros aprende a detectar diferentes patrones a lo largo del proceso de entrenamiento.

Respecto a las capas de *pooling*, conocidas como capas de submuestreo o agrupamiento, son utilizadas para reducir la dimensionalidad espacial de las características extraídas por las capas convolucionales. Esto se logra tomando el valor más alto (*max-pooling*) dentro de una región específica de la imagen, lo que les reduce tamaño a los mapas de características mientras persevera la información más significativa. Así mismo, estas capas ayudan a que el modelo sea más resistente a cambios sutiles en la ubicación de las características identificadas.

En las últimas etapas de la red, se encuentran las capas completamente conectadas, que funcionan de manera similar a las de una red neuronal tradicional. Estas capas toman características procesadas por las capas previas y las integran para producir la clasificación final. El resultado es un conjunto de probabilidades que indican la pertenencia de la entrada a cada una de las posibles categorías.

1.3 Motores eléctricos trifásicos

Los motores eléctricos trifásicos de inducción son dispositivos electromecánicos que se encargan de convertir la energía eléctrica en energía mecánica, utilizando un sistema de corriente alterna. Se aplican ampliamente dentro de la industria debido a la eficiencia, durabilidad y capacidad de poder generar potencias elevadas.

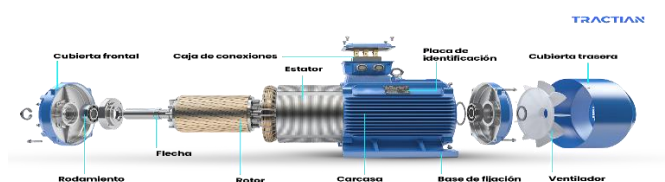
1.3.1 Características y funcionamiento

1.3.1.1 Características. Un motor eléctrico es conocido como una máquina de inducción, se les denomina de esta forma debido a que el voltaje en el rotor es responsable de generar la corriente y el campo magnético del mismo, esto por la inducción en los devanados del rotor, en lugar de estar conectado de forma directa por cables. La característica principal de esta máquina es que no requiere de una corriente continua (DC) para su funcionamiento.

Estas máquinas de inducción es posible utilizarlas tanto como motor como generador, sin embargo, presenta desventajas como generador ya que pierde eficiencia en la conversión al no estar optimizado para este proceso, lo que reduce su eficiencia al convertir energía mecánica de vuelta a eléctrica.

Figura 4

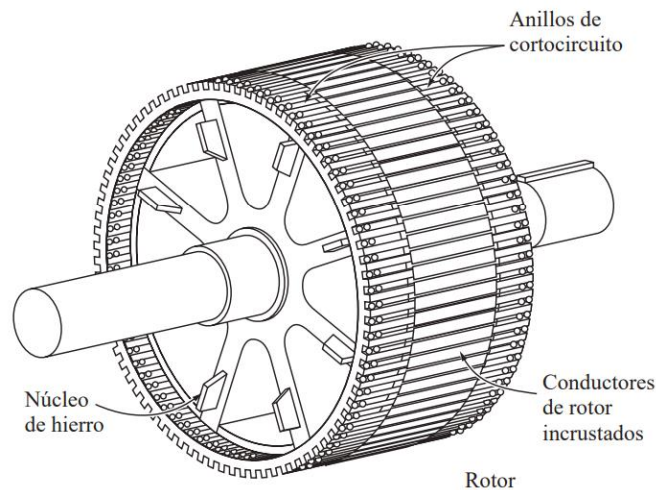
Motor eléctrico de inducción



Nota. De Tractian. (s.f.). “¿Cómo funcionan los motores eléctricos”

Un motor eléctrico se compone físicamente de un estator, similar al de la máquina asíncrona, pero con un rotor construido de diferente manera. En cuanto a esta construcción existen dos tipos de rotores diferentes: rotor de jaula de ardilla y otro conocido como rotor devanado.

Figura 5
Rotor de jaula de ardilla



Nota. Tomado de Máquinas Eléctricas (p. 232), por J. Chapman, 2012, McGraw- Hill

La figura 5 muestra el rotor de jaula de ardilla. Este tipo de rotores dentro de los motores de inducción son comúnmente utilizados en la industria debido a su simplicidad, robustez y bajo rendimiento. Consiste en una serie de barras conductoras de material como aluminio o cobre colocadas dentro de las ranuras en la superficie del rotor. Estas se encuentran unidas en ambos lados por anillos conductores, creando así un circuito cerrado. Este diseño recibe el nombre de “jaula de ardilla” debido a solo se observa un conjunto de barras y anillos que se asemeja a una rueda de ejercicio utilizada por ardillas.

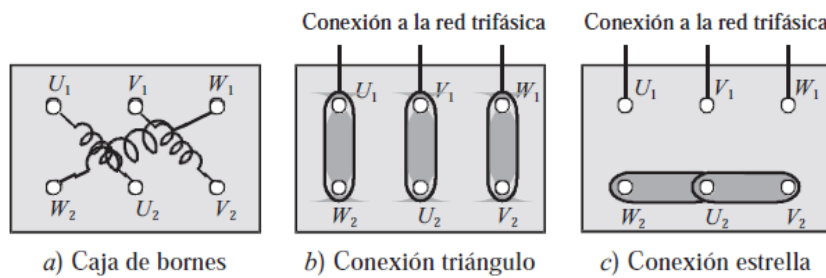
El rotor de jaula de ardilla presenta como principal ventaja su durabilidad y que utiliza pocos componentes móviles que requieren de mantenimiento, como escobillas o anillos rozantes. Esto lo convierte en una buena opción en aplicaciones donde se requiere una mayor confiabilidad con un mantenimiento menor. Así mismo, este diseño garantiza un arranque rápido asegurando un funcionamiento estable en distintas condiciones de operación.

Estos motores, además de contar con sus componentes principales como el rotor y el estator, también cuenta con otros elementos mecánicos que son necesarios para el funcionamiento adecuado de estos equipos. Entre estos se encuentran las tapas (trasera y delantera), rodamientos, carcasa, anillos, etc. En los motores con potencia mediana o elevada suele existir un ventilador en el eje, esto con el propósito de disipar el calor generado por la máquina debido a las pérdidas que aparecen en el motor.

Otro de los elementos importantes a considerar en estos motores es la caja de conexiones, ya que en los motores de inducción trifásicos existen distintos tipos de disposición en la caja de bornes de la máquina. A esta caja se llevan los extremos de los bobinados tal como se muestra en la figura 6.:

Figura 6

Conexiones en estrella y triangulo en la caja de bornes



Nota. Tomado de máquinas eléctricas (p. 291), por J. Fraile M., 2008, McGraw- Hill

De acuerdo con los esquemas mostrados anteriormente es importante resaltar que los terminales de la misma fase no se encuentran enfrentados en la regleta de bornes. La disposición empleada da la facilidad a la conexión del motor a través de láminas de latón. Como se muestra, para una conexión en triángulo el terminal U_1 con W_2 van conectados, de la misma manera V_1 con U_2 y W_1 con V_2 . En cambio, para una conexión en estrella los terminales W_2, V_2 y U_2 se conectan entre sí.

Las conexiones en estrella y delta tienen sus respectivas aplicaciones, ya que, se emplea una conexión en estrella cuando se requiere que la máquina se conecta a una tensión más elevada que la que se indica en su placa característica, en cambio, se utiliza de una conexión en triángulo para una tensión aplicada menor que en el primer caso. En el caso de la conexión a estrella se tiene la ventaja que el motor consume una menor corriente durante el arranque lo que es útil para evitar picos que puedan dañar otros equipos o que se active alguna protección eléctrica, mientras que en la conexión en triángulo al recibir cada devanado la tensión de línea permite que este desarrolle su máxima potencia. Esto es necesario cuando se requiere de un mayor esfuerzo por parte del motor para mover cargas mayores. La combinación de estas dos estrategias ofrece un funcionamiento eficaz para el arranque y a su vez entrega la máxima potencia ofrecida por el motor.

1.3.1.2 Principio de funcionamiento. El funcionamiento de estas máquinas según su uso común se basa en la transformación de la energía eléctrica en energía mecánica para el movimiento de cargas. El motor está comprendido por el estator que es la parte fija, la cual está conformada por tres devanados que se encuentran desfasados a 120° entre si distribuidos espacialmente y dos pares de polos. Cuando se aplica una corriente trifásica a una

cierta frecuencia se origina una onda rotativa de fuerza de excitación magnética (f.m.m.) que se distribuye de manera sinusoidal a lo largo del entrehierro (espacio entre estator y rotor), lo que da lugar a un campo magnético giratorio, siendo la velocidad mecánica expresada de acuerdo con la siguiente fórmula:

$$n_1 = \frac{60f_1}{p} \text{ [rpm]}$$

Donde f_1 representa la frecuencia de alimentación en Hz , p hace referencia al número de polos que hay en el motor y n_1 representa la velocidad a la que rota el campo magnético o también llamada velocidad de sincronización. En la fórmula se describe el comportamiento de la velocidad de sincronismo la cual aumenta con la frecuencia de la corriente aplicada y disminuye a mayor número de polos. Este campo magnético genera un flujo que va variando en las fases del motor, induciendo en su circuito eléctrico una fuerza magnetomotriz la cual se describe por la Ley de Inducción de Faraday, representada por la siguiente fórmula:

$$e = -N \frac{d\phi}{dt}$$

Donde e representa a la fuerza electromotriz, N es la cantidad de vueltas del devanado y $\frac{d\phi}{dt}$ es la razón de cambio temporal del flujo magnético. Dado que el circuito eléctrico del motor se encuentra cerrado, la combinación de estas corrientes con el campo magnético rotatorio da lugar a fuerzas que actúan sobre los conductores del rotor, lo que genera un par motor capaz de hacer girar el rotor en la dirección del campo magnético, sin embargo, la fuerza electromotriz se considera nula si el rotor alcanzara la velocidad de sincronismo, haciendo que no circule corriente por el rotor.

1.3.2 Tipos de fallas en motores

Los motores eléctricos de inducción son activos críticos para la mayoría de las plantas a nivel industrial debido a su variedad de aplicaciones. Estas fallas pueden suceder en cualquier parte del motor, incluyendo el estator, el rotor, los rodamientos y los sistemas de alimentación. Los problemas más comunes comprenden desde cortocircuitos en los devanados del estator y desgaste en los rodamientos, hasta fallas por sobrecalentamiento y desequilibrio de las fuentes de alimentación eléctrica. Es importante la detección temprana de estos fallos ya que el tiempo de inactividad ocasionado puede generar significativas pérdidas en la producción y aumentar los costos operativos.

1.3.2.1 Fallas por barras rotas. La detección de barras rotas en el rotor es una de las fallas más importantes en los motores eléctricos, siendo estudiada con mayor detenimiento debido a la dificultad de identificar sus etapas iniciales. En la mayoría de los casos el motor continúa operando normalmente sin mostrar signos claros del problema, lo que hace difícil localizarlo en primera instancia. Sin embargo, una barra rota se puede manifestar como un aumento en el consumo de corriente y causar daños potenciales en el estator. En el peor escenario esto puede ocasionar una destrucción irreversible del motor. Además, estos daños generan vibraciones que afectan a los componentes acoplados al motor como acoplamientos, bandas de transmisión o engranajes.

Bajo condiciones normales, las corrientes simétricas en un rotor en buen estado se induce un campo magnético giratorio que va a la velocidad de sincronismo. Sin embargo, la ruptura de una barra genera un campo magnético adicional en la dirección contraria a la velocidad de deslizamiento. Esto provoca la aparición de oscilaciones en el espectro de frecuencia que afectan tanto a la velocidad como al torque del motor.

1.3.2.2 Fallas en rodamientos. Según el fabricante NSK, los rodamientos son elementos muy útiles en maquinaria ya que con su movimiento giratorio facilita el movimiento, apoya y guía componentes que giran entre sí, además que reduce la fricción entre elementos móviles. Sin embargo, los diversos problemas en estas piezas como defectos en las pistas, bolas o rodillos de los rodamientos generan vibraciones de alta frecuencia. Estas vibraciones también pueden ser generadas debido a tolerancias y fricciones en sus componentes internos.

Las principales causas de fallas en rodamientos son las siguientes:

- Sobrecarga
- Falta de lubricación o exceso de esta
- Contaminación externa
- Montaje inadecuado
- Defectos del fabricante
- Presencia de vibraciones durante el estado de reposo del cojinete
- Flujo de corriente a través del cojinete

En la etapa inicial de un cojinete de rodillos en deterioro, suelen formarse pequeñas grietas en la parte superior de los rodamientos y en las pistas, estas son difíciles de captar por el ojo humano. Sin embargo, estas grietas comienzan a producir vibraciones y aumento de la temperatura, que en muchos casos no son fáciles de percibir. En este punto, no es lo más recomendable realizar la sustitución del cojinete, sin embargo, se puede comenzar a observar dicha tendencia y mediante diversas estrategias de mantenimiento, estimar en cuanto tiempo tardará en fallar dicha pieza.

Además, cuando se presenta un defecto en la pista externa del rodamiento, se generan impulsos cada vez que una bola entra en contacto con la zona defectuosa. La velocidad de rotación del rotor es la misma velocidad de rotación de la pista interna, ya que se encuentran girando al mismo ritmo que el eje. Aunque la carga radial inicial se distribuye entre varias bolas durante la operación, cuando un defecto en una superficie del cojinete golpea en una superficie complementaria, se genera un impulso que excita las resonancias del sistema mecánico. Estos impactos ocurren de manera cíclica a una frecuencia específica, denominada frecuencia de paso de bolas sobre la pista exterior (f_{BPOF}), que viene calculada por la siguiente expresión:

$$f_{BPOF} = \frac{N_B}{2} f_r \left(1 - \frac{D_b}{D_c} \cos\theta \right)$$

Donde θ es el ángulo de contacto entre la superficie de la bola, D_c es el diámetro del rodamiento, D_b el diámetro de bola y N_b representa el número de bolas del rodamiento.

Generalmente, cuando ya es posible visualizar el agrietamiento en estos cojinetes, estas se traducen en vibraciones de mayor magnitud, ya que es posible detectarlo mediante el ruido que se genera, además con una inspección termográfica es posible mirar el aumento de temperatura.

1.3.2.3 Fallas por desbalance. El desbalance ocasionado en el rotor de un motor eléctrico dentro de la industria suele ser un tipo de anomalía común. Se entiende que para que exista una condición de desbalance o desequilibrio debe de existir una distribución excéntrica en la masa del rotor, si bajo estas condiciones comienza a girar, se generará una fuerza centrífuga a partir de la rotación, la cual repercute como una fuerza adicional sobre los rodamientos y chumaceras. El desbalance puede clasificarse de la siguiente manera:

- Desbalance estático
- Desbalance dinámico
- Desbalance par

En el desbalance de un rotor es común identificarla en su espectro de vibración como una frecuencia de rotación en donde presenta un pico con amplitud elevada, lo que puede llegar a afectar de manera significativa en el estado general de la maquinaria. Cuando el motor está mecánicamente desbalanceado, se generan pequeñas vibraciones radiales en la estructura del rotor. Estas vibraciones alcanzan su máximo nivel cuando la velocidad de rotación coincide con la velocidad natural del sistema, la cual viene dada por la siguiente expresión:

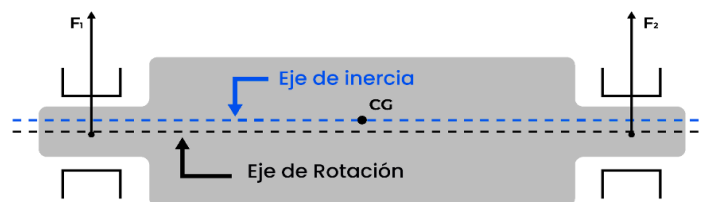
$$\omega_0 = \sqrt{\frac{k}{m}}$$

Donde k es el coeficiente de rigidez del sistema y m es la masa desbalanceada. La señal vibratoria oscila a la misma frecuencia de rotación del rotor, incrementándose a medida que lo va haciendo la amplitud de la frecuencia de rotación, así como sus armónicos. El análisis de estas señales vibratorias es una forma rápida y efectiva de diagnosticar la presencia de desbalance en motores de inducción.

Dentro de las clasificaciones de desbalanceo antes mencionadas, se encuentra el desbalanceo estático, el cual se define como el desplazamiento paralelo del eje principal de inercia con respecto al eje geométrico de rotación. Este tipo de desbalanceo se observa en la Figura 7:

Figura 7

Desbalanceo estático del rotor

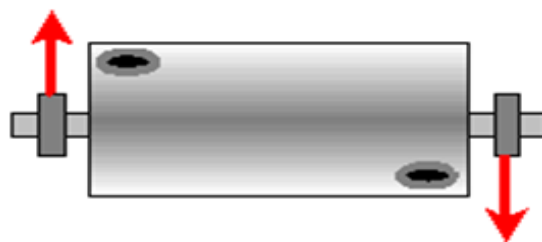


Nota. De Tractian. (s.f.). “Fallas de motores eléctricos: Conoce las 10 causas más comunes y como evitarlas”

El siguiente caso en mención es el desbalanceo par o acoplado el cual se define como la intersección en el centro de gravedad de la masa entre el eje principal de inercia y el eje geométrico del motor. Este tipo de desbalanceo se observa en la figura 8:

Figura 8

Desbalanceo par o acoplado

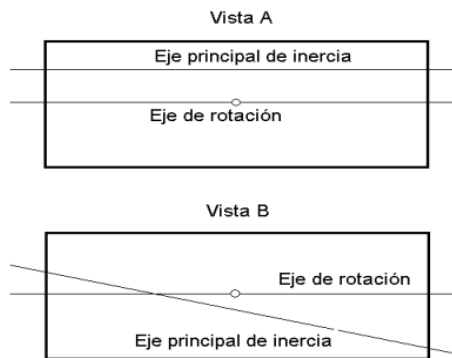


Nota. Tomado de Predictiva21 (s.f.). “Desbalanceo” 2021.

El último caso de tipo de desbalanceo mencionado es el dinámico, el cual se presenta como la no existencia de intersección entre el eje principal de inercia y el eje geométrico del rotor ni tampoco la existencia de paralelismos entre estos dos ejes. Este tipo se observan en la figura 9:

Figura 9

Desbalanceo dinámico



Nota. La figura considera caso de desbalanceo dinámico.

1.4 Mantenimiento predictivo

El mantenimiento predictivo es una estrategia avanzada de gestión de activos que se apoya de datos en tiempo real y análisis para la evaluación de posibles fallos en equipos o sistemas. Consta de una variedad de ensayos no destructivos enfocados a realizar un seguimiento a los equipos para poder revisar signos de advertencia o diagnosticar que es lo que está sucediendo con el comportamiento de la máquina.

Con este tipo de mantenimiento es posible la detección de anomalías en etapas tempranas, lo cual permite colaborar con una mejor programación de las reparaciones correspondientes sin llegar a cortar el proceso de producción, alargando el tiempo de vida de los activos.

Entre los ensayos más comunes se tienen los siguientes:

1.4.1 Análisis de vibraciones

Esta estrategia de análisis está basada en el estudio de funcionamiento de las máquinas rotativas a través del comportamiento de sus vibraciones.

Cada una de las máquinas experimentan un cierto nivel de vibración, incluso cuando esta se encuentra bajo parámetros de operación normal. Sin embargo, ante la presencia de una anomalía, estos niveles de operación varían, llegándose a alterar, lo que indica que es necesario realizar una inspección del equipo.

Lo que se debe de tener en cuenta para la validación de este método es conocer los datos técnicos del equipo a intervenir como: la velocidad de rotación, tipo de rodamientos de motor, correas, número de álabes. Además, es fundamental definir en que puntos se realizarán las mediciones y contar con el equipo idóneo para para este análisis.

Figura 10

Analizador de vibraciones



Nota. Tomado de Técnicas de mantenimiento predictivo utilizado en la industria (p. 224), por W. Olarte C, M. Boltero A y B. Cañon A., 2010.

Las vibraciones son analizadas mediante la medición de su amplitud y realizando una descomposición en el espectro de frecuencia. En el caso que se observe que la amplitud de la vibración supera los límites permitidos, lo que indica que algo no funciona correctamente y se necesita realizar una intervención en el equipo en cuestión.

Dentro de los problemas que se pueden detectar mediante el análisis de vibraciones, se tienen los siguientes:

- Desbalance
- Resonancia
- Desalineamiento
- Solturas mecánicas
- Rodamientos dañados
- Anomalías en engranajes
- Problemas eléctricos en motores
- Problemas en bandas

1.4.2 Termografía

La técnica de la termografía estudia el comportamiento de la temperatura en una máquina, esto con el propósito de identificar posibles aumentos o verificar que el equipo viene operando bajo buenas condiciones.

La energía que emiten estos equipos mirados desde su punto superior se propaga en forma de ondas electromagnéticas que viajan a la velocidad de la luz. Esta emisión se encuentra directamente relacionada con la temperatura de la máquina, lo que significa que mientras se tenga mayor temperatura, la energía liberada va a ser mayor. Ya que estas ondas presentan una longitud que excede la capacidad de percepción del ojo humano, es necesario un instrumento que sea capaz de convertir esta energía en un rango de espectro visible, facilitando así su análisis. En la figura 11 se evidencia como el instrumento genera una imagen de radiación infrarroja partiendo de la temperatura superficial de la máquina:

Figura 11
Cámara termográfica



Nota. Tomado de Técnicas de mantenimiento predictivo utilizado en la industria (p. 224), por W. Olarte C, M. Boltero A y B. Cañon A., 2010.

Gracias a esta técnica que viene dada por imágenes térmicas que proporcionan estas cámaras, es posible analizar las variaciones de temperatura, ya que, con este tipo de incremento usualmente suelen diagnosticarse fallas electromecánicas dentro de algún componente del equipo.

Estas cámaras termográficas suelen tener aplicaciones en los siguientes campos:

- Estructuras refractarias
- Instalaciones eléctricas
- Equipos mecánicos

1.4.3 Análisis por ultrasonido

Esta estrategia de mantenimiento predictivo se basa en el comportamiento de las ondas sonoras a altas frecuencias producidas por las situaciones problemáticas.

El oído humano está diseñado para detectar sonidos cuyas frecuencias se encuentren dentro del rango de 20 Hz a 20 kHz. Por esta razón, cuando un componente de algún equipo presenta un fallo, el sonido generado no es percibido, ya que esta frecuencia supera los 20 kHz.

Las ondas ultrasónicas se atenúan rápidamente debido a su corta longitud, lo que facilita la identificación de la fuente que los genera, incluso en entornos donde el ambiente es muy ruidoso.

Los dispositivos que convierten estas ondas ultrasónicas en ondas audibles se conocen como medidores de ultrasonido o detector ultrasónico. A través de estos equipos las señales ultrasónicas son transformadas de tal forma que puedan ser escuchadas mediante audífonos o visualizados desde una interfaz, así como se observa en la figura 12:

Figura 12

Medidor ultrasónico



Nota. Tomado de Técnicas de mantenimiento predictivo utilizado en la industria (p. 225), por W. Olarte C, M. Boltero A y B. Cañon A., 2010.

Este análisis ultrasónico permite detectar fallos desde fricciones en máquinas rotativas, fugas en válvulas, inspección de estado de lubricación, pérdidas de vacío y fugas en tuberías.

1.4.4 Análisis de aceite

El análisis de aceite realiza la evaluación del estado operativo de las máquinas mediante el estudio de propiedades físicas y químicas del aceite lubricante, además de controlar el estado de contaminación de aceite y presencia de partículas ferrosas. Esta técnica es una de las principales herramientas para detectar fallos prematuros en equipos.

El aceite es un componente fundamental para los equipos, ya que constantemente por el movimiento y/o rotación de los componentes, se ven expuestos a fricciones y desgastes, además que el aceite controla la temperatura y elimina las impurezas. Sin embargo, una vez que el aceite se degrada o se encuentra altamente contaminado, este pierde sus propiedades, por lo que ya no puede cumplir con sus funciones, provocando fallos en los equipos.

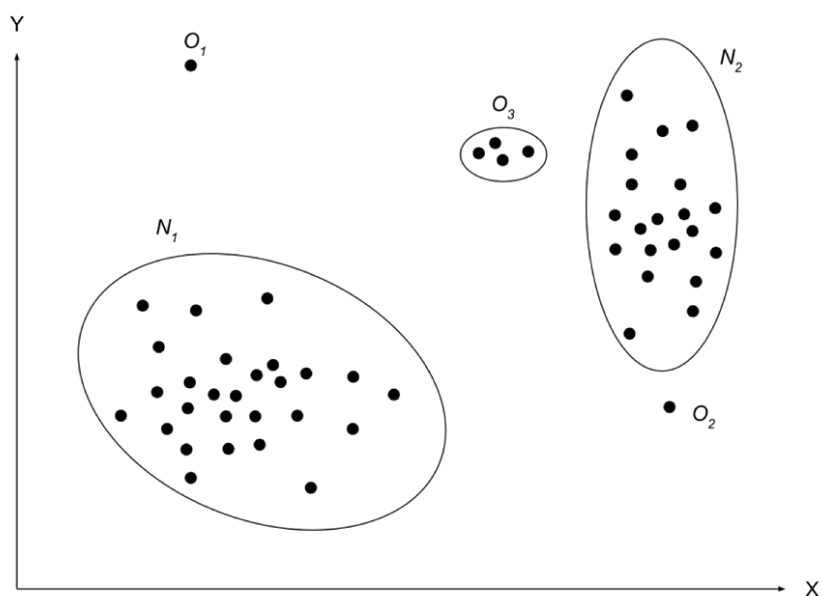
1.5 Redes neuronales en la detección de anomalías

Dentro del ámbito industrial, donde se necesita una operación continua además de optimizar los tiempos de inactividad no planificados por el equipo de mantenimiento, resulta fundamental poder detectar tempranamente anomalías o fallos en equipos industriales como motores eléctricos, reductores y demás componentes dentro de una fábrica. Los métodos de monitoreo y mantenimiento habitualmente empleados, aunque resultan efectivos, presentan limitaciones respecto a la identificación de patrones complejos o sutiles que puedan anteceder a una falla. En este contexto, el uso de redes neuronales surge como una herramienta poderosa, ofreciendo una mayor precisión en la detección de comportamientos anómalos mediante el análisis de datos históricos y en tiempo real.

Una anomalía se puede definir como una observación o serie de datos que se desvían notablemente de sus condiciones normales operativas, que no guarda relación con una tendencia o rango de normalidad establecido. En la figura 13 se pueden ilustrar las anomalías dentro de un espacio bidimensional:

Figura 13

Anomalías dentro de un espacio bidimensional



Nota. De "A note about finding anomalies," por K. Cao, 2020, Towards Data Science.

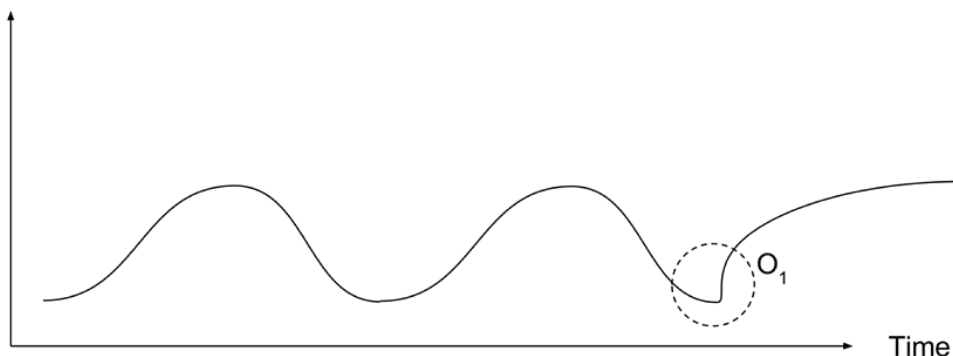
De la figura tomada de "A note about finding anomalies" se observa que N_1 y N_2 son regiones donde se muestran datos normales, ya que la mayoría de los puntos se encuentran alrededor de estas. Los puntos O_1 , O_2 y la región O_3 son considerados como anomalías ya que se encuentran alejados de las zonas de comportamiento normal. Un ejemplo representativo puede ser donde uno o varios parámetros de un motor, reductor, etc. se desvían de sus valores normales ya sea por distintos motivos como incremento de vibraciones, sobrecargas térmicas o desbalances de fase. Por lo tanto, la detección de anomalías es un proceso importante, por lo que se considera una ventaja en diversos sistemas de toma de decisiones.

Las anomalías pueden clasificarse dentro de las tres categorías siguientes:

- **Anomalías puntuales:** cuando un objeto destaca frente a otros como una anomalía, este se clasifica como una anomalía puntual. Este es el tipo de anomalía más fácil de identificar y es común que muchas investigaciones lo incluyan en su análisis. En la figura 13 los puntos O_1 y O_2 son ejemplos representativos de anomalías puntuales.
- **Anomalías contextuales:** Se considera una anomalía contextual cuando se presenta un comportamiento anómalo en contexto específico, conocido también como una anomalía condicional. En la figura 13 se observa un contexto periódico en el cual el punto O_1 es considerado como una anomalía debido a que se desvía de un patrón periódico específico.
- **Anomalías colectivas:** Se presentan cuando un grupo de objetos relacionados se distingue como anómalo en comparación de otros. Bajo este contexto, no es el único el que se considera en anomalía, sino el grupo completo.

Figura 14

Ejemplo de anomalía contextual



Nota. De "A note about finding anomalies," por K. Cao, 2020, Towards Data Science.

La detección de anomalías en equipos industriales es esencial para el mantenimiento predictivo, permitiendo identificar señales tempranas de fallas, optimizando la gestión de los

equipos. Sin embargo, en sistemas industriales dinámicos como motorreductores, se presentan varios desafíos ya que estos sistemas operan en entornos cambiantes y están sujetos a variaciones dentro del funcionamiento, lo cual dificulta la identificación de patrones anómalos.

En el artículo titulado *“Collaborative Contextual Anomaly Detection for Industrial Equipment Groups”* se implementa un enfoque colaborativo para la detección de anomalías contextuales, basado en métodos que agrupan los equipos con contextos similares para generar perfiles de degradación específicos. Mediante la comparación de estos perfiles perteneciente a cada grupo, se logran identificar patrones atípicos con mayor precisión. Este método resulta efectivo especialmente en entornos donde se dispone de múltiples equipos con condiciones similares, ya que permite reconocer desviaciones en parámetros clave, como vibración, temperatura, velocidad, etc. que puedan identificar fallos inminentes.

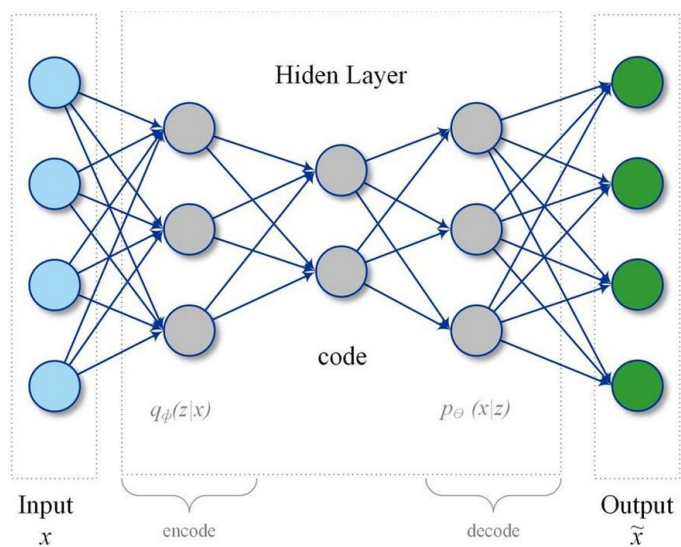
La metodología propuesta por el autor no solo distingue entre la degradación normal y severa, sino que también logra adaptarse a la variabilidad de los equipos al considerar su contexto operativo. Sin embargo, su eficacia depende de contar con suficientes datos para lograr el agrupamiento colaborativo, esto representa una limitación importante en esta clase de proyectos. Esta estrategia aporta una solución robusta al mantenimiento predictivo, optimizando la vida útil y el desempeño de los equipos al predecir fallos con una precisión mejorada y costos de operación reducidos.

Dentro del artículo titulado *“How to use machine learning for anomaly detection and condition monitoring”* se presenta un enfoque que se basa en el uso de redes neuronales auto decodificadoras (conocidos también como autoencoders) que si bien comparten principios similares con el análisis estadístico clásico, se diferencian en algunos aspectos clave, ya que se trata de una red neuronal no supervisada que aprende de representaciones eficientes de datos, comúnmente utilizadas para la reducción de la dimensionalidad. La estructura de un autodecodificador que incluye una capa de entrada, capas ocultas y una capa de salida que intenta reconstruir las entradas, generando así una codificación comprimida de los datos.

El proceso de detección de anomalías con autodecodificadores comienza por el entrenamiento de la red con datos que representan el estado operativo normal del equipo, con el objetivo de comprimir y reconstruir las variables, aquí se pueden incorporar interacciones no lineales. A medida que el equipo se degrada, estas interacciones se ven afectadas, lo que incrementa un error de reconstrucción de la red, este error por consecuencia se emplea para evaluar el estado de salud del equipo. Un aumento en el error sugiere posibles anomalías, permitiendo detectar degradaciones o desfases tempranos. Esta reconstrucción se utiliza con un determinado error cuadrático medio

Figura 15

Estructura de red neuronal autoencoder



Nota. De " How to use machine learning for anomaly detection and condition monitoring," por V. Flovik, 2018, Towards Data Science.

Capítulo 2

Desarrollo e implementación de modelo

Los algoritmos de inteligencia artificial y machine learning tienen mucha influencia sobre los avances tecnológicos y científicos, así como también tiene diferentes campos de aplicación como en educación, medicina, ingeniería, entre otros. En este caso específico, resaltar la importancia de las redes neuronales en el campo de mantenimiento predictivo en empresas agroindustriales, debido a su capacidad de lograr manejar un gran volumen de datos o también llamados “Big Data”, basándose en el comportamiento del cerebro humano.

En este capítulo se detalla el proceso completo del desarrollo del modelo de redes neuronales para la detección de anomalías en motorreductores agroindustriales el cual está basado en una red neuronal autoencoder esto por su capacidad de análisis de secuencias de datos temporales. Estas redes se utilizan ampliamente en la compresión y reconstrucción de datos.

Las redes neuronales autoencoder son modelos de aprendizaje no supervisado pensados para aprender de una representación comprimida de los datos de entrada, generando una o más salidas con el valor más cercano posible a la entrada original. Bajo el contexto de la detección de anomalías, el autoencoder está diseñado únicamente para reconstruir datos “normales”, por lo que para lograr este objetivo se tiene dentro de los datos de entrada una señal binaria que indica el estado de operación del equipo donde se tiene un valor de 1 para el encendido y 0 cuando el equipo se encuentra apagado, con esto se consigue identificar los periodos normales de operación del equipo y trabajar el entrenamiento con estos. El aprender con datos “buenos” significa que cuando se le introduce un dato anómalo, el modelo genera un error de reconstrucción mayor al esperado, lo que permite identificar patrones que no se ajustan al comportamiento estándar.

Dentro del presente capítulo se aborda el proceso completo, desde la recolección y el preprocesamiento de los datos necesarios, hasta el diseño, entrenamiento y validación del modelo. Además, se presentan los resultados obtenidos y se analizará su desempeño y resultados de este. Según Calderón Gasca, Franco Niño y Pérez Bernal (2023) el preprocesamiento de datos es importante para unificar y refinar las bases de datos, continuando con un análisis exploratorio con el objetivo de identificar cuáles son las variables más relevantes en la detección de anomalías.

2.1 Recolección y preprocesamiento de datos

La recolección de datos es un paso importante en cualquier tipo de estudio que implique el análisis de datos y detección de anomalías, especialmente dentro de los modelos de redes neuronales como los autoencoder, ya que, este modelo aprende a reconocer patrones y reconstruir los datos a partir de la información con la que ha sido entrenado.

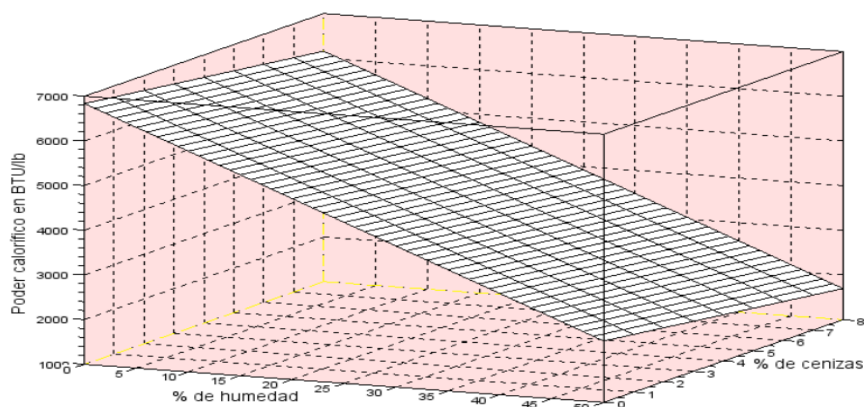
El propósito del preprocesamiento de datos según Rodríguez J. (2008) es principalmente el de corregir las inconsistencias en los datos que servirán como base para el análisis en procesos de minería de datos. En el caso de fuentes de datos estructuradas, este propósito no varía lo que hace posible la aplicación de estas amplias técnicas estadísticas y de aprendizaje automático.

Mediante el preprocesamiento de los datos se busca que los datos utilizados en tareas de análisis o descubrimiento de conocimiento mantengan su coherencia. Para la mayoría de los problemas actuales relacionados con volúmenes grandes de información, es crucial el uso de soluciones escalables y distribuidas, ya que las series temporales secuenciales no logran manejar estas magnitudes. A lo largo del tiempo han surgido diversas plataformas de procesamiento a gran escala que abordan el desafío del Big Data. Dentro de estas plataformas se requiere que los datos sean escalables para que puedan realizar las tareas más importantes en lo que refiere a la analítica de datos avanzados.

Dentro de esta etapa de recolección y preprocesamiento se han recolectado datos históricos de operación de un periodo de zafra o molienda de un motorreductor aplicado en procesos agroindustriales para el secado de bagazo de caña de azúcar, este es un proceso que se ve como alternativa para que los ingenios azucareros logren optimizar el uso de combustible para alimentar sus calderas lo que se traduce en la generación de una cantidad mayor de energía eléctrica a partir de este combustible como lo es la biomasa. El reducir la humedad en el bagazo representa un aumento en el poder calorífico del mismo, además, significa unas mejoras importantes en la eficiencia del horno de caldera durante el proceso de combustión. A continuación, en la figura 16 se muestra las fases de combustión de bagazo en relación con el porcentaje de humedad, poder calorífico y porcentaje de cenizas:

Figura 16

Pérdida del poder calorífico de bagazo en relación con su humedad y cenizas



Nota. De "Secado de bagazo, una alternativa para ahorro de combustible" por M. Muñoz, 2015, Cengicaña.

2.1.1 Características de motorreductor

Dentro de los procesos agroindustriales, los motores eléctricos son componentes esenciales cuyo desempeño influye directamente en la eficiencia y continuidad operativa de la planta. La recopilación de datos permite el registro de parámetros relevantes de operación y rendimiento mediante diversos sensores e instrumentos de medición especiales para este tipo de aplicaciones. Esto proporciona un panorama detallado del comportamiento del motor y de su estado de operación, aquí se identifican puntos importantes que son necesarios para la detección de anomalías, eligiendo aquellas variables que representen desgaste o desviaciones significativas en el rendimiento del motor. Además, el conocer las características del equipo y especificaciones como la potencia nominal, velocidad y corriente de operación, parámetros considerados críticos para entender el funcionamiento bajo las condiciones de carga de proceso de secado. Por otro lado, el reductor planetario que se tiene acoplado cuenta con características como la relación de reducción, torque máximo y las especificaciones de lubricación, fundamentales para la detección de anomalías. Con esto se logra identificar si el equipo opera dentro de los límites recomendados o por el contrario muestra señales tempranas de desgaste.

Del motor eléctrico se tienen las siguientes características:

Tabla 1

Parámetros de motorreductor

Parámetro	Valor	Unidad
Carcasa	450L	
Potencia	900	HP
Frecuencia	60	Hz
Polos	6	
Rotación nominal	1194	rpm
Corriente primaria	1012	A
Tensión	460	V
Corriente de arranque	6578	A
Corriente en vacío	285	A
Par nominal	5367	Nm

Parámetro	Valor	Unidad
Vibración	2.8	mm/s
Elevación de temperatura	80	°C

Teniendo en cuenta el contexto previo, se considera a este equipo motorreductor que conforma parte del activo de un secador de bagazo del sector agroindustrial, como un activo crítico para cualquier planta dentro del rubro, por lo mismo, se necesita que este activo se encuentre en un monitoreo constante de sus parámetros de operación para garantizar la disponibilidad y evitar paradas inesperadas en el proceso. Entre estos se encuentran mediciones variables como temperatura de los acoplamientos, vibraciones en puntos específicos tanto para el motor como para el reductor, que para estas aplicaciones se utilizan de reductores planetarios, corrientes, etc.

Para el siguiente trabajo de investigación se utilizó de la base de datos SQL de Process Historian de sistema PCS7 de Siemens para la recolección de datos históricos que archivan datos a largo plazo de los diversos datos y mensajes de los procesos industriales que se monitorean desde aquí. En la tabla 2 se muestran las variables recolectadas para una molienda de aproximadamente 4.7 meses:

Tabla 2

Señales recolectadas de motorreductor

Nombre de atributo	Descripción	Unidad
TT_MasaSupLM Value Y	Temperatura de Bronce Lado Motor Masa Superior	°C
TT_MasaCanLM Value Y	Temperatura de Bronce Lado Motor Masa Cañera	°C
TT_MasaSupLL Value Y	Temperatura de Bronce Lado Libre Masa Superior	°C
TT_MasaCanLL Value Y	Temperatura de Bronce Lado Libre Masa Cañera	°C
LT_ChuteMolinoA Value Y	Nivel de Chute de entrada de Bagazo Lado A	%
LT_ChuteMolinoB Value Y	Nivel de Chute de entrada de Bagazo Lado B	%
I_Motor UH Value Y	Corriente de Motor de Bomba Hidraulica	A

Nombre de atributo	Descripción	Unidad
PT_UH Value Y	Presión de Motor de Bomba Hidraulica	bar
ESTADO_Motor	Estado de operación de motor principal	-
TT_Motor R Value Y	Temperatura de Bobina de Fase R motor principal	°C
TT_Motor S Value Y	Temperatura de Bobina de Fase S motor principal	°C
TT_Motor T Value Y	Temperatura de Bobina de Fase T motor principal	°C
TT_Motor NDE Value Y	Temperatura de Rodamiento de motor Lado Libre	°C
TT_Motor DE Value Y	Temperatura de Rodamiento de motor Lado Acople	°C
VT_Motor DE Value Y	Vibración en Lado Acople de motor principal	mm/s
VT_Motor NDE Value Y	Vibración en Lado Libre de motor principal	mm/s
I_Motor Value Y	Corriente de Motor Principal	A
ST_Motor Value Y	Velocidad de Eje de Motor principal	RPM
VT_Reductor I Value Y	Vibración en Etapa I de Reductor	mm/s
VT_Reductor II Value Y	Vibración en Etapa II de Reductor	mm/s
TorqueReductor ValueY	Torque de Reductor	kNm

Estos valores representados en la tabla anterior serán preprocesados antes de ingresar al modelo, lo que implica realizar un análisis exploratorio de los datos como el análisis de histogramas, correlación y validación de su importancia en la detección de anomalías

2.1.2 Preprocesamiento de los datos

Como primer paso para el preprocesamiento de los datos, se realizó un análisis de los datos por variable que se tienen del dataset de entrada, observándose en la tabla 3 los siguientes resultados:

Tabla 3
Información del set de datos

Columna	Conteo No Nulos	Tipo de dato
TT_MasaSupLM ValueY	191170	float64
TT_MasaCanLM ValueY	191170	float64
TT_MasaSupLL ValueY	191170	float64
TT_MasaCanLL ValueY	191170	float64
LT_ChuteMolinoA ValueY	191170	float64
LT_ChuteMolinoB ValueY	191170	float64
I_MortorUH ValueY	191170	float64
PT_UH ValueY	191170	float64
ESTADO_Motor	29033	float64
TT_MotorR ValueY	191170	float64
TT_MotorS ValueY	191170	float64
TT_MotorT ValueY	191170	float64
TT_MotorNDE ValueY	191170	float64
TT_MotorDE ValueY	191170	float64
VT_MortorNDE ValueY	191170	float64
VT_MotorDE ValueY	191170	float64
I_Motor ValueY	191170	float64
ST_Motor ValueY	191170	float64
VT_ReductorI ValueY	191170	float64

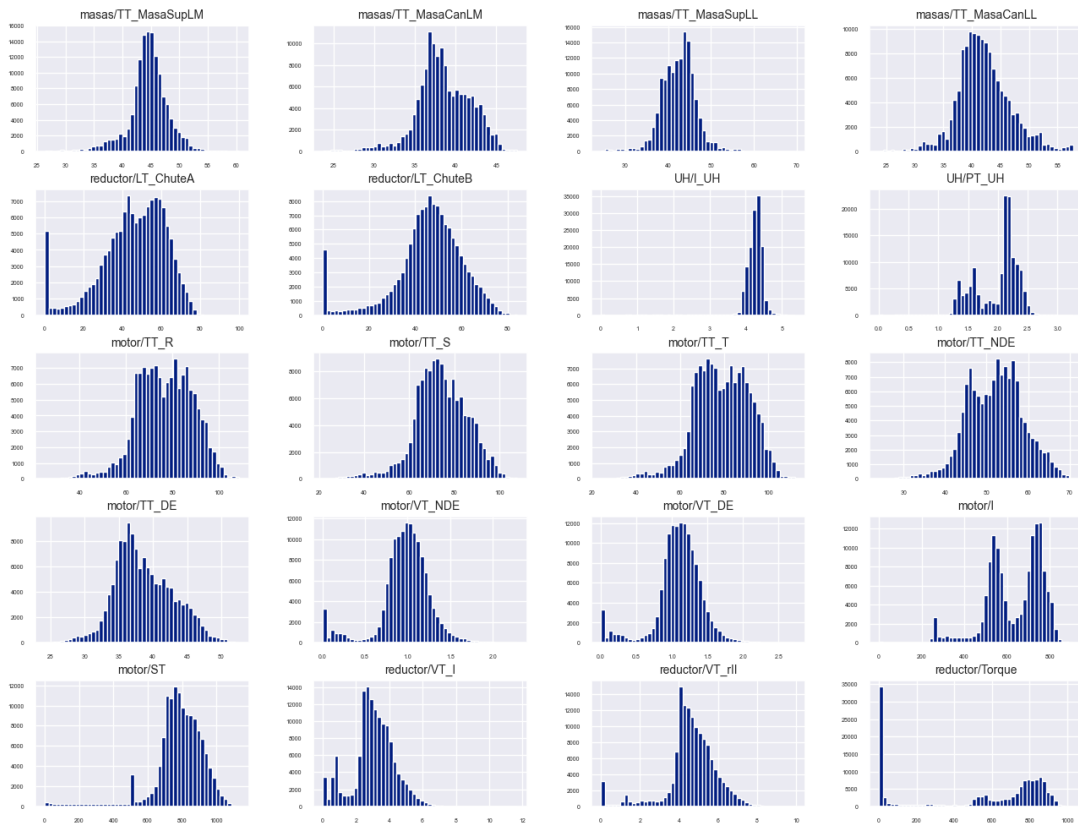
Columna	Conteo No Nulos	Tipo de dato
VT_ReductorII ValueY	191170	float64
TorqueReductor ValueY	191170	float64

Como se observa en la tabla 3, se encuentran alrededor de 191170 muestras por variable, a excepción de la variable llamada “ESTADO_Motor”, esto se debe a que el sistema antes mencionado de donde se recolectan los datos, para esta variable esta almacenando los cambios de estado en la señal de operación del motor eléctrico, por lo que para uniformizar las muestras, se aplica una estrategia de imputación de datos llamado *forward fill* (ffill) el cual es utilizado para completar los datos nulos o faltantes de una serie de tiempo utilizando el ultimo valor conocido hacia adelante, esto es útil, ya que para la señal en mención mantiene la continuidad de los datos.

Continuando con el preprocesamiento y limpieza de los datos, se procede a realizar un filtrado en las señales de entrada, esto con el fin de disminuir el ruido de las señales, permitiendo un ingreso de información limpia dentro del modelo de detección de anomalías. Para este caso se ha aplicado un filtro llamado “*Moving Average Filter*” o también conocido como filtro de media móvil el cual es óptimo para reducir el ruido aleatorio conservando una respuesta en el paso con la suficiente nitidez, este es un muy buen filtro si se quiere aplicar en señales codificadas en el dominio del tiempo.

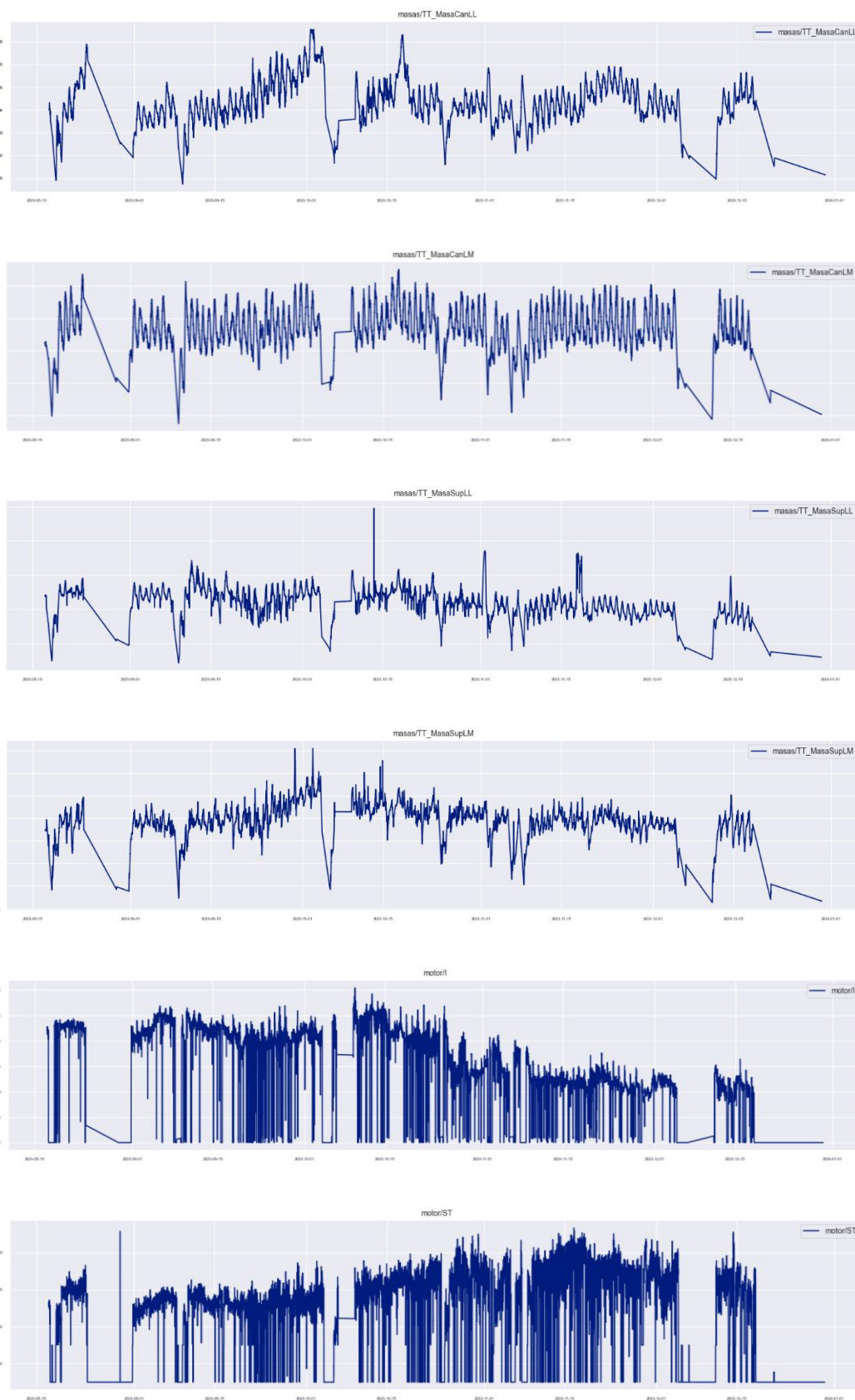
Una vez realizado este filtrado en las variables, siguiendo el principio de la red neuronal autoencoder, se necesita de una data en periodos normales, por lo que para lograr este objetivo se toma la señal de estado de motor, en donde se eliminan los periodos en los cuales el motorreductor se ha encontrado apagado o fuera de servicio por algún motivo.

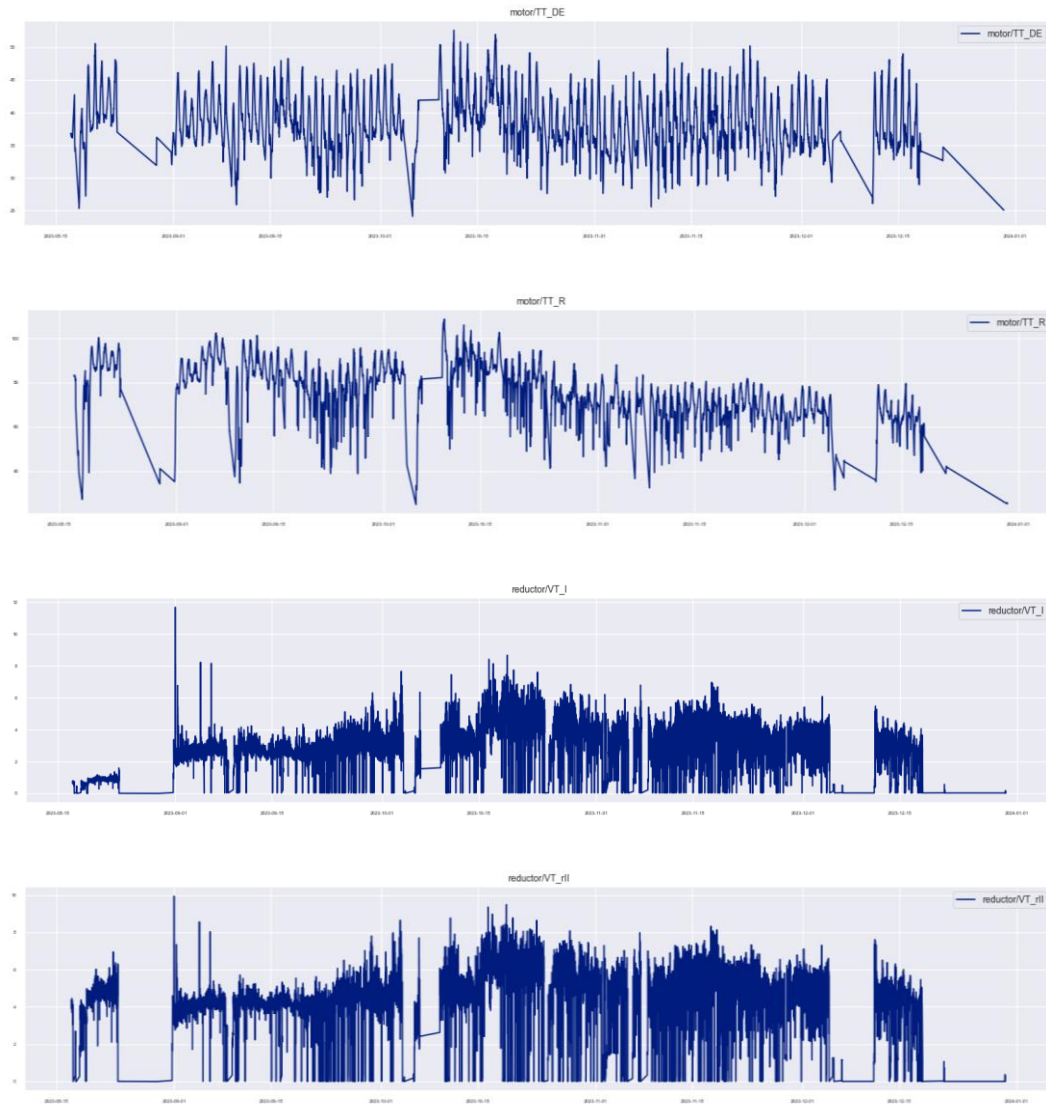
Con el fin de comprender el comportamiento de cada señal se realiza el histograma respectivo que permita visualizar la distribución de los valores para cada variable con lo cual se pueden identificar si se presentan valores atípicos que puedan influir en el modelo. En la figura 17 se observa el histograma para cada una de las variables en cuestión:

Figura 17*Histograma de sensores de motorreductor*

Como se observa en la figura 17, algunos histogramas no continúan una distribución normal, algunas señales presentan valores atípicos. Esta información es relevante ya que permite definir el tipo de normalización o escalado que se debe seguir para el modelo de anomalías.

Previamente se muestra las tendencias de los datos preprocesados siguiendo los criterios anteriormente descritos, esto es posible observar en la figura 18:

Figura 18*Tendencias preprocesadas de señales*



De estos gráficos de tendencias se puede observar que para cada una de las señales se ha eliminado los periodos en donde el motorreductor ha estado apagado o en estado “0”, esto ayuda a que el modelo pueda aprender con esta data en los periodos de operación. Aquí se siguen las siguientes técnicas de análisis:

a) Análisis de correlaciones

Esta técnica de análisis es usualmente utilizada en donde se requiere de la evaluación estadística de la asociación entre dos variables numéricas. Esto se considera así cuando existen indicios de sociedad entre dos variables, es decir, que una proporciona información sobre la otra. De ser así ambas se pueden alterar al mismo tiempo durante un periodo determinado.

Dentro de este análisis existe el análisis de correlación positiva, el cual se presenta cuando el aumento de cualquiera de las dos variables ocasiona un efecto de

la mayor cantidad de datos, mientras que el segundo contiene un conjunto de prueba más pequeño. Para este modelo se selecciona la estrategia de 80-20, en donde el 80% de los datos pasan a ser entrenados en el modelo, mientras que el 20% restante se prepara para una posterior validación del modelo. La división se realiza de manera aleatoria, con el fin de garantizar la variabilidad en ambos conjuntos, tal cual se detalla en la tabla 4:

Tabla 4
Información de la división de Train y Test

Conjunto	Cantidad de Datos	Porcentaje
Train	123667	90%
Test	13741	10%
Total	137408	100%

Con este preprocesamiento previo ya es posible continuar con el diseño de la red neuronal autoencoder, tal cual se detalla en el siguiente apartado.

2.2 Diseño de red neuronal autoencoder

Para el desarrollo del modelo de red neuronal autoencoder previamente se ha determinado la distribución para los datos de entrenamiento y validación. En este modelo la arquitectura está conformada por una arquitectura de tipo encoder – decoder, en donde el encoder se encarga de comprimir los datos de entrada a una representación en menor dimensión, conocido también como capa latente, a continuación, se realiza la reconstrucción en el decoder, intentando que la salida sea lo más similar posible al dato de entrada.

Para la construcción del modelo de detección de anomalías, como ya se había mencionado anteriormente para el ingreso de las variables al modelo de red neuronal se necesita hacer un proceso de escalamiento y normalización de los datos que es un paso fundamental para procesar la información de las señales ya que estas suelen ser multidimensionales como se observa en la tabla 2. El proceso garantiza que cada una de las variables tengan igual importancia.

2.2.1 Escalado y normalización de datos

El proceso de escalado y normalización según Rodríguez J. (2021) consta de herramientas que mejoran el rendimiento de la predicción del algoritmo utilizado con el fin de reducir su costo computacional. Dentro de esto se encuentra a la clasificación, predicción, reconocimiento de patrones y señales como las tareas más comunes en donde se aplica este tipo de estrategias, la cual describe las características mediante una ecuación específica. Un

enfoque comúnmente empleado en la normalización es la estandarización la cual indica que cada una de las características deberán tener una media correspondiente a 0 y una desviación estándar de 1. La normalización también es conocida como escala de varianza de unidad, esta se puede calcular mediante la siguiente formula:

$$\bar{X}_{ij} = \frac{X_{ij} - \bar{X}_i}{S_i}$$

Donde X_{ij} representa al valor original del elemento en la posición i, j ; \bar{X}_i es el valor promedio de cada variable i , es decir, aplicado para el caso de motorreductor es el promedio de todos los valores de esa característica en el conjunto de datos y S_i representa a la desviación estándar de cada variable i .

En el presente caso para el proceso de normalización y escalado se ha empleado el escalador `sklearn.preprocessing.MinMaxScaler` el cual transforma las características dentro de un rango dado que por defecto es 0 y 1. Se debe de tener en cuenta que para este tipo de escalado los valores nulos o faltantes no se tienen en cuenta al momento del ajuste, por lo cual se mantienen en la transformación. Este tipo de escalado viene dado por la siguiente ecuación:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Donde X es el valor original de la variable, X_{min} es el valor mínimo para cada variable, X_{max} representa al valor máximo y X' es el valor escalado por el método. Lo resaltante en este método es que a pesar de que el rango de datos ha cambiado, la distribución de estos se mantiene, lo cual lo vuelve en una buena opción cuando se requiere trabajar con datos que no siguen a una distribución normal.

Figura 20

Código para normalización y escalamiento de los datos

```
# Normalización de la data
scaler = preprocessing.MinMaxScaler()
X_train = scaler.fit_transform(df_train)
X_test = scaler.fit_transform(df_test)
```

Como se observa en la figura 20, se realiza la normalización de la data tanto para la que será empleada en el entrenamiento como en la validación. Posterior a este proceso, se realiza un redimensionamiento de los datos, esto porque en el contexto de las redes neuronales recurrentes, especialmente en las de tipo LSTM (Long Short-Term Memory) los datos de entrada requieren estar en un formato específico: [muestras, pasos de tiempo, características], mientras que la información que se brinda en la entrada tiene el siguiente dimensionamiento: [muestras, características]. El proceso de redimensionamiento se realiza tal cual se muestra en la figura 21:

Figura 21

Código para redimensionamiento de los datos

```
#Reshape inputs for LSTM
X_train = X_train.reshape(X_train.shape[0],1,X_train.shape[1])
print('Training data shape:',X_train.shape)
X_test = X_test.reshape(X_test.shape[0],1,X_test.shape[1])
print('Test data shape:',X_test.shape)
```

Se plantea una red neuronal autoencoder de 4 capas ocultas de dimensiones 12,4,4 y 12 basado en LSTM que está diseñada para trabajar con datos secuenciales de series de tiempo, en donde la entrada se recibe con una forma de (n_timesteps, n_features) que son los números de pasos de secuencia y el número de características por paso de tiempo respectivamente.

La parte del codificador o Encoder está conformada por dos capas. Una primera capa que contiene 12 neuronas, una función de activación de tipo “ReLU” con una regularización de tipo L2 con el fin de evitar un sobreajuste. Se utiliza dentro de esta capa una secuencia de retorno verdadera, ya que en capas LSTM de un autoencoder garantiza que cada paso de tiempo en la secuencia original se conserve a la salida de la capa. Esto es importante ya que en cada paso de la secuencia de tiempo se aprende de las relaciones temporales entre los pasos, además que alimenta a las siguientes capas LSTM para que trabajen bajo esta misma secuencia. La segunda capa correspondiente al Encoder trabaja con 4 neuronas bajo la misma función de activación “ReLU”, a diferencia de la primera capa, aquí no se requiere de una secuencia de retorno ya que se desea condensar la información a un vector de tamaño 4. Esto comienza a crear una versión comprimida de los datos de entrada o también llamada representación codificada.

Luego de esta capa, continua la capa de reconstrucción que es donde se comienza a expandir la salida comprimida del Encoder a lo largo del eje temporal, replicando en cada paso de tiempo el vector codificado en la secuencia original. Se toma un vector de tamaño 4 y se replica según el total de número de muestras en la entrada, preparando la fase de decodificación.

Culminando la etapa de condensación o también llamado “cuello de botella”, empieza la etapa de decodificación en donde se utiliza de una tercera capa de 4 neuronas, de igual forma con la función de activación ya antes mencionada. Aquí se requiere nuevamente de una secuencia de retorno que devuelva la completa secuencia de los datos, dando comienzo a la etapa de reconstrucción de los datos a partir de una representación comprimida. En la cuarta capa del Decoder se emplea una capa LSTM de 12 neuronas devolviendo la secuencia completa. Se va afinando la reconstrucción.

Finalmente se tiene una capa de salida que tendrá tantas unidades como características se encontraban en la entrada original. Con esto se define la creación del modelo de red neuronal autoencoder que está en la capacidad de reconstruir datos normales con baja pérdida en el error.

2.3 Entrenamiento y validación del modelo

En el presente apartado se detalla la etapa de entrenamiento y validación del modelo de red neuronal propuesto en el apartado anterior. Esto es importante para posteriormente evaluar y mejorar el rendimiento en la detección de anomalías. El entrenamiento del modelo consiste en ajustar los parámetros internos según el diseño de la red para que se pueda aprender de valores representativos de los datos de entrada. A través de la etapa de validación se verifica la capacidad del modelo de identificar patrones ocultos, evaluando su precisión su capacidad de detectar comportamientos de falla.

Con el fin de lograr que el modelo sea robusto, evitando los sobreajustes, se utilizaron previamente técnicas de preprocesamiento, estrategias de división de conjunto de datos en divisiones de entrenamiento y validación.

Dentro de la ejecución del modelo se considera un número de 1000 épocas para el entrenamiento, colocando la función de Early Stopping la cual se encarga de detener el modelo si es que no se llegase a tener una mejora en la pérdida una vez pasada 200 épocas, tal cual se muestra en la figura 22:

Figura 22

Evaluación de función de pérdida en entrenamiento y validación

```

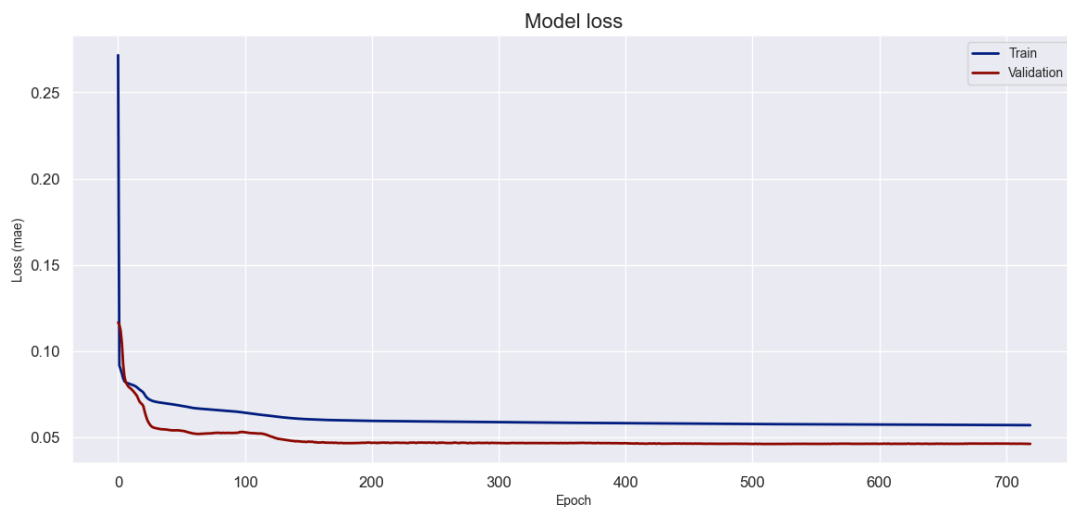
Epoch 1/1000
198/198 ----- 6s 5ms/step - loss: 0.4499 - val_loss: 0.0988
Epoch 2/1000
198/198 ----- 1s 3ms/step - loss: 0.0889 - val_loss: 0.1112
Epoch 3/1000
198/198 ----- 1s 3ms/step - loss: 0.0834 - val_loss: 0.1088
Epoch 4/1000
198/198 ----- 1s 3ms/step - loss: 0.0752 - val_loss: 0.0724
Epoch 5/1000
198/198 ----- 1s 3ms/step - loss: 0.0550 - val_loss: 0.0709
Epoch 6/1000
198/198 ----- 1s 3ms/step - loss: 0.0527 - val_loss: 0.0701
Epoch 7/1000
198/198 ----- 1s 3ms/step - loss: 0.0516 - val_loss: 0.0693
Epoch 8/1000
198/198 ----- 1s 3ms/step - loss: 0.0507 - val_loss: 0.0688
Epoch 9/1000
198/198 ----- 1s 3ms/step - loss: 0.0500 - val_loss: 0.0689
Epoch 10/1000
198/198 ----- 1s 3ms/step - loss: 0.0496 - val_loss: 0.0687
Epoch 11/1000
198/198 ----- 1s 3ms/step - loss: 0.0493 - val_loss: 0.0688
Epoch 12/1000
198/198 ----- 1s 3ms/step - loss: 0.0490 - val_loss: 0.0687
Epoch 13/1000
...
Epoch 683/1000
198/198 ----- 1s 3ms/step - loss: 0.0398 - val_loss: 0.0604
Epoch 684/1000
198/198 ----- 1s 3ms/step - loss: 0.0398 - val_loss: 0.0606

```

Una vez realizado este entrenamiento es posible visualizar la tendencia en la función de pérdida del modelo tanto para los datos de entrenamiento como los datos de validación, los cuales se presentan en la figura 23:

Figura 23

Grafico de función de pérdida de modelo



En la figura 23, se observa la convergencia del modelo de red neuronal, el cual encuentra su punto de mejora máximo después de 684 épocas. Además, se observa que pese

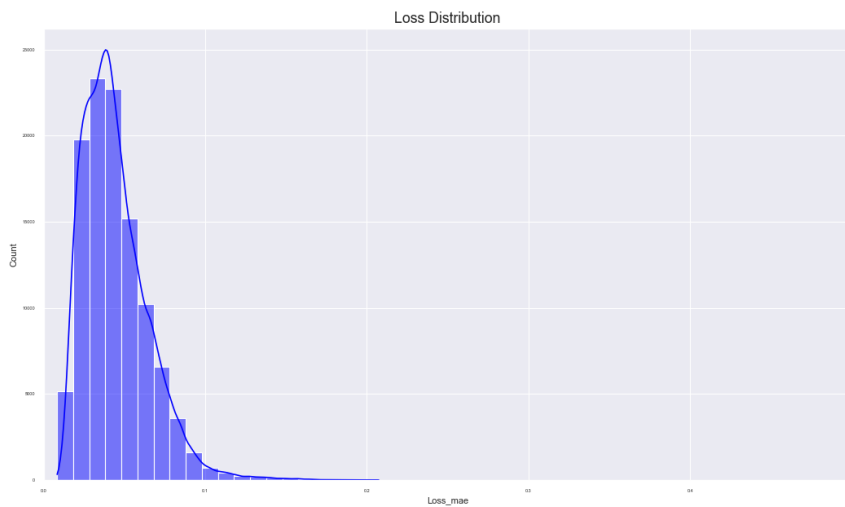
a que existe diferencia entre la data de entrenamiento y la data de validación, ambas siguen la misma tendencia, lo que significa que el modelo tiene un buen ajuste y que está aprendiendo de manera efectiva a reconstruir las entradas tanto en el conjunto de entrenamiento como el de validación, esto es un indicador que el modelo no se encuentra sobre ajustado (overfitting) ni subajustado (underfitting).

El objetivo de realizar la evaluación de la función de pérdida en el entrenamiento es identificar el valor de pérdida para cada dato que registre el Autoencoder ya entrenado, con esto se define un umbral de anomalías para decidir si el registro dado se trata de un valor anómalo o normal. Este factor de anomalías o técnicamente llamado como valor de pérdida se calcula teniendo en cuenta el error absoluto medio (MAE) el cual encuentra la diferencia entre el valor que ha sido predicho por el modelo, es decir, la salida que entrega la red neuronal y el valor de entrada al Autoencoder.

La figura 24 muestra la distribución de todos los valores de función de pérdida que han sido calculados para el entrenamiento:

Figura 24

Histograma de la distribución de función de pérdida



Como se observa en la figura, el error sigue una distribución normal, lo que es una buena señal para modelo de detección ya que es un indicativo de que la reconstrucción de los datos es constante y que además en la mayoría de las iteraciones se tiene una pérdida cercana al promedio y menos instancias con pérdidas muy altas o bajas. Esto quiere decir además que los datos de entrada son representativos al proceso. Para la determinación del umbral de anomalías, es decir, establecer una desviación significativa del sistema se debe de fijar un umbral partiendo de la distribución del error de reconstrucción entre el dato de salida y el dato de entrada, empleando técnicas estadísticas como el cálculo de la media y de la desviación estándar del error durante el periodo de entrenamiento, como ejemplo se puede utilizar la suma de la media más tres veces la desviación estándar, lo cual indica que cualquier

error que supere este valor, entra en una alta probabilidad de anomalía del sistema. Otra alternativa empleada es definir el umbral con el percentil 95 o 99 del error, esto es ideal cuando la distribución de este no sigue a una distribución normal, con situaciones con datos asimétricos. Este método es ideal cuando se requiere de un control más fino sobre la probabilidad de falsos positivos o negativos. También existe la posibilidad de definir un umbral dinámico que ajustan su umbral en función a factores externos como la carga del sistema, condiciones ambientales, hora del día, etc.; esto genera una detección de anomalías mucho más sensible y que se adapta a las variaciones normales del sistema.

Para el presente trabajo de investigación, se ha considerado la definición del umbral siguiendo métodos estadísticos y analizando la distribución de la función del error como se muestra en la figura 24. Por lo que la fórmula de definición del umbral queda de la siguiente manera:

$$threshold = \mu_{p\acute{e}rdida} + 6 * \sigma_{p\acute{e}rdida}$$

Donde:

- $\mu_{p\acute{e}rdida}$: es la media de función de pérdida (Mean Loss)
- $\sigma_{p\acute{e}rdida}$: desviación estándar de la pérdida

El umbral de anomalías se encuentra sumando seis desviaciones estándar a la media de pérdida. Con esto se define el umbral indicando que para cualquier valor que supere el establecido por el cálculo se identificará como una anomalía dentro del motorreductor de molino. Con esto se genera una nueva señal llamada "Anomaly" que indica 0 para una operación normal y 1 cuando se registra una condición de anomalía

2.4 Resultados y análisis

En esta etapa del trabajo de investigación se presentan los principales hallazgos obtenidos tras aplicar el modelo de detección de anomalías en motorreductor que es utilizado en el proceso de secado de bagazo. Este análisis incluye la identificación de los patrones y comportamientos anómalos, así como la interpretación de estos en el contexto del funcionamiento y rendimiento del equipo. Mediante gráficos y tablas se presentan las tendencias y comportamiento de las variables analizadas, permitiendo evaluar la eficacia del modelo. Estos resultados aportan un gran valor en optimizar el mantenimiento predictivo, asegurando una operación más confiable y eficiente.

Teniendo en cuenta que en el apartado anterior se definió el cálculo para el umbral de anomalías, se obtuvo que dicho umbral fue de 0.145, por lo que cualquier valor de error

superior al mencionado, es catalogado como anomalía. En la figura 25 se muestra el código de cómo se va a almacenar la nueva señal que indica anomalía:

Figura 25

Creación de señal binaria de anomalía

```

scored_train['Threshold'] = threshold
scored_train['Anomaly'] = scored_train['Loss_mae'] > scored_train['Threshold']
# Convertir scored Anomaly en booleano
scored_train['Anomaly'] = scored_train['Anomaly'].astype(int)
✓ 0.0s

```

A partir de la creación de esta señal mostrada en la figura 25, es posible encontrar la cantidad de anomalías encontradas por el modelo y sobre todo en que periodos y bajo que condiciones fueron encontradas. La tabla 5 describe la cantidad de anomalías encontradas por el modelo:

Tabla 5

Identificación de anomalías del modelo autoencoder

	Datos	Porcentaje
Anómalos	325	0.24%
Normales	137102	99.76%
Total	137427	100.00%

La tabla 5 muestra la distribución de los datos clasificados por el modelo de detección de anomalías. De un total de 137427 muestras analizadas, el 0.24% (325 muestras) fueron identificadas como anómalas, mientras que el 99.76% (137102 muestras) correspondieron a datos en condiciones normales o que se encuentran dentro del rango de operación normal. Esta baja proporción de datos anómalos indica que el modelo es selectivo y eficiente al momento de identificar desviaciones significativas, minimizando la probabilidad de falsos positivos. La mayoría de los datos se consideran normales, lo cual demuestra una estabilidad operativa del motorreductor durante el periodo de observación, aunque las muestras anómalas identificadas requieren de una revisión para comprender su causa y relevancia en la salud del equipo.

Cabe resaltar que se realizó este paso como una primera prueba en la cual se excluyó a la variable de torque del reductor. Esta decisión se tomó con el fin de identificar el impacto de esta variable en el desempeño del modelo y en la capacidad de detección de anomalías.

Para la reconstrucción del error y entrenamiento del modelo se dividieron los datos en dos subconjuntos:

- 80% de la totalidad de los datos, los cuales corresponden a escenarios de operación normal del equipo.
- 20% de los datos, fueron utilizados en esta primera prueba para la evaluación del modelo en la detección de anomalías

Posterior a este proceso de identificación de anomalías, es importante también saber cuál ha sido la causa de este error alto en la reconstrucción, por lo que, conociendo los resultados se puede evaluar cada una de las señales para así tener en cuenta cual ha sido la que ha tenido mayor influencia sobre el modelo.

Para observar las variables que influyen dentro del modelo a manera general se ha utilizado de herramientas que permitan conocer esta influencia ya que muchas veces estos modelos de detección de anomalías resultan muy potentes, pero con cierta complejidad al momento de interpretar los resultados, sin embargo, los valores SHAP (Shapley Additive exPlanations) son utilizados para la interpretación de estos modelos, especialmente complejos o de tipo “caja negra” como lo son las redes neuronales. SHAP tiene como propósito explicar la predicción de una instancia específica calculando la contribución individual de cada característica de esa predicción.

Haciendo esta mención previa, se ha calculado para la primera prueba estos valores SHAP para observar cómo se refleja la influencia de cada variable ingresada al modelo. Tal cual se observa en la figura 26:

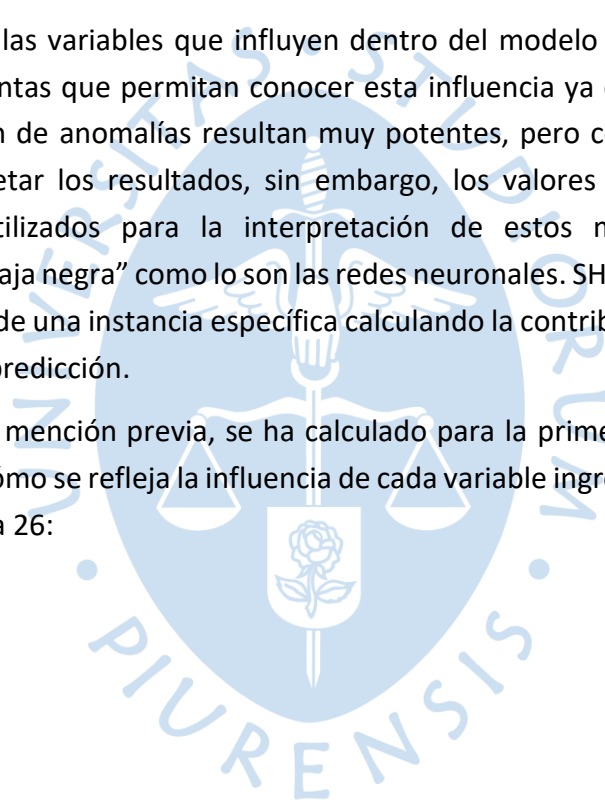
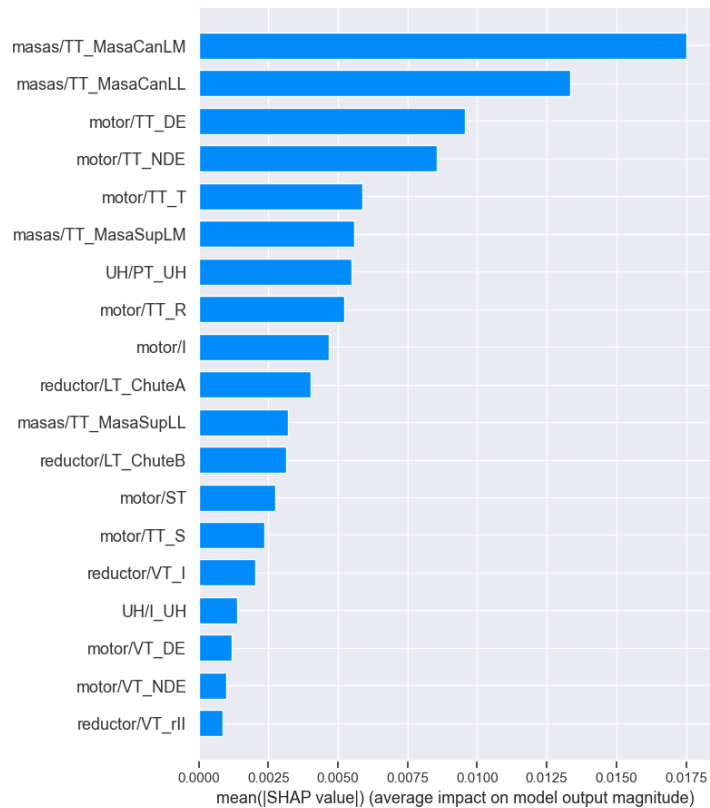


Figura 26*Importancias SHAP sobre modelo autoencoder*

Como se observa en la figura 26 las características más influyentes son “masas/TT_MasaCanLM” y “masas/TT_MasaCanLL” que corresponden a la temperatura de bronce de la Masa Cañera Lado Motor y Lado Libre respectivamente, estas tienen los valores medios de SHAP más altos, lo que quiere decir que son las características con mayor impacto sobre el modelo. Esto indica que estas variables se encuentran fuertemente correlacionadas.

Las primera cinco características son las más relevantes con una influencia acumulada considerable, por lo que se puede decir que lo más probable es que aporte información más valiosa al modelo.

Analizada esta primera prueba, se realizó una segunda considerando ahora el torque del reductor, pero incrementando el set de entrenamiento y reduciendo el de validación, quedando como estrategia 90-10. Esto con el fin de analizar el comportamiento del modelo, obteniéndose los siguientes resultados:

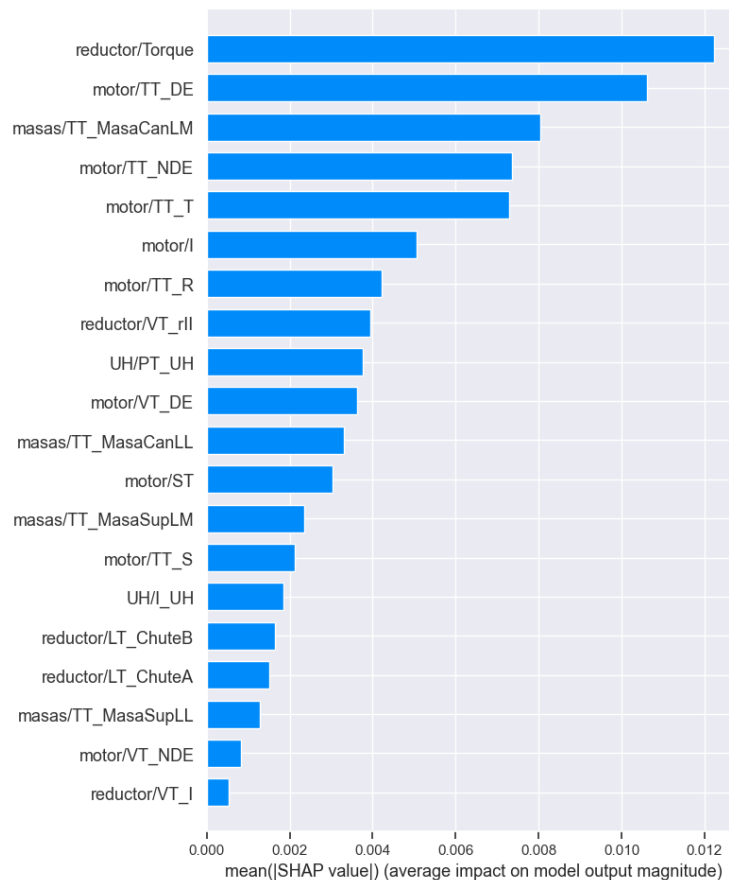
Tabla 6
Identificación de anomalías del modelo autoencoder – Prueba 2

	Datos	Porcentaje
Anómalos	2373	1.73%
Normales	135035	98.27%
Totales	137408	100.00%

La tabla 6 muestra los resultados obtenidos de la segunda prueba de detección de anomalías, incluyendo el torque de reductor y con un conjunto de datos de entrenamiento del 90% y 10% reservado para la validación. El modelo identificó 2373 instancias como anomalías, las cuales representan al 1.73% del total de datos. Este aumento en el porcentaje de anomalías encontradas puede estar relacionado con la inclusión de la última variable en mención, esto genera una sensibilidad sobre el modelo.

En la mayoría de las instancias se han encontrado datos normales, en total 135035 muestras fueron clasificadas como normales o no anomalías, correspondiente al 98.27% de los datos.

Para este caso también se han calculado los valores SHAP para evaluar las importancias de cada una de las variables sobre el modelo, obteniendo la siguiente grafica de la figura 27:

Figura 27*Importancias SHAP sobre modelo autoencoder – Prueba 2*

Como se muestra en la figura se tiene la importancia de cada una de las variables sobre el modelo, calculada mediante valores de SHAP. Para este caso el torque del reductor aparece en primera posición de esta gráfica, lo que quiere decir que, tiene una mayor influencia sobre las decisiones del modelo, esto seguido de la variable “motor/TT_DE” que hace referencia a la temperatura de motor lado libre. Esto refleja claramente que la incorporación del torque en esta prueba ha tenido un alto impacto, ya que proporciona información crítica del comportamiento del equipo para la identificación de condiciones anómalas.

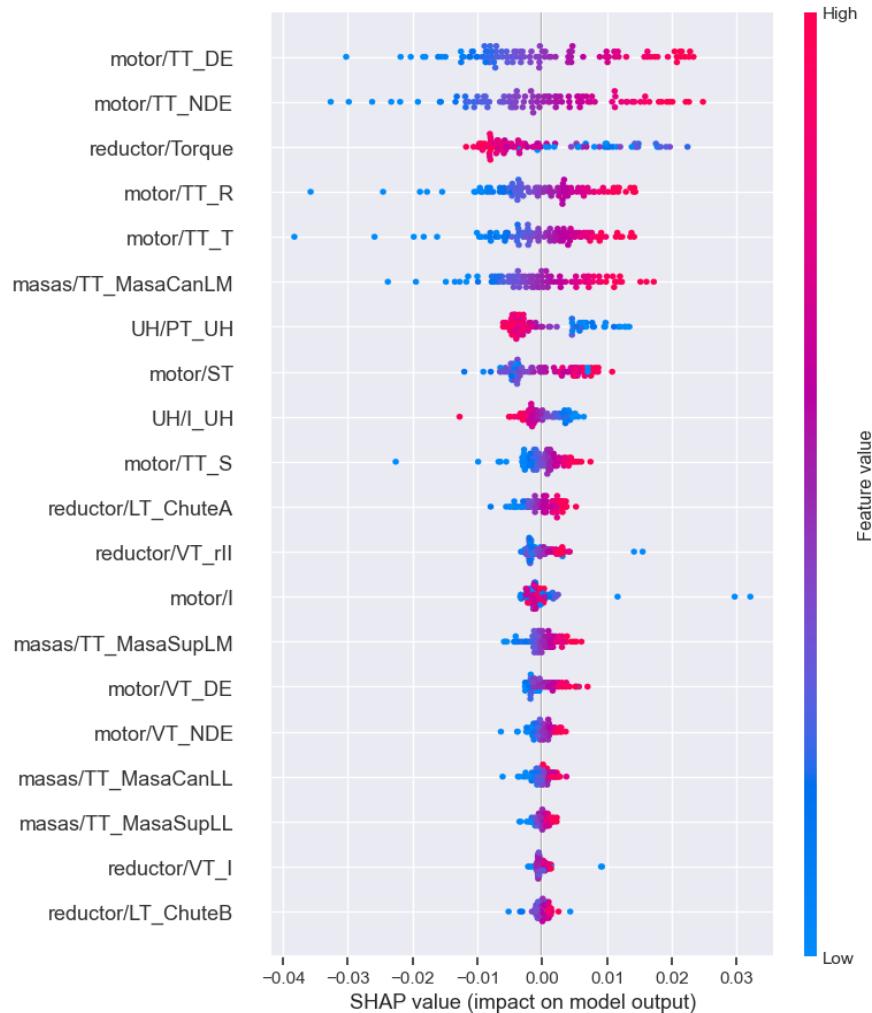
Además, resaltar de la importancia que tienen las temperaturas de masas de caña en el reductor y temperaturas de acoplamientos en el reductor. Esto sugiere que estas variables capturan las dinámicas operativas más significativas que el modelo utiliza para encontrar comportamientos inusuales.

La distribución de importancias coincide en que los parámetros del motorreductor son fundamentales para la detección de anomalías, lo cual se alinea con el objetivo de monitorear el rendimiento de un motorreductor de secado de bagazo de caña de azúcar en el sector agroindustrial.

A continuación, en la figura 28, se muestra otro panorama respecto a las importancias de las variables colocadas en el modelo, calculado mediante valores SHAP, que interpretan la contribución de cada característica en el modelo:

Figura 28

Gráfico resumen de SHAP – Modelo autoencoder



A diferencia del gráfico mostrado en la figura 27, en esta se presentan adicionalmente la distribución de cada una de las variables y la relación con las características. Dentro del eje horizontal se observa el impacto que cada variable tiene sobre la predicción del modelo. Los valores positivos indican que la variable contribuye a una mayor probabilidad de anomalía, mientras que los valores negativos reducen esta posibilidad. En el eje vertical las variables están ordenadas según su influencia global, desde la más importante ('motor/TT_DE') hasta la menos relevante ('reductor/LT_ChuteB').

Dentro de las principales observaciones a analizar por la figura 28, se tiene que las temperaturas de rodamientos del motor lado libre y acople (motor/TT_NDE y motor/TT_DE respectivamente) y el torque del reductor (reductor/Torque) son las variables que presentan

la mayor dispersión de valores SHAP, esto quiere decir que, las condiciones de temperatura en los rodamientos y el torque del reductor son factores clave para la identificación de anomalías, ya que para valores altos de temperatura, estos tienden a contribuir positivamente a las anomalías, mientras que los valores bajos disminuyen la posibilidad de fallo; de manera similar ocurre para el torque en donde según el gráfico presenta que valores bajos de este parámetro influyen positivamente a esta detección. Variables como 'reductor/LT_ChuteB' y 'reductor/VT_I' tienen un menor impacto en las predicciones del modelo, lo que podría reflejar una mejor relación directa en el comportamiento del sistema.

El análisis realizado con los valores SHAP permite identificar las variables clave que el modelo considera para la detección de anomalías, esto es importante ya que ayuda a validar el modelo y es un indicador clave para priorizar las variables que se deben tener más de cerca en el monitoreo.

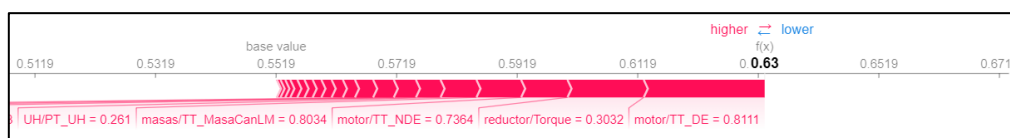
En estos casos anteriores, se tiene una visión global del comportamiento del modelo de red neuronal autoencoder, identificando las variables que afectan más en el comportamiento del modelo. A continuación, se presentan casos específicos en donde el modelo ha sido capaz de identificar anomalías.

Para evaluar el comportamiento individual de cada anomalía detectada por el modelo, se emplearon gráficos tipo SHAP Waterfall, los cuales permiten descomponer la predicción del modelo en las contribuciones individuales de las características más relevantes. Cada gráfica muestra como el valor base, que presenta la salida promedio del modelo, se ajusta mediante las características hacia la predicción final. A continuación, se muestran los siguientes casos ilustrativos:

Caso 1: Anomalía positiva

Figura 29

Gráfico SHAP Waterfall – Anomalía positiva



Para el caso mostrado en la Figura 29, se tiene una anomalía positiva en donde la predicción final del modelo es de 0.63, lo que supera al valor base de 0.5519. Esto se debe a las contribuciones positivas de características como:

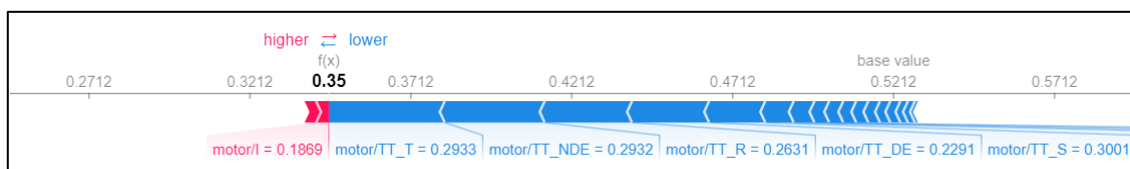
Tabla 7*Importancia SHAP de predicción positiva*

Características	Contribución [%]
masas/TT_MasaSupLM	80.34%
motor/TT_NDE	73.64%
reductor/Torque	30.32%
motor/TT_DE	81.11%
UH/PT_UH	26.1%

Las características representadas en color rosado indican que sus valores altos aumentaron a la salida del modelo, lo que sugiere a una detección de comportamiento anómalo positivo. Esta anomalía se asocia con eventos en los que las variables operativas superan los límites esperados que están potencialmente relacionados con la sobrecarga, el sobrecalentamiento de los cojinetes de apoyo del motor o condiciones fuera de lo normal por parte del sistema. Las temperaturas en los rodamientos del motor, según los valores SHAP, muestran elevación lo que puede ser un indicativo de una fricción excesiva o fallos en la lubricación, además, combinándolo con el efecto de la variable 'reductor/Torque' de 0.834 sugiere que el equipo podría experimentar una mayor carga trabajo.

La suma de todos estos efectos indica un evento anómalo positivo. Este tipo de anomalía podría estar relacionada con una sobrecarga en el sistema. Estos valores de importancia son un buen indicativo para que el equipo de mantenimiento a cargo de estos equipos tome las medidas preventivas de monitoreo de la condición para evitar un desgaste prematuro de los componentes críticos.

Caso 2: Anomalía negativa

Figura 30*Gráfico SHAP Waterfall – Anomalía negativa*

En contraste al caso anterior, la figura 30 evidencia una anomalía negativa con una predicción final de 0.35, valor por debajo del valor base que es de 0.5212. Este comportamiento refleja un caso normal, caracterizado por una disminución significativa de la

actividad operativa del equipo, posiblemente asociada con casos de subutilización o inicio de una condición de fallo.

Las características con mayor impacto negativo son:

Tabla 8
Importancia SHAP de predicción anomalía negativa

Características	Contribución [%]
motor/TT_T	29.33%
motor/TT_NDE	29.32%
motor/TT_R	26.31%
reductor/LT_ChuteA	18.54%
motor/ST	10.94%

Según se muestra en la tabla 8, las variables de temperaturas de rodamientos del motor son bajas, lo que puede ser un indicativo de una menor generación de calor debido a una disminución de carga de bagazo en el molino secador. Además, se observa un incremento en las temperaturas de las fases T y R del bobinado del motor eléctrico, esto combinado con el efecto de baja velocidad de este puede traducirse como un menor suministro de corriente al motor observado como un estado de baja carga inclusive con una inactividad momentánea del equipo.

Las contribuciones mostradas en la figura 30, las que se encuentran representadas en color azul, reflejan una condición de baja carga o un estado de reposo parcial del sistema. Este comportamiento podría deberse a fallos incipientes, interrupciones en el proceso o una operación de menor capacidad. Sin embargo, es importante el monitoreo de este comportamiento, ya que de tratarse de un funcionamiento prolongado bajo estas circunstancias puede ocasionar problemas secundarios como la acumulación de bagazo en la parte de las masas que se encuentran acopladas al reductor o condiciones inadecuadas para el equipo.

El análisis de ambos casos manifiesta la capacidad del modelo en detectar desviaciones operativas tanto hacia valores elevados (anomalías positivas) como hacia valores reducidos (anomalías negativas). Con los gráficos mostrados para ambos casos no solo identifica anomalías, sino que también brindan un desglose claro del impacto de cada característica en la predicción final.

2.4.1 Métricas evaluadas

Queda demostrada la importancia de las redes neuronales en la industria, sin embargo, también se debe poder cuantificar el rendimiento de estas para poder obtener resultados y tomar mejores decisiones al momento de realizar mejoras o alguna implementación en la industria como en fábricas industriales en la detección de anomalías.

En esta etapa se presentan y explican las métricas utilizadas en la evaluación del rendimiento del modelo de red neuronal autoencoder para la detección de anomalías sin etiquetas. Cabe resaltar que, debido a la naturaleza no supervisada del modelo, las métricas están enfocadas al análisis de los errores de reconstrucción generados por el modelo y en como estos pueden señalar posibles anomalías en el proceso. A continuación, se detallan las métricas seleccionadas y su interpretación:

- A) Error de reconstrucción promedio:** El error cuadrático promedio de reconstrucción es una métrica que evalúa las diferencias entre los datos originales y los reconstruidos por el autoencoder, reflejando la capacidad del modelo para aproximar los patrones de datos normales.
- B) Desviación estándar del error de reconstrucción:** Esta es una métrica comúnmente utilizada en estadística, ideal para evaluar la dispersión de los errores alrededor de la media. Para este caso en particular, esta métrica ayuda a identificar variaciones significativas en la reconstrucción, ya que, el tener un valor alto puede ser un indicador de una desviación considerable de las muestras del comportamiento esperado.
- C) Percentiles del error de reconstrucción:** Se evalúan estos datos con el fin de evaluar la sensibilidad del modelo ante posibles anomalías. Los resultados proporcionan una línea base para la observación del modelo en cómo responde a patrones no antes vistos y así ajustar el umbral de anomalías de una manera adecuada.

A continuación, se muestran los resultados obtenidos para las métricas evaluadas por el modelo autoencoder:

Para el error promedio de reconstrucción se realizó el siguiente cálculo:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Teniendo en cuenta la fórmula anterior se tiene el siguiente resultado:

Figura 31*Cálculo de error cuadrático medio*

```
# Error de reconstrucción promedio
mae_error = np.mean((X_test_pred-X_test_reshaped)**2,axis=1)
mae_avg = np.mean(mae_error)
mae_avg
```

✓ 0.0s

0.022575671445965804

Como se observa en la figura 28 se tiene para el modelo autoencoder un error calculado de 0.0225. Este valor representa la magnitud media de la diferencia entre los datos originales y los reconstruidos. Este resultado indica que el autoencoder ha logrado capturar y modelar correctamente los patrones de datos normales, además que reconstruye las características con una desviación mínima, señal que el autoencoder ha aprendido correctamente.

Para el cálculo de la desviación estándar se tiene lo siguiente:

Figura 32*Cálculo de la desviación estándar*

```
mse_std = np.std(mse_error)
```

```
mse_std
```

✓ 0.0s

0.00744567051458405

Según la figura 29, para el cálculo de la desviación estándar se tiene un resultado de 0.0074, lo que refleja la variabilidad de los errores de reconstrucción en las muestras de datos normales, su baja variabilidad indica que el modelo es bastante consistente al reconstruir datos, lo que se busca para este modelo de detección de anomalías, ya que cualquier desviación significativa de este patrón se puede interpretar como una anomalía. Además, con esto se tiene un mejor manejo sobre los falsos positivos, ya que se minimiza la posibilidad de que los datos normales sean incorrectamente etiquetados como anomalías, se concluye que el modelo es confiable, lo que es bueno para fijar umbrales adecuados.

Los percentiles del error de reconstrucción para el modelo de red neuronal autoencoder son los siguientes:

Tabla 9
*Percentiles del error medido
 para modelo*

Percentil	Resultado
90	0.0299
95	0.0345
99	0.0508

Estos percentiles representan los valores de error de reconstrucción por debajo de los cuales se encuentran el 90%, 95% y 99% de las muestras de los datos normales respectivamente.

El percentil 90 indica que el 90% de los errores de reconstrucción son menores o iguales a 0.0299. Esto implica que la mayoría de las muestras normales tienen errores de reconstrucción relativamente bajos, lo que es un buen indicador de que el modelo ha aprendido correctamente de los datos normales, además, como se mencionó anteriormente, estos pueden ser útiles para establecer el umbral de detección de anomalías, teniendo en cuenta que la sensibilidad del modelo será mayor, esto dependerá mucho de la variabilidad del sistema, para el presente caso del motorreductor no podría considerarse de tener variables muy cambiantes como las vibraciones de las dos etapas del reductor planetario.

El percentil 95 muestra donde se encuentra el 95% de los errores de reconstrucción, por lo que se interpreta que en este porcentaje los datos se encuentran por debajo de 0.0345 tal cual la tabla 7. Este es un indicador de que los datos normales tienen errores de reconstrucción muy cercanos al promedio, con una ligera dispersión. Este percentil suele ajustarse para umbrales más estrictos para la detección de anomalías, ya que solo el 5% de las muestras deberían superar ese límite.

El percentil 99 indica que solo el 1% de las muestras normales tienen un error de reconstrucción superior a 0.0508. Este valor representa el umbral superior de los errores de reconstrucción para las muestras dentro de la "normalidad". Este percentil suele utilizarse para modelos de detección de anomalías más conservadores y menos sensible a falsos positivos, ya que solo el 1% de las muestras superarán los errores mayores a este valor de umbral.

Conclusiones

Teniendo en cuenta los objetivos planteados al inicio de la presente tesis, quedan demostrados los buenos resultados del diseño de red neuronal Autoencoder para la detección de anomalías, consiguiendo un error de reconstrucción de 0.0225, lo cual es significativo que el modelo ha conseguido aprender adecuadamente los patrones de datos normales. Esto llevado a un entorno agroindustrial logra identificar con antelación eventos que ayudan a prevenir fallos en motorreductores.

Del análisis de correlación realizado se concluye que este proceso permitió identificar las relaciones lineales por encima de 0.9, las cuales fueron consideradas como significativas entre las variables del sistema, destacando agrupaciones con alta correlación que sugieren la redundancia de la información en ciertos parámetros. Aquí se debió tener en cuenta, además, la parte operativa del motorreductor como el ejemplo del torque, que pese a tener una alta correlación por encima de 0.9 con señales como la corriente del motor, es una variable crítica que aporta información específica sobre las condiciones mecánicas, como el esfuerzo aplicado sobre el eje, que otras variables no logran reflejar directamente.

El análisis de convergencia del modelo demuestra que el autoencoder ha alcanzado un equilibrio óptimo después de 684 épocas, evidenciando un buen ajuste sin signos de sobreajuste ni subajuste. Esto se refleja en la consistencia de la tendencia observada entre los conjuntos de entrenamiento y validación, lo cual indica que el modelo está generalizando adecuadamente y aprendiendo efectivamente de las características intrínsecas de los datos.

Según uno de los objetivos planteados, se realizó el análisis técnico correspondiente llegando a la conclusión en complemento con el desarrollo de red neuronal, que uno de los principales tipos de fallas encontrados fueron las sobrecargas, vibraciones fuera de límite tanto para la etapa I como la etapa II del reductor ('reductor/VT_I y reductor/VT_II'), desgastes específicamente en los rodamientos ya que al tener monitoreadas las temperaturas en estos puntos específicos tanto para el motor eléctrico como el reductor, se encontraron casos de elevaciones de temperatura en las señales 'motor/TT_DE' y 'motor/TT_NDE' que hacen referencia a las temperaturas en los puntos de apoyo del motor. Esta es una información relevante para un equipo de mantenimiento de planta, ya que el modelo permite desglosar el desfase en señales lo que ayuda a identificar rápidamente que parámetro tiene mayor influencia sobre la anomalía y con esto tener un mejor plan de trabajo de mantenimiento preventivo.

Siguiendo con los objetivos planteados acerca del desarrollo de red neuronal se concluye que, el modelo obtiene buenos resultados y es aplicable en un entorno agroindustrial real ya se tiene una distribución del error muy cercana al promedio, con una tendencia hacia valores bajos y una desviación estándar de 0.007, medida que indica una baja dispersión entre datos. Esto es fundamental para seleccionar adecuadamente un umbral de anomalías capaz de detectar desviaciones significativas. El desempeño de este está directamente influenciado

por las condiciones de operación de un motorreductor y mantenimiento, lo que minimiza el impacto de las fallas.



Referencias

- Calderón, F., Franco, K., Pérez, J. (2023). Detección de anomalías en series de tiempo del funcionamiento de bombas electrosumergibles (BES) de producción en un campo de petróleo en Colombia. Universidad de los Andes. Recuperado de <https://hdl.handle.net/1992/73121>
- Casado N. (2022). Redes neuronales convolucionales y aplicaciones [Tesis para título, Universidad Complutense de Madrid] Docta Complutense. <https://docta.ucm.es/entities/publication/313b5a83-65c6-4177-b62d-c1e3711dc837>
- Castilla F. (2019). Detección y prognosis de anomalías aplicada a máquinas industriales [Tesis para Máster, Universitat Oberta de Catalunya]. UOC. <https://openaccess.uoc.edu/>
- Chapman S. (2012). *Máquinas Eléctricas*. (3a ed.) Editorial McGraw-Hill
- Correa C., Escobar J., Mena J. (2023). Implementación de analítica de datos para el mantenimiento predictivo de motores eléctricos [Tesis para título, Institución Universitaria Pascual Bravo]. Repositorio Institucional. <https://repositorio.pascualbravo.edu.co/>
- Escuela Especializada en Ingeniería ITCA-FEPADE (2018). *Algoritmos de aprendizaje automático para análisis y predicción de datos*. Revista Tecnológica, 11, 36-40
- Flovik V. (2018). *How to use machine learning for anomaly detection and condition monitoring*. Towards Data Science. Medium. <https://towardsdatascience.com/how-to-use-machine-learning-for-anomaly-detection-and-condition-monitoring-6742f82900d7>
- Fraille J. (2008). *Máquinas Eléctricas*. (6a ed.) Editorial McGraw-Hill
- García, J. R., & Rodríguez, E. A. (2017). Sistema de control automático para el monitoreo y control de nivel en un tanque de almacenamiento. CENGICANA. Recuperado de <https://cengicana.org/files/20170102124850310.pdf>
- Geron A. (2023). *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems* (Tercera Edición). Editorial O'reilly
- González, L., & Pérez, M. (2022). Estimación de áreas de cultivo mediante Deep Learning y programación convencional. arXiv. <https://arxiv.org/abs/2207.12310>
- Kibish S. (2018). *A note about finding anomalies*. Towards Data Science. Medium. <https://towardsdatascience.com/a-note-about-finding-anomalies-f9cedee38f0b>
- Nuñez-Sanchez F. (2016). Diseño de un sistema de reconocimiento automático de matrículas de vehículos mediante una red neuronal convolucional [Tesis para Máster en Ingeniería Informática, Universitat Oberta de Catalunya] UOC. <https://openaccess.uoc.edu/>

- Rivas W., Mazón B. (2018). *Redes neuronales artificiales aplicadas al reconocimiento de patrones* (Primera Edición). Editorial UTMACH, 2018
- Salas R. (s.f.). *Redes Neuronales Artificiales*. [PDF]
- ScienceDirect. (n.d.). Moving average filter. En ScienceDirect Topics. Recuperado el 13 de noviembre de 2024, de <https://www.sciencedirect.com/topics/engineering/moving-average-filter#:~:text=Moving%20Average%20Filters&text=The%20moving%20average%20is%20the,retaining%20a%20sharp%20step%20response>.
- SHAP. (n.d.). *An introduction to explainable AI with Shapley values*. SHAP Documentation. Retrieved November 21, 2024, from https://shap.readthedocs.io/en/latest/example_notebooks/overviews/An%20introduction%20to%20explainable%20AI%20with%20Shapley%20values.html
- Shi K., Song H., Yu H., Zhu Z. (2016). Collaborative Contextual Anomaly Detection for Industrial Equipment Groups. *2016 9th International Symposium on Computational Intelligence and Design (ISCID)*. <https://doi.org/10.1109/ISCID.2016.1094>
- Wenig, J., & others. (2022). Optimizing query processing for machine learning algorithms in modern data systems. *Proceedings of the VLDB Endowment*, 15(8), 1779-1792. Recuperado de <https://www.vldb.org/pvldb/vol15/p1779-wenig.pdf>
- Zapeta, A., Galindo, G., Juan, S., & Martínez, M. (2022). Métricas de rendimiento para evaluar el aprendizaje automático en la clasificación de imágenes petroleras utilizando redes neuronales convolucionales. Recuperado de <https://ciencialatina.org/index.php/cienciala/article/view/3420>

Apéndices



Apéndice A. Modelo de Red Neuronal Autoencoder para la detección de anomalías en motorreductores de molinos de secado

```
# Importación de librerías
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
import os
from sklearn import preprocessing
import tensorflow
from numpy.random import seed
from keras.layers import Dense
from keras.models import Model, Sequential, load_model
from keras.layers import Input, Dropout, Dense, LSTM, TimeDistributed,
RepeatVector
from keras.models import Model
from keras.models import model_from_json
from keras import regularizers
from joblib import dump, load
import tensorflow as tf
from tensorflow.keras.models import load_model
from keras.callbacks import EarlyStopping
from sklearn.metrics import accuracy_score, precision_score, recall_score
from numpy.random import seed
import shap
matplotlib.rcParams['figure.figsize'] = [20, 8]
sns.set(style='darkgrid', palette='dark')

#ignorar warnings
```

```
import warnings
warnings.filterwarnings('ignore')

# Establecer la semilla para TensorFlow y NumPy
tf.random.set_seed(10)
np.random.seed(10)

# Carga de datos

df = pd.read_csv('DATOS MOTOR MOLINO/data_molino_2023.csv', index_col=0)
df.index = pd.to_datetime(df.index)
df.head() # Mostrar las primeras 5 filas

df.shape

list(df)

df.info()

df = df.drop(columns=['ZT_MasaSupLM ValueY',
                    'ZT_MasaSupLL ValueY',
                    'TT_UH ValueY',
                    'AT_UH ValueY',
                    'ESTADO_UH'])

df['ESTADO_Motor'] = df['ESTADO_Motor'].ffill().bfill()

# Crear un nuevo DataFrame con la información relevante
df_info = pd.DataFrame({
    "Column": df.columns,
    "Non-Null Count": df.notnull().sum(),
    "Dtype": df.dtypes
}).reset_index(drop=True)
```

```

# Exportar como CSV o mostrar
df_info.to_csv('df_info.csv', index=False) # Guardar si es necesario
df_info

# Filtrado de señales

df['ESTADO_Motor'] =
df['ESTADO_Motor'].rolling(window=20,min_periods=1).mean()

df['I_MortorUH ValueY'] = df['I_MortorUH
ValueY'].rolling(window=20,min_periods=1).mean()

df['I_Motor ValueY'] = df['I_Motor
ValueY'].rolling(window=20,min_periods=1).mean()

df['LT_ChuteMolinoA ValueY'] = df['LT_ChuteMolinoA
ValueY'].rolling(window=20,min_periods=1).mean()

df['LT_ChuteMolinoB ValueY'] = df['LT_ChuteMolinoB
ValueY'].rolling(window=20,min_periods=1).mean()

df['PT_UH ValueY'] = df['PT_UH
ValueY'].rolling(window=20,min_periods=1).mean()

df['ST_Motor ValueY'] = df['ST_Motor
ValueY'].rolling(window=20,min_periods=1).mean()

df['TT_MasaCanLL ValueY'] = df['TT_MasaCanLL
ValueY'].rolling(window=20,min_periods=1).mean()

df['TT_MasaCanLM ValueY'] = df['TT_MasaCanLM
ValueY'].rolling(window=20,min_periods=1).mean()

df['TT_MasaSupLL ValueY'] = df['TT_MasaSupLL
ValueY'].rolling(window=20,min_periods=1).mean()

df['TT_MasaSupLM ValueY'] = df['TT_MasaSupLM
ValueY'].rolling(window=20,min_periods=1).mean()

df['TT_MasaSupLM ValueY'] = df['TT_MasaSupLM
ValueY'].rolling(window=20,min_periods=1).mean()

df['TT_MotorDE ValueY'] = df['TT_MotorDE
ValueY'].rolling(window=20,min_periods=1).mean()

df['TT_MotorNDE ValueY'] = df['TT_MotorNDE
ValueY'].rolling(window=20,min_periods=1).mean()

df['TT_MotorR ValueY'] = df['TT_MotorR
ValueY'].rolling(window=20,min_periods=1).mean()

df['TT_MotorS ValueY'] = df['TT_MotorS
ValueY'].rolling(window=20,min_periods=1).mean()

df['TT_MotorT ValueY'] = df['TT_MotorT
ValueY'].rolling(window=20,min_periods=1).mean()

```

```
df['TorqueReductor ValueY'] = df['TorqueReductor
ValueY'].rolling(window=20).mean()
```

```
df['VT_MortorNDE ValueY'] = df['VT_MortorNDE
ValueY'].rolling(window=4,min_periods=1).mean()
```

```
df['VT_MotorDE ValueY'] = df['VT_MotorDE
ValueY'].rolling(window=4,min_periods=1).mean()
```

```
df['VT_ReductorI ValueY'] = df['VT_ReductorI
ValueY'].rolling(window=4,min_periods=1).mean()
```

```
df['VT_ReductorII ValueY'] = df['VT_ReductorII
ValueY'].rolling(window=4,min_periods=1).mean()
```

```
df = df[df['ESTADO_Motor']!=0]
```

```
df = df.drop(columns=['ESTADO_Motor'])
```

```
df
```

```
df.describe()
```

```
columnas = ['masas/TT_MasaSupLM',
            'masas/TT_MasaCanLM',
            'masas/TT_MasaSupLL',
            'masas/TT_MasaCanLL',
            'reductor/LT_ChuteA',
            'reductor/LT_ChuteB',
            'UH/I_UH',
            'UH/PT_UH',
            'motor/TT_R',
            'motor/TT_S',
            'motor/TT_T',
            'motor/TT_NDE',
            'motor/TT_DE',
            'motor/VT_NDE',
            'motor/VT_DE',
            'motor/I',
            'motor/ST',
            'reductor/VT_I',
```

```
'reductor/VT_rII',
'reductor/Torque']
```

```
df.columns = columnas
```

```
df.isnull().sum()
```

```
plt.rc('font',size = 10)
```

```
plt.rc('axes',labelsize=10,titlesize=10)
```

```
plt.rc('legend',fontsize = 10)
```

```
plt.rc('xtick',labelsize = 5)
```

```
plt.rc('ytick',labelsize = 5)
```

```
df.hist(bins=50,figsize=(16,12))
```

```
plt.show()
```

```
# Reglas de limpieza
```

```
df['UH/I_UH'] = df['UH/I_UH'].clip(lower=3.5,upper=5)
```

```
df['UH/PT_UH'] = df['UH/PT_UH'].clip(lower=0.5,upper=5)
```

```
df['masas/TT_MasaCanLL'] = df['masas/TT_MasaCanLL'].clip(lower=20,upper=100)
```

```
df['masas/TT_MasaCanLM'] = df['masas/TT_MasaCanLM'].clip(lower=20,upper=100)
```

```
df['masas/TT_MasaSupLL'] = df['masas/TT_MasaSupLL'].clip(lower=20,upper=100)
```

```
df['masas/TT_MasaSupLM'] = df['masas/TT_MasaSupLM'].clip(lower=20,upper=100)
```

```
df['motor/I'] = df['motor/I'].clip(lower=300,upper=1400)
```

```
df['motor/ST'] = df['motor/ST'].clip(lower=300,upper=1400)
```

```
df['motor/TT_DE'] = df['motor/TT_DE'].clip(lower=0,upper=110)
```

```
df['motor/TT_NDE'] = df['motor/TT_NDE'].clip(lower=0,upper=110)
```

```
df['motor/TT_R'] = df['motor/TT_R'].clip(lower=0,upper=140)
```

```
df['motor/TT_S'] = df['motor/TT_S'].clip(lower=0,upper=140)
```

```
df['motor/TT_T'] = df['motor/TT_T'].clip(lower=0,upper=140)
```

```
df['motor/VT_DE'] = df['motor/VT_DE'].clip(lower=0,upper=7)
```

```
df['motor/VT_NDE'] = df['motor/VT_NDE'].clip(lower=0,upper=7)
```

```
df['reductor/LT_ChuteA'] = df['reductor/LT_ChuteA'].clip(lower=0,upper=100)
```

```
df['reductor/LT_ChuteB'] = df['reductor/LT_ChuteB'].clip(lower=0,upper=100)
df['reductor/Torque'] = df['reductor/Torque'].clip(lower=450,upper=1500)
df['reductor/VT_I'] = df['reductor/VT_I'].clip(lower=0,upper=15)
df['reductor/VT_rII'] = df['reductor/VT_rII'].clip(lower=0,upper=15)
```

```
# Eliminar filas con valores nulos
```

```
df=df.dropna()
```

```
df.shape
```

```
df_temp = df.copy()
```

```
ruta_archivo = 'DATOS MOTOR MOLINO/images/'
```

```
os.makedirs(ruta_archivo,exist_ok=True)
```

```
for item in df.columns:
```

```
    nombre_archivo = item.replace('/', '_')
```

```
    plt.figure(figsize=(20,4))
```

```
    plt.plot(df_temp[item],label=item)
```

```
    plt.title(item)
```

```
    plt.legend()
```

```
    plt.grid(True)
```

```
    plt.savefig(ruta_archivo+nombre_archivo+'.png')
```

```
    plt.show()
```

```
# Crear una máscara para el triángulo superior
```

```
mask = np.triu(np.ones_like(df.corr(), dtype=bool))
```

```
plt.figure(figsize=(20,12))
```

```
sns.heatmap(df.corr(),mask=mask,annot = True,fmt='.4f',cmap='Blues')
```

```
plt.title('Matriz de Correlacion')
```

```
plt.show()
```

```
# Se divide el df en df_train y df_test
```

```

n_rows = df.shape[0]
train_size = int(n_rows*0.9)

df_train = df.iloc[:train_size]
df_test = df.iloc[train_size:]

df_train.shape, df_test.shape

# Modelo de Deep Learning

# Normalización de la data
scaler = preprocessing.MinMaxScaler()
X_train = scaler.fit_transform(df_train)
X_test = scaler.fit_transform(df_test)

#Reshape inputs for LSTM
X_train = X_train.reshape(X_train.shape[0],1,X_train.shape[1])
print('Training data shape:',X_train.shape)
X_test = X_test.reshape(X_test.shape[0],1,X_test.shape[1])
print('Test data shape:',X_test.shape)

# Se define la red neuronal autoencoder
def autoencoder_model(X):
    inputs = Input(shape=(X.shape[1], X.shape[2]))
    L1 = LSTM(12,activation='relu',return_sequences=True,
              kernel_regularizer = regularizers.l2(0.00))(inputs)
    L2 = LSTM(4,activation='relu',return_sequences=False)(L1)
    L3 = RepeatVector(X.shape[1])(L2)
    L4 = LSTM(4,activation='relu',return_sequences=True)(L3)
    L5 = LSTM(12,activation='relu',return_sequences=True)(L4)
    output = TimeDistributed(Dense(X.shape[2]))(L5)
    model = Model(inputs=inputs,outputs=output)

```

```
return model

# Creacion del modelo autoencoder
model = autoencoder_model(X_train)
model.compile(optimizer='adam',loss='mae')
model.summary()

# Configuracion del EarlyStopping
early_stop =
EarlyStopping(monitor='val_loss',patience=200,restore_best_weights=True)

# Ajustar el modelo de datos
nb_epochs = 1000
batch_size = 500
history = model.fit(X_train,X_train,epochs=nb_epochs,batch_size=batch_size,
                    validation_split = 0.10, shuffle = True, callbacks =
[early_stop]).history

# Plot the training losses
fig, ax = plt.subplots(figsize=(14,6),dpi=100)
ax.plot(history['loss'],'b',label='Train',linewidth = 2)
ax.plot(history['val_loss'],'r',label='Validation',linewidth = 2)
ax.set_title('Model loss',fontsize = 16)
ax.set_ylabel('Loss (mae)')
ax.set_xlabel('Epoch')
ax.legend(loc='upper right')
plt.show()

# Predecir con el modelo
X_pred = model.predict(X_train)
X_pred = X_pred.reshape(X_pred.shape[0],X_pred.shape[2])
X_pred = pd.DataFrame(X_pred,columns=df_train.columns)
X_pred_original = scaler.inverse_transform(X_pred)
```

```
X_pred.index = df_train.index

# Calcular la perdida de MAE
scored_train = pd.DataFrame(index=df_train.index)
X_train_reshaped = X_train.reshape(X_train.shape[0],X_train.shape[2])
scored_train['Loss_mae'] = np.mean(np.abs(X_pred - X_train_reshaped),axis=1)

# Configurar la figura de Matplotlib
plt.figure(figsize=(16,9),dpi=80)
plt.title('Loss Distribution',fontsize=16)

# Crear histograma de la distribución de perdida
sns.histplot(scored_train['Loss_mae'],bins=20,kde=True,color='Blue')
plt.xlim([0.0,0.5])

# Mostrar grafico
plt.show()

# Se calcula un threshold de anomalia
mean_loss = np.mean(scored_train['Loss_mae'])
std_loss = np.std(scored_train['Loss_mae'])
threshold = mean_loss + 6 * std_loss
print(f'Threshold es: {threshold}')

scored_train['Threshold'] = threshold
scored_train['Anomaly'] = scored_train['Loss_mae'] > scored_train['Threshold']
# Convertir scored Anomaly en booleano
scored_train['Anomaly'] = scored_train['Anomaly'].astype(int)

# Evaluacion de la Red Neuronal

X_test_pred = model.predict(np.array(X_test))
```

```

X_test_pred = model.predict(X_test)
X_test_pred = X_test_pred.reshape(X_test_pred.shape[0],X_test_pred.shape[2])
X_test_pred = pd.DataFrame(X_test_pred, columns= df_test.columns)
X_test_pred.index = df_test.index
scored_test = pd.DataFrame(index=df_test.index)
X_test_reshaped = X_test.reshape(X_test.shape[0],X_test.shape[2])
scored_test['Loss_mae'] = np.mean(np.abs(X_test_pred - X_test_reshaped),
axis=1)
scored_test['Threshold'] = threshold
scored_test['Anomaly'] = scored_test['Loss_mae'] > scored_test['Threshold']
# Convertir scored Anomaly en booleano
scored_test['Anomaly'] = scored_test['Anomaly'].astype(int)

fig,ax = plt.subplots(figsize = (10,6))
scored_train[['Loss_mae', 'Threshold']].plot(logy=False,color=['blue', 'red'],ax
=ax)
scored_test[['Loss_mae', 'Threshold']].plot(logy=False,color=['green', 'red'],ax
=ax,linestyle='--')
plt.show()

scored = pd.concat([scored_train,scored_test],axis=0)
scored.columns=['Factor de Anomalia','Umbral Anomalia','Anomalia']

scored['Factor de Anomalia'] = 100*scored['Factor de Anomalia']/scored['Umbral
Anomalia']
scored['Factor de Anomalia'] = scored['Factor de Anomalia'].clip(upper=110)

scored['Umbral Anomalia'] = 100
scored.tail()

scored.to_csv('DATOS MOTOR MOLINO/data/scored_motorreductor.csv',index=True)

scored['Factor de Anomalia'].plot(figsize=(20,6),color='blue',title='Score de
Anomalias')

```

```

# Función de predicción compatible con SHAP
def model_predict(X):
    if len(X.shape) == 2:
        X = X.reshape((X.shape[0], 1, X.shape[1]))
    preds = model.predict(X)
    preds_reshaped = preds.reshape(preds.shape[0], preds.shape[2])
    return preds_reshaped

n = 100

# Tomar una muestra aleatoria de 100 de los datos para acelerar el tiempo de
# cómputo
X_sample = X_train[np.random.choice(X_train.shape[0], n)]

# Resumir el conjunto de datos de fondo usando shap.kmeans
background = shap.kmeans(X_sample.reshape(X_sample.shape[0],
X_sample.shape[2]), 100) #n

# Crear el KernelExplainer con el conjunto de datos resumido
explainer = shap.KernelExplainer(model_predict, background)

# Calcular los valores SHAP para todas las instancias de entrenamiento
shap_values = explainer.shap_values(X_sample.reshape(X_sample.shape[0],
X_sample.shape[2]))

# Eliminar la tercera dimensión (donde todos los valores son cero)
shap_values_2d = np.array([shap_values[i][:, 0] for i in
range(len(shap_values))])

# Crear un DataFrame de los valores SHAP
df_shap = pd.DataFrame(shap_values_2d, columns=df.columns)

# Visualizar los valores SHAP
shap.summary_plot(shap_values_2d, X_sample.reshape(X_sample.shape[0],
X_sample.shape[2]), feature_names=df.columns, plot_type='bar')

```

```

# Visualizar los valores SHAP
shap.summary_plot(shap_values_2d, X_sample.reshape(X_sample.shape[0],
X_sample.shape[2]), feature_names=df.columns)

shap.initjs()
# Función de predicción compatible con SHAP
def model_predict(X):
    if len(X.shape) == 2:
        X = X.reshape((X.shape[0], 1, X.shape[1]))
    preds = model.predict(X)
    preds_reshaped = preds.reshape(preds.shape[0], preds.shape[2])
    return preds_reshaped

n = 100
# Tomar una muestra aleatoria de 100 de los datos para acelerar el tiempo de
cómputo
X_sample = X_train[800:900,:]

# Resumir el conjunto de datos de fondo usando shap.kmeans
background = shap.kmeans(X_sample.reshape(X_sample.shape[0],
X_sample.shape[2]), 10)

# Crear el KernelExplainer con el conjunto de datos resumido
explainer = shap.KernelExplainer(model_predict, background)

# Calcular los valores SHAP para todas las instancias de la muestra
shap_values = explainer.shap_values(X_sample.reshape(X_sample.shape[0],
X_sample.shape[2]))

# Eliminar la tercera dimensión (si aplica)
shap_values_2d = np.array([shap_values[i][:, 0] for i in
range(len(shap_values))])

# Seleccionar una instancia específica para el gráfico force_plot
# Cambiar el índice `instance_index` para visualizar diferentes instancias

```

```
instance_index = np.argmax(np.abs(shap_values_2d.sum(axis=1))) # Cambiar este
valor según la instancia que quieras visualizar
```

```
# Visualizar el force_plot para una instancia específica
```

```
shap.force_plot(
    explainer.expected_value[0], # Valor esperado del modelo
    shap_values_2d[instance_index], # Valores SHAP para la instancia
    seleccionada
    X_sample.reshape(X_sample.shape[0], X_sample.shape[2])[instance_index], #
    Características de la instancia seleccionada
    feature_names=df.columns # Nombres de las características
)
```

```
# Error de reconstrucción promedio
```

```
mae_error = np.mean((X_test_pred-X_test_reshaped)**2,axis=1)
```

```
mae_avg = np.mean(mae_error)
```

```
mae_avg
```

```
mse_error = np.mean((X_test_pred-X_test_reshaped)**2,axis=1)
```

```
mse_avg = np.mean(mse_error)
```

```
percentiles = np.percentile(mse_error,[90,95,99])
```

```
percentiles
```


```
mse_std = np.std(mse_error)
```

```
mse_std
```

Anexos



Anexo A. Ficha Técnica del motor eléctrico

HOJA DE DATOS					
Motor Trifásico de Inducción - Rotor de Jaula					
Cliente :					
Línea del producto		: W50 Trifásico		Código del producto : 13038194	
Carcasa		: 6808/09		Tiempo de rotor bloqueado	
Potencia		: 900 HP (660 kW)		: 32s (frío) 18s (caliente)	
Polos		: 6		Elevación de temperatura	
Frecuencia		: 60 Hz		: 80 K	
Tensión nominal		: 4160 V		Régimen de servicio	
Corriente nominal		: 118 A		: Cont.(S1)	
Corriente de arranque		: 873 A		Temperatura ambiente	
Ip/In		: 7.4x(Cód. H)		: -20°C hasta +40°C	
Corriente en vacío		: 48.0 A		Altitud	
Rotación nominal		: 1192 rpm		: 1000 m	
Resbalamiento		: 0.67 %		Grado de protección	
Torque nominal		: 548 kgfm		: IP55	
Torque de arranque		: 140 %		Método de refrigeración	
Torque máximo		: 229 %		: IC411 - TEFC	
Clase de aislamiento		: F		Forma constructiva	
Factor de servicio		: 1.00		: F-1	
Momento de inercia (J)		: 31.1 kgm ²		Sentido de giro ¹	
				: Ambos	
				Nivel de ruido ²	
				: 81.0 dB(A)	
				Método de Arranque	
				: Partida directa	
				Masa aproximada ³	
				: 3350 kg	
Potencia		50% 75% 100%			
Rendimiento (%)		95.4 95.8 95.8			
Cos Φ		0.65 0.75 0.81			
Notas					
Esta revisión reemplaza y cancela la anterior, la cual deberá ser eliminada. (1) Mirando la punta delantera del eje del motor. (2) Medido a 1m y con tolerancia de +3dB(A). (4) Al 100% de la carga completa.			Los valores indicados son valores promedio con base en ensayos y para alimentación en red senoidal, sujetos a las tolerancias de la norma NEMA MG-1.		
Rev.	Resumen de los cambios		Ejecutado	Verificado	Fecha
Ejecutor					
Verificador				Pagina	Revisión
Fecha	24/11/2024			1 / 5	

CURVA DE PAR Y CORRIENTE X ROTACIÓN

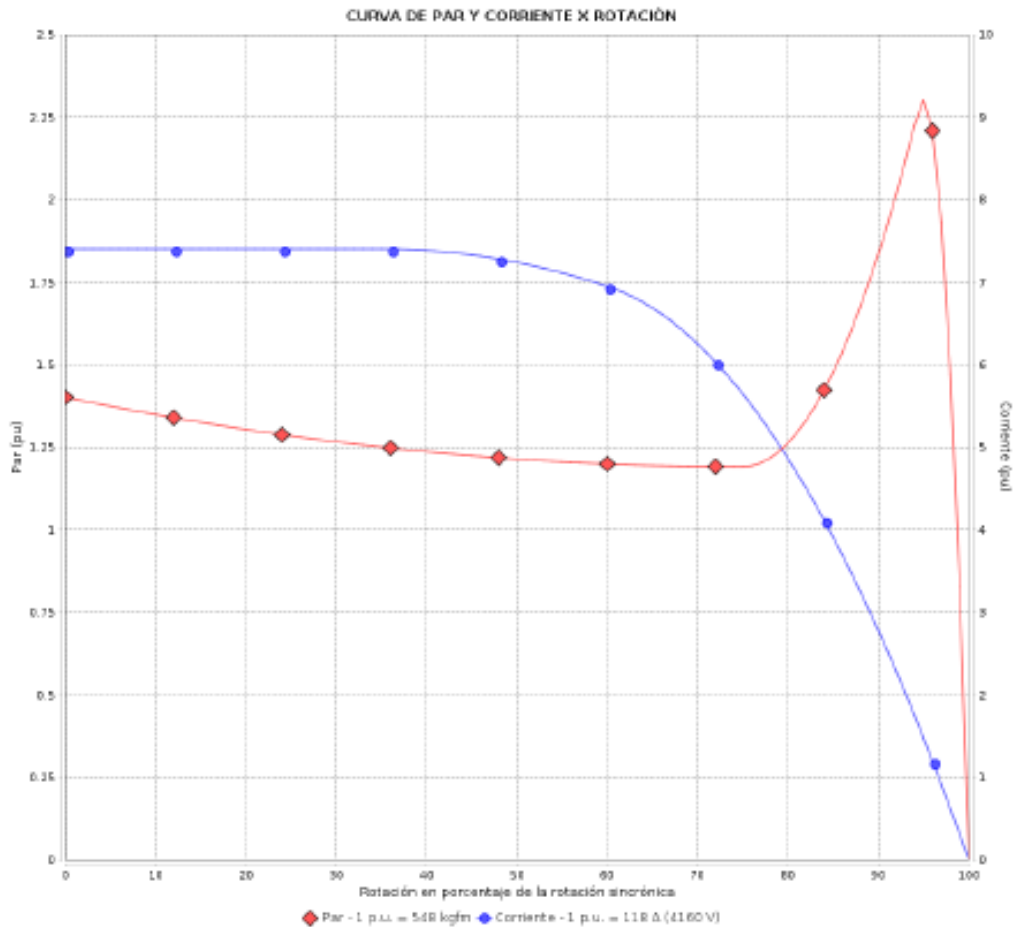
Motor Trifásico de Inducción - Rotor de Jaula



Cliente :

Línea del producto : W50 Trifásico

Código del producto : 13038194



Desempeño : 4160 V 60 Hz 6P

Corriente nominal	: 118 A	Momento de inercia (J)	: 31.1 kgm ²
Ip/In	: 7.4	Régimen de servicio	: Cont.(S1)
Torque nominal	: 548 kgfm	Clase de aislamiento	: F
Torque de arranque	: 140 %	Factor de servicio	: 1.00
Torque máximo	: 229 %	Elevación de temperatura	: 80 K
Rotación nominal	: 1192 rpm		

Tiempo de rotor bloqueado : 32s (frío) 18s (caliente)

Rev.	Resumen de los cambios	Ejecutado	Verificado	Fecha
Ejecutor			Página	Revisión
Verificador			2 / 5	
Fecha	24/11/2024			

Propiedad de WEG S/A. Prohibida la reproducción sin permiso.
Sujeto a cambios sin previo aviso

CURVA DE DESEMPEÑO EN CARGA

Motor Trifásico de Inducción - Rotor de Jaula

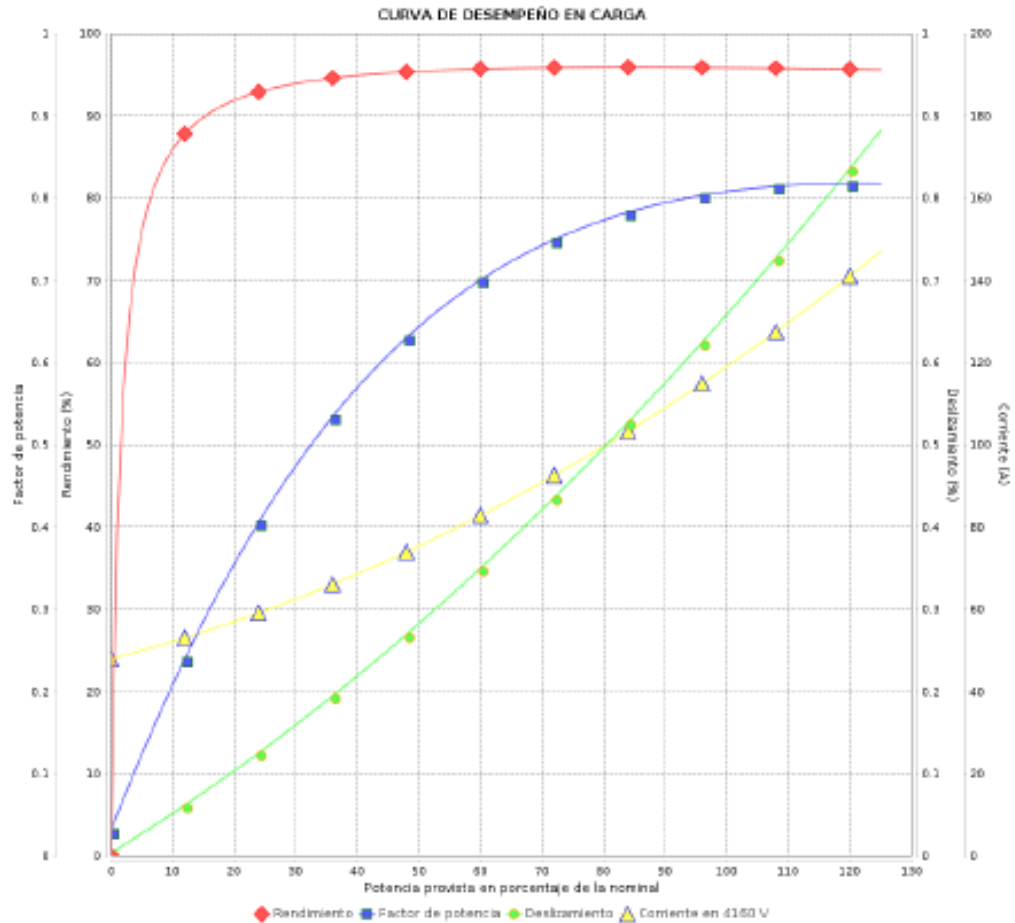


Cliente :

Línea del producto : W50 Trifásico

Código del producto :

13038194



Desempeño : 4160 V 60 Hz 6P

Corriente nominal : 118 A

Ip/In : 7.4

Torque nominal : 548 kgfm

Torque de arranque : 140 %

Torque máximo : 229 %

Rotación nominal : 1192 rpm

Momento de inercia (J) : 31.1 kgm²

Régimen de servicio : Cont.(S1)

Clase de aislamiento : F

Factor de servicio : 1.00

Elevación de temperatura : 80 K

Rev.	Resumen de los cambios	Ejecutado	Verificado	Fecha
Ejecutor				
Verificador			Página	Revisión
Fecha	24/11/2024		3 / 5	

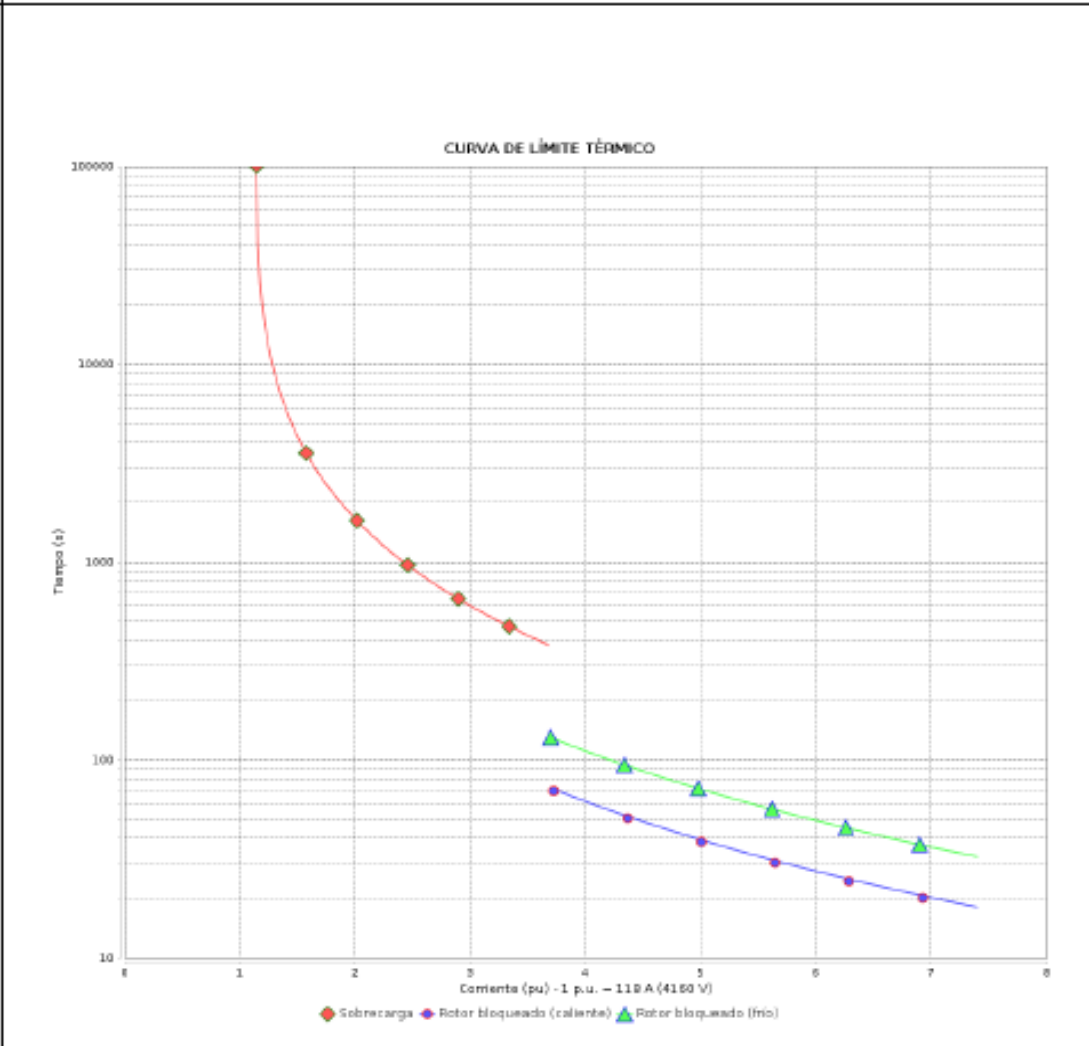
Propiedad de WEG S/A. Prohibida la reproducción sin permiso.
Sujeto a cambios sin previo aviso

CURVA DE LÍMITE TÉRMICO

Motor Trifásico de Inducción - Rotor de Jaula



Cliente	:	
Línea del producto	: W50 Trifásico	Código del producto : 13038194



Desempeño	: 4160 V 60 Hz 6P		
Corriente nominal	: 118 A	Momento de inercia (J)	: 31.1 kgm ²
Ip/In	: 7.4	Régimen de servicio	: Cont.(S1)
Torque nominal	: 548 kgfm	Clase de aislamiento	: F
Torque de arranque	: 140 %	Factor de servicio	: 1.00
Torque máximo	: 229 %	Elevación de temperatura	: 80 K
Rotación nominal	: 1192 rpm		

Constante de calentamiento	
Constante de enfriamiento	

Rev.	Resumen de los cambios	Ejecutado	Verificado	Fecha
Ejecutor				
Verificador				
Fecha	24/11/2024		Página 4 / 5	Revisión

Propiedad de WEG S/A. Prohibida la reproducción sin permiso. Sujeto a cambios sin previo aviso.

Nota. Extraído de la ficha técnica de moto trifásico