



Universidad
Nacional
de Loja

Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales no
Renovables

Carrera de Ingeniería Electromecánica

Exploración y aplicación de "*Diagnostic Feature Designer*" para
desarrollar modelos de diagnóstico de fallos en plantas eólicas

Trabajo de Titulación, previo a la
obtención del título de Ingeniero
Electromecánico

AUTOR:

Wilmer Andrés Santos Ordoñez

DIRECTOR:

Ing. José Fabricio Cuenca Granda Mg. Sc.

Loja - Ecuador

2024

Certificación

Loja, 24 de julio de 2024

Ing. José Fabricio Cuenca Granda, Mg. Sc.

DIRECTOR DE TRABAJO DE TITULACIÓN

CERTIFICO:

Que he revisado y orientado todo el proceso de elaboración del Trabajo de Titulación denominado: **Exploración y aplicación de "DIAGNOSTIC FEATURE DESIGNER" para desarrollar modelos de diagnóstico de fallos en plantas eólicas**, previo a la obtención del título de **Ingeniero electromecánico**, de la autoría del estudiante **Wilmer Andrés Santos Ordoñez** con cédula de identidad Nro. **1105424848**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.

Ing. José Fabricio Cuenca Granda, Mg. Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

Autoría

Yo, **Wilmer Andrés Santos Ordoñez**, declaro ser autor del presente trabajo de titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mí del trabajo de titulación en el Repositorio Digital Institucional – Biblioteca Virtual.



Firma:

Cédula de identidad: 1105424848

Fecha: 24 de julio de 2024.

Correo electrónico: wilmer.santos@unl.edu.ec

Teléfono: 0961680958

Carta de autorización por parte del autor, para consulta, reproducción parcial o total y/o publicación electrónica del texto completo, del Trabajo de Titulación.

Yo **Wilmer Andrés Santos Ordoñez** declaro ser autor del Trabajo de Titulación denominado: **Exploración y aplicación de "DIAGNOSTIC FEATURE DESIGNER" para desarrollar modelos de diagnóstico de fallos en plantas eólicas**, como requisito para optar por el título de **Ingeniero electromecánico**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional. Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad. La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Titulación que realice un tercero. Para constancia de esta autorización, en la ciudad de Loja, a los veinticuatro días del mes de julio del dos mil veinticuatro.



Firma:

Cédula de identidad: 1105424848

Dirección: Paltas – Casanga - Palo Blanco.

Fecha: 24 de julio de 2024.

Correo electrónico: wilmer.santos@unl.edu.ec

Teléfono: 0961680958

DATOS COMPLEMENTARIOS:

Director del Trabajo de Titulación: Ing. José Fabricio Cuenca Granda, Mg. Sc.

Dedicatoria

Dedicado a mis seres queridos: Wilmer, Mery y Doménica, quienes me han brindado su entrega, sacrificio y confianza para afrontar los desafíos de la exigente disciplina de la Ingeniería. Agradezco también a mis familiares, amigos y formadores por acompañarme en el camino del conocimiento y la disciplina.

Wilmer Andrés Santos Ordoñez

Agradecimiento

Mis sinceros agradecimientos a la Universidad Nacional de Loja, a la Facultad de Energía, Industrias y Recursos Naturales No Renovables, así como a la Carrera de Ingeniería Electromecánica, por brindarme la oportunidad de adquirir conocimientos en el campo de la Ingeniería Electromecánica y, de esta manera, poder contribuir al servicio de la sociedad y a las necesidades que aparecen frecuentemente en ingeniería.

Asimismo, mi agradecimiento al Ing. José Fabricio Cuenca Granda, por su disposición y compromiso en el asesoramiento de este trabajo de investigación y a la Corporación Eléctrica del Ecuador – CELEC, por los datos proporcionados de los aerogeneradores para el análisis de datos.

Wilmer Andrés Santos Ordoñez

Índice de contenidos

Portada	i
Certificación	ii
Autoría	iii
Dedicatoria	v
Agradecimiento	vi
Índice de contenidos	vii
Índice de tablas:	xi
Índice de figuras:.....	xii
Índice de anexos:.....	xiv
Simbología:	xv
1. Título	1
2. Resumen	2
3. Introducción	4
4. Marco teórico	6
4.1 Aerogeneradores.....	6
4.2 Mantenimiento predictivo	7
4.3 Aprendizaje Automático (Machine Learning)	7
4.4 DIAGNOSTIC FEATURE DESIGNER.....	8
4.4.1 ...Importar datos	9
4.4.2 ...Preprocesar los datos.....	10
4.4.2.1 Selección, limpieza y transformación de Datos.	10

4.4.2.2	Variables con valores faltantes	11
4.4.2.3	Variables con valores atípicos	11
4.4.2.4	Procesar datos	13
4.4.3	...Indicadores de condición	14
4.4.3.1	Características en el dominio del tiempo	15
4.4.3.2	Características en el dominio de tiempo-frecuencia.....	16
4.4.3.3	Características en el dominio de la frecuencia	17
4.4.3.4	Selección de características	17
4.4.4	...Entrenar modelos	18
5.	Metodología.....	22
5.1	Equipos y materiales	22
5.1.1	...Equipos	22
5.1.2	...Materiales.....	22
5.2	Procedimiento.....	22
5.3	Primero objetivo.....	22
5.3.1	...Preprocesamiento de datos.....	22
5.3.2	...Importar datos.	23
5.3.3	...Procesamiento de datos.....	23
5.3.4	...Generación de características.....	23
5.3.5	...Selección de características.....	23
5.3.6	...Exportar datos.	23
5.3.7	...Entrenar modelos.	24
5.4	Segundo objetivo.....	24
5.4.1	...Recolección de datos.....	25
5.4.2	...Preprocesamiento de los datos.	25
5.4.3	...Importar datos a DFD	26
5.4.4	...Procesar datos en DFD.....	27
5.4.5	...Indicadores de condición.	28
5.4.5.1	Características en el dominio del tiempo:	28
5.4.5.2	Características en el dominio tiempo-frecuencia:	28
5.4.5.3	Características en el dominio de la frecuencia:	28
5.4.6	...Selección de características.....	29

5.4.7...Entrenar modelos	29
5.5 Tercer objetivo	31
6. Resultados	32
6.1 Primer objetivo.....	32
6.2 Segundo objetivo.....	33
6.2.1...Recolección de datos.....	33
6.2.2...Preprocesamiento e importación de datos.....	34
6.2.3...Procesar los datos.....	38
6.2.4...Indicadores de condición	39
6.2.5...Características en el dominio del tiempo	39
6.2.6...Características en el dominio del tiempo-frecuencia	43
6.2.7...Características en el dominio de la frecuencia.....	44
6.2.8...Ranking de las características	45
6.2.9...Entrenar modelos	46
6.3 Tercer Objetivo	47
6.3.1...Guías para interpretar resultados.....	47
6.3.2...Pasos para el preprocesamiento de datos (limpieza de datos)	47
6.3.3...Pasos para importar datos	48
6.3.4...Pasos en el procesamiento de datos	49
6.3.5...Pasos para extraer características (dominio en el tiempo)	50
6.3.6...Pasos para extraer características (dominio en el tiempo-frecuencia)	51
6.3.7...Pasos para extraer características (dominio en la frecuencia).....	52
6.3.8...Selección de características (ranking features).....	53
6.3.9...Exportar características importantes	54
6.3.10.Programa (classification learner).....	54
7. Discusión.....	56
8. Conclusiones.....	58
9. Recomendaciones.....	59

10. Bibliografía.....	60
11. Anexos.....	62

Índice de tablas:

Tabla 1. Métodos para análisis de datos faltantes.....	11
Tabla 2. Métodos para encontrar valores atípicos.....	12
Tabla 3. Técnicas de procesamientos presentes en DFD.....	13
Tabla 4. Funciones de DFD para obtener características en el dominio del tiempo.....	15
Tabla 5. Funcionalidades de dominio tiempo-frecuencia.....	16
Tabla 6. Características en el dominio de la frecuencia.....	17
Tabla 7. Selección de características.....	17
Tabla 8. Tipos de modelos en la predicción por clasificación.....	19
Tabla 9. Funciones utilizadas en el preprocesamiento.....	26
Tabla 10. Documentación y características de DFD.....	32
Tabla 11. Análisis de los resultados estadísticos en el dominio del tiempo.....	41
Tabla 12. Análisis de los resultados estadísticos en el dominio del tiempo-frecuencia.....	44
Tabla 13. Análisis de los resultados estadísticos en el dominio de la frecuencia.....	45
Tabla 14. Resultados de los modelos entrenados en Classification Learner.....	47

Índice de figuras:

Figura 1. Transformación de la energía en turbinas eólicas.	6
Figura 2. Partes de un aerogenerador.	7
Figura 3. Gráfica mantenimiento predictivo.	7
Figura 4. Proceso de aprendizaje automático.	8
Figura 5. Interfaz DFD.	9
Figura 6. Flujo de trabajo de DFD.	9
Figura 7. Pasos generales para importar datos a DFD.	10
Figura 8. Técnicas de aprendizaje automático supervisado y no supervisado.	18
Figura 9. Matriz de confusión.	19
Figura 10. Método de búsqueda en Google Académico.	22
Figura 11. Ejemplo de preparación de datos para importar a DFD.	23
Figura 12. Metodología para documentar las funcionalidades de DFD.	24
Figura 13. Carpeta de archivos de destino en Matlab.	25
Figura 14. Temperaturas seleccionadas del convertidor.	26
Figura 15. Ventana para la selección y configuración de las variables.	27
Figura 16. Métodos de welch & autorregresivo.	27
Figura 17. Estadísticos en el dominio del tiempo.	28
Figura 18. Descomposición de modo empírico.	28
Figura 19. Estadísticos de la frecuencia.	28
Figura 20. Opciones de selección de características en DFD.	29
Figura 21. Ventana de selección de modelos de aprendizaje (Classification Learner).	30
Figura 22. Metodología para el desarrollo de modelos de diagnóstico	30
Figura 23. Esquema del manual de resultados.	31
Figura 24. Lista de fallos de los aerogeneradores.	33
Figura 25. Porcentaje de fallo en los aerogeneradores.	34
Figura 26. Datos limpios faltantes y atípicos.	34
Figura 27. Resultados preprocesamiento de datos aerogenerador.	36
Figura 28. Ensamble de tablas dentro de una matriz.	37
Figura 29. Ventana de interfaz DFD.	38
Figura 30. Formas de procesamiento de los datos.	38
Figura 31. Procesamiento: interpolación, análisis espectral (Welch & Autorregresivo).	39
Figura 32. Funcionalidad del programa para los indicadores de condición.	39
Figura 33. Características de las señales (Características en el dominio del tiempo).	39

Figura 34. Histograma de características en el tiempo.	40
Figura 35. Histograma 2 de las características en el tiempo.....	41
Figura 36. Histogramas de las características en el dominio del tiempo-frecuencia.....	43
Figura 37. Características en el dominio de la frecuencia.	45
Figura 38. Ranking de características: Análisis supervisado.....	46
Figura 39. Pasos para encontrar datos atípicos y faltantes.	48
Figura 40. Pasos en la construcción de la tabla de importación.	48
Figura 41. Pasos para importar la tabla de las temperaturas.....	49
Figura 42. Pasos en la interpolación de los datos.	49
Figura 43. Pasos métodos Welch & Autorregresivo.....	50
Figura 44. Pasos para encontrar las características en el dominio del tiempo.....	50
Figura 45. Pasos para obtener dominio tiempo-frecuencia.....	52
Figura 46. Pasos para obtener las características en el dominio de la frecuencia.....	53
Figura 47. Método de selección de características.	53
Figura 48. Ventana de exportación de las características seleccionadas.	54
Figura 49. Pasos para importar las características a clasification learner.....	55
Figura 50 Entrenamiento de modelos con su Precisión.....	55

Índice de anexos:

Anexo 1. Datos de los sensores de los aerogeneradores.....	62
Anexo 2. Código preprocesamiento de datos.	63
Anexo 3. Designación de código (0-1) Fault Query para el convertidor.....	65
Anexo 4 Código para procesamiento & indicadores de condición.....	66
Anexo 5 Selección de características.....	72
Anexo 6 Código de entrenamiento de modelos.....	73
Anexo 7. Código de importe del archivo (complete_dataset.csv).	76
Anexo 8. Variables de temperatura seleccionadas del convertidor.	78
Anexo 9 Preprocesamiento y preparación de la tabla para DFD.....	79
Anexo 10. Certificado de traducción del resumen.....	81

Simbología:

DFD = Diagnostic feature designer.

CSV = Valores separados por comas .

MAD = Desviación absoluta mediana .

A = Conjunto de datos.

ke = Función de escalada.

y = Valor a encontrar en la interpolación.

y_0 = Valor inicial en las ordenadas.

y_1 = Valor superior en las ordenadas.

x = Valor dado en la interpolación.

x_0 = Valor inicial en las abscisas.

x_1 = Valor mayor de las abscisas.

Q_k = Posición del cuartil.

Q_3 = Cuartil tres.

Q_1 = Cuartil uno.

k_t = escala para la amplitud del rango intercuartílico.

x_i = Cada uno de los valores del conjunto.

u = Media.

σ = Desviación estándar

$P_{\text{Welch}}(f)$ = Densidad espectral de potencia (Welch).

N = tamaño de la ventana para dividir la señal en segmentos.

M = Número total de segmentos superpuestos.

Kur = Curtosis.

Sk = Coeficiente de asimetría (skewness).

x_p = Valor pico o máximo.

x_{IF} = Valor de impulso.

$X_i(f)$ = Transformada d Fourier.

X_t = Variable en el tiempo del modelo autorregresivo.

c = Constante modelo autorregresivo.

$\phi_1, \phi_2, \dots, \phi_p$ = Coeficientes autorregresivos.

ε_t = Factor estocástico en el tiempo t .

n = Número de elementos de la muestra.

\bar{x} = Pomedio de los datos.

ε_t = Factor estocástico en el tiempo t.
 FI = Factor de impulso (curtosis espectral).
 IMF = Funciones de modo intrínseco.
 EMD = Descomposición de modo empírico.
 FA = Factor de aclaramiento (curtosis espectral).
 t_{test} = Evaluación de la media de una muestra.
 TSA = Promedio sincrónico en el tiempo.
 $XLSX$ = Hoja de cálculo de excel almacenada en formato.
 DME = Descomposición de modo empírico.
 FMI = Hoja de cálculo de excel almacenada en formato.
 RMS = Valor cuadrático medio.
 $SINAD$ = Señal a Ruido y Distorsión.
 SNR = Señal de la Señal Ruido.
 $P(x|y = k)$ = Función discriminante y cuadrático.
 $g(u)$ = Regresión logístico binario.
 $P(y| x_1, x_2, \dots x_n)$ = Gaussiano Naive Bayes.
 $f(x)$ = Lineal y cuadrático SVM.
 $k(x_i, x_j)$ = Cúbica SVM.
 $P(y = 1|x)$ = Probabilidad de la observación (clasificadores lineales).

1. Título

Exploración y aplicación de "*Diagnostic Feature Designer*" para desarrollar modelos de diagnóstico de fallos en plantas eólicas.

2. Resumen

Este trabajo de titulación explora la aplicación "*Diagnostic Feature Designer*" (DFD) de MATLAB, con el fin de comprender sus funcionalidades y que con esto se pueda crear modelos de diagnóstico de fallos en aerogeneradores. Para esto, se utilizó la documentación de MathWorks como guía en el desarrollo de los scripts, contemplando la parte del preprocesamiento, procesamiento de datos y características en el dominio del tiempo, tiempo-frecuencia y la frecuencia. Para encontrar estos modelos de diagnóstico se seleccionó los datos de temperatura del convertidor, proporcionados por la Corporación Eléctrica del Ecuador – CELEC. Se eligió estos datos porque representan el 33% de fallos en aerogeneradores. Con estas temperaturas, procesadas en todos los dominios, se encontraron 5 características importantes seleccionadas mediante el método supervisado prueba de dos clases (T-test): las características, promedio(mean), valor pico (peak value), valor cuadrático medio (RMS), valor pico con funciones de modo intrínseco y banda de potencia (band power), fueron utilizadas por el orden de importancia del método T-test. Estas características o indicadores de condición se entrenaron en la aplicación Classification Learner, dando al modelo de árbol, como el que mayor precisión entregó en el entrenamiento, con una precisión del 97.27%. Este resultado de la aplicación DFD, indica la importancia de la utilización de la herramienta para la obtención de los modelos de diagnóstico y los indicadores de condición generados en este caso en particular para aerogeneradores.

Palabras claves: DFD, MathWorks, T-test, método supervisado, Classification Learner, indicadores de condición.

Abstract

This degree work explores the application "Diagnostic Feature Designer" (DFD) of MATLAB in order to understand its functionalities and, consequently, create models for diagnosing failures in wind turbines. For this purpose, the MathWorks documentation was used as a guide in the development of the scripts, covering the preprocessing, data processing, and feature extraction in the time, time-frequency, and frequency domains. To find these diagnostic models, the temperature data of the converter provided by the Electric Corporation of Ecuador (CELEC) was selected. These data were chosen because they represent 33% of failures in wind turbines. With these temperatures, processed in all domains, 5 important characteristics were identified using the supervised two-class test method (T-test): mean, peak value, root mean square (RMS), peak value with intrinsic mode functions, and band power. These features were used according to the order of importance determined by the T-test method. These characteristics or condition indicators were trained in the Classification Learner application, with the tree model delivering the highest accuracy during training, achieving an accuracy of 97.97%. This result from the DFD application underscores the importance of using this tool to obtain diagnostic models and condition indicators, particularly in the case of wind turbines.

Keywords: DFD, MathWorks, T-test, supervised method, Classification Learner, condition indicators.

3. Introducción

Este estudio se centra en la exploración y aplicación del DFD, una herramienta fundamental integrada en el entorno de desarrollo de MATLAB. Está específicamente diseñada para facilitar la creación de modelos de diagnóstico dirigidos a la detección de fallos en maquinaria, mantenimiento, ingeniería y en disciplinas que se necesite predecir anomalías, como en este caso específico de las plantas eólicas. A través de la importación y procesamiento de datos, esta plataforma de software emplea técnicas estadísticas para el filtrado y la visualización de señales. Un aspecto sobresaliente de este enfoque radica en su capacidad para extraer indicadores de condición mediante el análisis de señales en diversos dominios, como el tiempo, la frecuencia y el tiempo-frecuencia. La aplicación permite la selección de características e indicadores de condición de manera más práctica, lo que facilita la configuración de modelos apropiados para el entrenamiento y la detección de fallos en aerogeneradores. El flujo de trabajo comienza con la preparación e importación de los datos, seguida de la visualización de las señales seleccionadas. Luego, se diseñan las características o indicadores de condición y se clasifican estas características para el entrenamiento de modelos. Esto es aplicable al mantenimiento de una variedad de máquinas, incluyendo máquinas rotativas, sistemas eléctricos, máquinas con señales térmicas, automóviles y aerogeneradores (MathWorks, 2019a).

La aplicación DFD es utilizada incluso con gemelos digitales para la detección de fallos en maquinaria industrial utilizando códigos de fallos como por ejemplo 0 y 1, obteniendo una matriz de confusión o correlación la cual es llevada a la ventana de trabajo de Matlab para su análisis y entrenamiento (Singh et al., 2023). También es utilizada en electrónica de potencia en la degradación en los convertidores que de igual forma extrae automáticamente los indicadores de condición, donde es el fuerte de esta aplicación (Zhao et al., 2021). Además, se utiliza en el monitoreo de temperatura de las placas PCBs, seleccionando las características más apropiadas y comparando varios algoritmos de aprendizaje automático para determinar el método más eficaz de localización de puntos calientes (Yule et al., 2024). Pero la información más importante está directamente en MathWorks, que es donde más se utilizado para este estudio de esta aplicación.

Como se conoce que las máquinas presentan fallos en el tiempo y con la necesidad de predecir las anomalías en los aerogeneradores, acompañado de la complejidad en el análisis de datos, y el desafío de trabajar con conjuntos de datos, es importante trabajar rápidamente en un algoritmo que encuentre modelos de diagnóstico, pero como esto puede ser complejo y confuso sobre todo en la programación, entonces es útil un programa como DFD donde su interfaz sea

clara al momento de manejar datos para el mantenimiento predictivo, por esto representa una oportunidad para mejorar nuestro análisis y procesamiento de datos. Esta herramienta permite realizar un análisis más profundo de las funcionalidades, explorando cada una de las técnicas de procesamiento de datos disponibles. Además, la retroalimentación obtenida durante este proceso es fundamental para la búsqueda y desarrollo de un modelo de diagnóstico adecuado para los aerogeneradores.

Este trabajo pretende mejorar la comprensión de los procedimientos necesarios para llevar a cabo el mantenimiento predictivo en aerogeneradores. Destaca la utilidad que ofrece la aplicación para procesar los datos y encontrar modelos, la cual puede ser aprovechada por ingenieros, estudiantes y la comunidad científica. Este estudio puede considerarse como una contribución al desarrollo de otros temas de investigación dentro del sector, ofreciendo beneficios significativos para el avance y la innovación en el campo del mantenimiento predictivo y el análisis de datos.

El tema abarca una amplia gama de aplicaciones en ingeniería, particularmente en el ámbito del mantenimiento predictivo. El avance constante de la inteligencia artificial es notable en este campo. Por consiguiente, esta investigación tiene el potencial de beneficiar a numerosas personas al facilitar la predicción de fallos en aerogeneradores y otras máquinas.

El objetivo general que cubre toda esta investigación es “Explorar y aplicar las funcionalidades de DIAGNOSTIC FEATURE DESIGNER para desarrollar modelos de diagnóstico de fallos, que permitan mejorar la efectividad del mantenimiento predictivo en plantas eólicas.”

Los objetivos específicos son los siguientes:

- Caracterizar y documentar las funcionalidades de DIAGNOSTIC FEATURE DESIGNER en el contexto del diagnóstico de fallos en plantas eólicas.
- Desarrollar modelos de diagnóstico de fallos en plantas eólicas utilizando DIAGNOSTIC FEATURE DESIGNER.
- Generar guías de uso e interpretación de resultados para DIAGNOSTIC FEATURE DESIGNER en el contexto de plantas eólicas.

Sin embargo, las limitaciones pueden surgir debido a la disponibilidad de datos y a los recursos computacionales necesarios para enfrentar el desafío del análisis de datos.

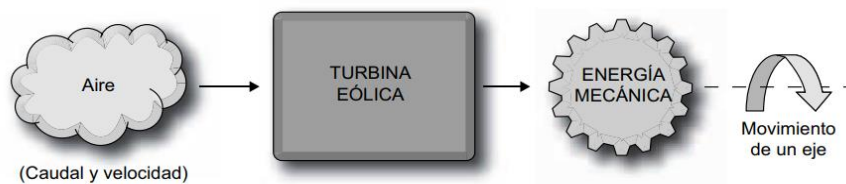
4. Marco teórico

4.1 Aerogeneradores

La energía eólica, es la energía cinética del viento. Para el aprovechamiento de dicha energía se han desarrollado a lo largo de la historia diferentes sistemas tecnológicos. Sin embargo, en la actualidad, la aplicación más generalizada de la energía contenida en el viento es la producción de electricidad mediante aerogeneradores que, aprovechando el conocimiento de múltiples disciplinas, se diseñan, construyen y operan con las tecnologías más avanzadas y se conectan, frecuentemente configurando los denominados parques eólicos, a algún tipo de red eléctrica (González et al., 2009, p. 334).

La transformación de esta energía se lo hace mediante turbinas eólicas, solo que aquí el fluido que impacta sobre las palas del elemento giratorio es el aire en movimiento, el viento, como se muestra en la **Figura 1**, transformando la energía en movimiento de un eje en energía eléctrica.

Figura 1. Transformación de la energía en turbinas eólicas.

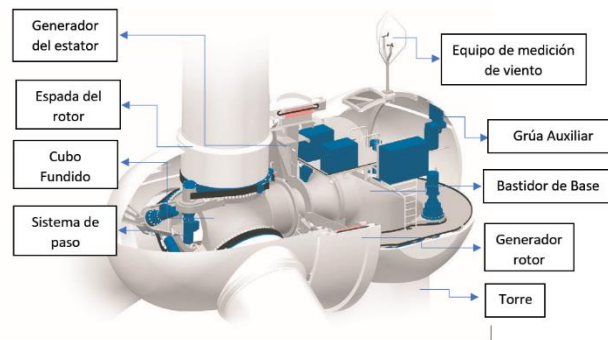


Nota. Adaptado de esquema conceptual de una turbina eólica, Centrales de Energías Renovable, 2009.

“Por lo general el sistema generador adopta un esquema de diseño general que involucra el eje horizontal, tres palas hélice, contra el viento, ajuste de paso variable, accionador directo con imán permanente interconectado a un generador síncrono” (Goldwind, 2008, p. 1).

En la **Figura 2**, se observa algunas de las partes que conforman a uno de los aerogeneradores que operan en la Central Eólica Villonaco.

Figura 2. Partes de un aerogenerador.



Nota. Tomado del manual de funcionamiento del sistema generador de turbinas eólicas serie Goldwind de 1500 kW, Goldwing, 2008

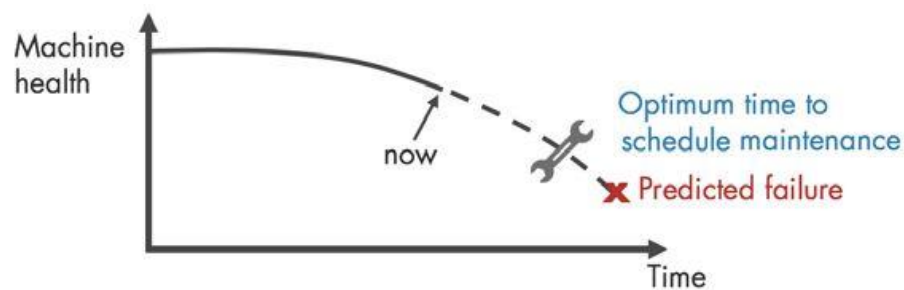
4.2 Mantenimiento predictivo

Según lo que menciona Mathworks (2011, p. 5) en uno de sus documentos:

El mantenimiento predictivo le permite estimar el tiempo de falla de una máquina. Conocer el tiempo de falla previsto le ayuda a encontrar el momento óptimo para programar el mantenimiento de su equipo. El mantenimiento predictivo no solo predice una falla futura, sino que también identifica problemas complejos en la máquina, y lo ayuda a identificar qué piezas deben repararse.

En la **Figura 3** se puede observar una gráfica que muestra en tiempo donde se presume dicho mantenimiento o cambio de componente.

Figura 3. Gráfica mantenimiento predictivo.



Nota. Tiempo de predicción del mantenimiento. Tomado de documento mantenimiento predictivo con Matlab Mathworks, 2019.

4.3 Aprendizaje Automático (Machine Learning)

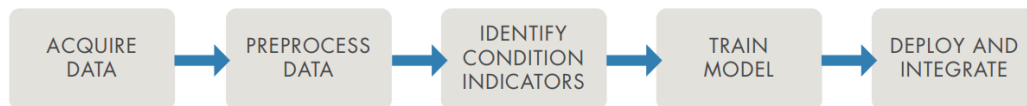
Mathworks (2019, p. 7) menciona que:

Los algoritmos se comienzan con datos que describen su sistema de una variedad de condiciones saludables y defectuosas, los datos brutos se preprocesan para darles una forma de la que se puedan extraer indicadores de condiciones, son características que ayudan a distinguir las condiciones sanas de las defectuosas, con esto se puede utilizar las características extraídas para entrenar a una máquina de aprendizaje que pueda:

- Detectar anomalías.
- Clasificar los distintos tipos de averías.
- Estimar la vida útil restante (del inglés remaining useful life, RUL) de su máquina.

En la Figura 4, se puede observar el proceso de aprendizaje automático, utilizado comúnmente en el análisis de datos.

Figura 4. Proceso de aprendizaje automático.



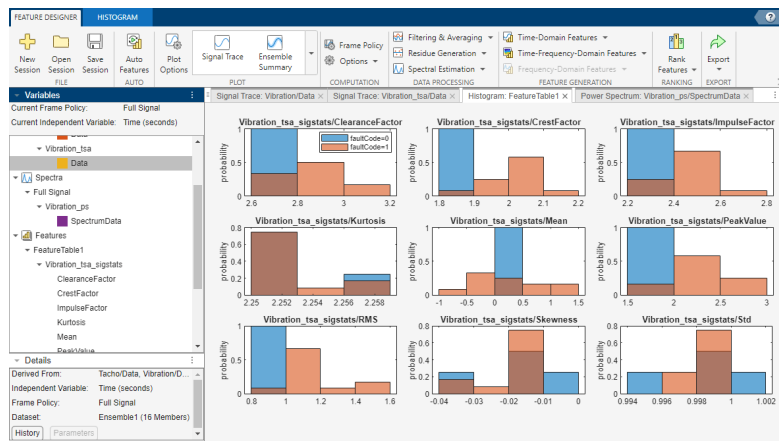
Nota. Pasos que se debe seguir en el aprendizaje automático. Tomado de proceso de aprendizaje automático, MathWorks, 2019

4.4 DIAGNOSTIC FEATURE DESIGNER

Con DFD se puede diseñar funciones del flujo de trabajo de mantenimiento predictivo mediante una interfaz gráfica multifunción. Esto se hace utilizando las variables que se tiene o que el usuario considere para el análisis, las cuales, en realidad son las mismas características diseñadas para el modelo a entrenar. Al diseñar y comparar características de forma interactiva se determina qué características son mejores para seleccionar entre datos de sistemas nominales y de sistemas defectuosos. Las funciones más efectivas se convierten en indicadores de condición para el diagnóstico y pronóstico de fallas.

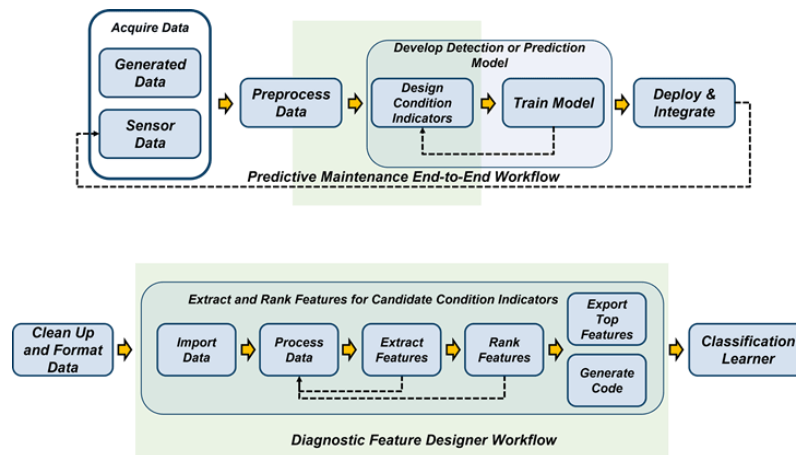
Se puede apreciar la interfaz de trabajo de la aplicación en la **Figura 5.**

Figura 5. Interfaz DFD.



El flujo de trabajo de DFD tiene relación con los pasos de trabajo en el mantenimiento predictivo según señala (MathWorks, 2023d). Estos pasos de la aplicación se pueden ver conectados en la parte de preprocesamiento y en la búsqueda de indicadores de condición. En la **Figura 6** se puede observar cada una de los pasos del flujo de trabajo que ofrece la aplicación.

Figura 6. Flujo de trabajo de DFD.



Nota. Flujo de trabajo de DFD. Tomado de (MathWorks, 2023d)

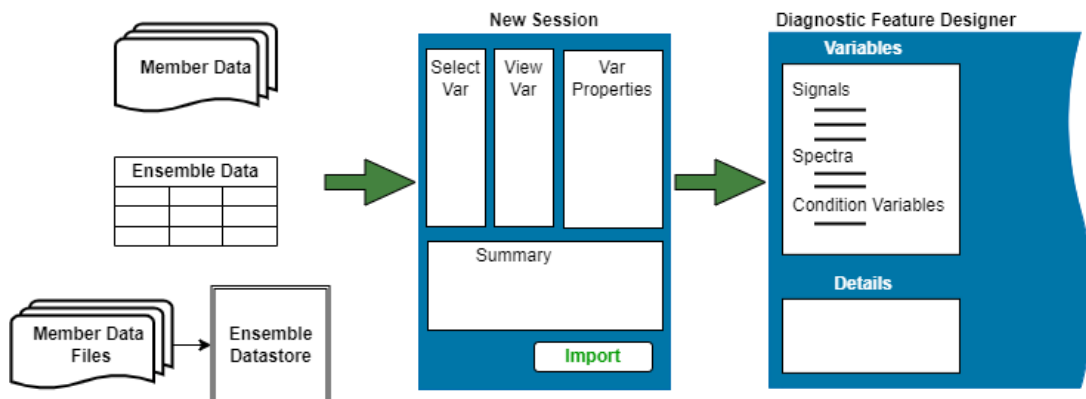
4.4.1 Importar datos

Los datos pueden provenir de mediciones en sistemas que utilizan sensores como acelerómetros, manómetros, termómetros, altímetros, voltímetros y tacómetros. La unidad principal para organizar y gestionar conjuntos de datos multifacéticos en la aplicación es el conjunto de datos. Un conjunto, es una colección de conjuntos de datos, medidos o simulados en un sistema en condiciones variables. Dentro de estas matrices o miembros contienen las variables con condición, por ejemplo, fallos de alguna máquina o algún sistema en específico. Al combinar los datos de sus miembros en un único conjunto de datos. En este conjunto de

datos, cada variable es una señal o modelo colectivo que contiene todos los valores de los miembros individuales.

Antes de importar, se debe limpiar o llenar los datos faltantes y atípicos. Este es un preprocesamiento básico donde está involucrada la organización de los datos a trabajar ya sea en un conjunto matricial o datos ensamblados como lo menciona (MathWorks, 2019f). En la **Figura 7** se pueden ver los pasos generales para importar los datos a DFD.

Figura 7. Pasos generales para importar datos a DFD.



Fuente: (MathWorks, 2019f).

4.4.2 Preprocesar los datos

Antes de todo, los datos deben ser examinados para detectar posibles anomalías que pudieran llevar a conclusiones erróneas, el uso apropiado de técnicas de análisis de datos puede mitigar los inconvenientes más comunes como: datos faltantes, valores atípicos, colinealidad, etc (Martínez, 2007, p. 17).

Los datos de la vida real suelen verse afectados por registros omitidos o valores atípicos, causados por fallos de comunicación o defectos de los sensores. Es necesario un filtro preliminar de lecturas de los datos para reducir las posibilidades de generar falsas alarmas.

4.4.2.1 Selección, limpieza y transformación de Datos.

Según MathWorks (2018, p. 10), los datos rara vez son perfectos, suelen tener valores faltantes, acompañado de datos atípicos y cierto nivel de ruido que se debe eliminar para obtener una imagen realista del comportamiento normal de los datos. Como los datos proceden de distintas fuentes, también será necesario combinarlos. Si se eliminan las anomalías, plantéese si desea reemplazarlas con valores aproximados o trabajar con un conjunto de datos más pequeño.

4.4.2.2 Variables con valores faltantes

Uno de los problemas que se presenta regularmente cuando se trabaja con grandes cantidades de datos, es la presencia de valores faltantes en algunas de las variables. Existen varias técnicas para la estimación de esos valores, como:

Sustituir los valores faltantes por algún estimador como la media, la mediana, aproximaciones, interpolaciones o completar con datos semejantes a los datos existentes.

Eliminar las observaciones que presenten valores faltantes.

En la **Tabla 1**, se presenta algunos de los métodos o técnicas utilizados para encontrar y llenar los datos faltantes.

Tabla 1. Métodos para análisis de datos faltantes.

Método	Descripción
Interpolación lineal	<p>Interpolación lineal de valores cercanos, valores faltantes. Para la interpolación lineal y poder llenar los datos faltantes se utiliza la Ecuación 1.</p> $y = \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) + y_0 \quad \text{Ecuación 1}$ <p>Donde: y es el valor a encontrar, y_1 representa al valor superior en las ordenadas, y_0 es el valor menor en las ordenadas, x_1 es el valor mayor en las abscisas, x es el valor en las ordenadas para el cual queremos interpolar el valor en las abscisas y x_0 es el valor menor en las abscisas.</p>
Valor constante	Valor escalar especificado, que es 0 el predeterminado
Valor previo	Valor anterior no faltante.
Valor siguiente	Siguiente valor no faltante.
Valor más próximo	Valor no perdido más cercano según lo definido por el eje x
Interpolación de spline	Interpolación spline cúbica por partes
Interpolación cúbica que preserva la forma.	Interpolación spline cúbica por partes que conserva la forma
Interpolación cúbica akima modificada	Interpolación cúbica de Hermite de Akima modificada
K-vecinos más cercanos	Media de los vecinos más cercanos definidos por una función de distancia.
Función personalizada	Método de relleno personalizado, especificado como una función local o un identificador de función

Fuente: (MathWorks, 2019h).

4.4.2.3 Variables con valores atípicos

Para Martínez (2007, p. 19) menciona que los datos recolectados, en las formas cualesquiera que se hayan obtenido los datos, muchas veces suelen presentarse observaciones

que al parecer no guardan ningún tipo de relación con respecto al resto de las observaciones. Son los llamados valores atípicos o “outliers”. Una de las técnicas gráficas más populares para la detección de “outliers” son los Diagramas de Cajas y Bigotes o “boxplot” los cuales permiten detectar aquellas observaciones sospechosamente alejadas de la masa de datos.

En la **Tabla 2** se puede apreciar algunos de los métodos que se han utilizado para encontrar valores atípicos en el preprocesamiento de datos.

Tabla 2. Métodos para encontrar valores atípicos.

Método	Descripción
Mediana móvil	Los valores atípicos se definen como elementos que superan el umbral especificado de desviaciones absolutas de la mediana (del inglés median absolute deviations, MAD) de la mediana local durante una ventana específica. El umbral predeterminado es 3.
Mediana	Los valores atípicos se definen como elementos que superan el umbral especificado de MAD. El umbral predeterminado es 3, la Ecuación 2 representa el MAD. $MAD = ke * mediana(A - mediana(A))$ Ecuación 2 Donde: A es el conjunto de los datos y ke es la función escalada $ke = -\frac{1}{\sqrt{2} \sqrt{-\ln(1-\frac{3}{2})}}$
Media	Los valores atípicos se definen como elementos que superan el umbral especificado de desviaciones estándar de la media. El umbral por defecto es 3. Este método es más rápido, pero menos robusto que Mediana.
Cuartiles	Los valores atípicos se definen como elementos que superan el umbral especificado de rangos intercuartílicos por encima del cuartil superior (75 por ciento) o por debajo del cuartil inferior (25 por ciento). El umbral predeterminado es 1.5. Este método es útil cuando los datos de entrada no se distribuyen normalmente. Para encontrar los cuartiles se toma en cuenta si el número de datos es impar o par, entonces la posición de cada cuartil se calcula con la Ecuación 3 : $Q_k = \frac{k(n+1)}{4}$ Ecuación 3 Donde k , es la posición del cuartil, n es el número de datos, para número de datos pares se elimina el uno en la ecuación. Para encontrar los datos atípicos se encuentra los umbrales superior e inferior, y se calcula con la Ecuación 4 : $Umbral_superior = Q_3 + k_t(Q_3 - Q_1)$ Ecuación 4 Y la $Umbral_inferior = Q_1 - k_t(Q_3 - Q_1)$ Ecuación 5 donde Q_3 es el tercer cuartil, Q_1 es el primer cuartil y k_t representa la escala para la amplitud del rango intercuartílico.

Fuente: (MathWorks, 2019h).

4.4.2.4 Procesar datos

“El programa tiene varias opciones para procesar los datos en nuevas señales, permitiendo encontrar nuevas formas de procesamiento. Estas señales nuevas se pueden utilizar como entradas para otras opciones de procesamiento o para la generación de funciones. La mayoría de las opciones de procesamiento operan en cada miembro del conjunto” (MathWorks, 2019b).

En la **Tabla 3**, se puede observar las técnicas que comúnmente se utilizan en la aplicación para el análisis de diferentes señales.

Tabla 3. Técnicas de procesamientos presentes en DFD.

Funcionalidades	Técnica de Procesamiento	Descripción
	Promedio sincrónico en el tiempo	<p>Promedio sincrónico en el tiempo (del inglés time-synchronous averaging, TSA) se utiliza para máquinas rotativas, donde se emplea señales de tacómetro o las velocidades de rotación.</p> <p>Media.</p> $u = \frac{\sum_{i=1}^n x_i}{n} \quad \text{Ecuación 6}$ <p>Donde: n es el número total de valores en el conjunto de datos, x_i es cada uno de los valores de conjunto.</p>
	Estadísticas del conjunto	<p>Mínimo.</p> <p>Máximo.</p> <p>Desviación estándar.</p>
Filtrado y promedio		$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N x_i - u ^2} \quad \text{Ecuación 7}$ <p>Donde: u es la media de la muestra, x_i es cada valor de la muestra, N es el tamaño de la muestra.</p>
	Eliminar armónicos	<p>Utiliza la frecuencia fundamental, ancho de banda de muesca, filtros de paso bajo y paso alto.</p> <p>Se filtra una señal TSA para generar señales adicionales que aíslan la respuesta de componentes específicos dentro de un sistema armónicamente interrelacionado. Estas señales aisladas luego se pueden utilizar para funciones de maquinaria giratoria que detectan fallas en ubicaciones y componentes físicos específicos.</p>
	Filtrar señales TSA	<p>Utilice el filtrado de señales TSA solo si las velocidades del eje son constantes en el tiempo.</p>

Funcionalidades	Técnica de Procesamiento	Descripción
Generación de Residuos	Extraer envolventes de señal	Cuando importa datos de señales de alto ancho de banda a DFD, puede extraer envolventes de señales de bajo ancho de banda de los datos de la señal para aislar la modulación de vibración de baja frecuencia cerca de las resonancias de la máquina. Esta señal de baja frecuencia puede resultar útil para detectar fallas en maquinaria giratoria.
	Interpolación	Métodos de interpolación como: lineal, vecino, valor anterior, valor siguiente, cúbica, etc.
	Restar referencia. Procesamiento de series temporales y funciones de series temporales.	Genere residuos restando una señal de referencia cuando desee centrar su análisis en la variación de la señal eliminando el comportamiento normal a nivel de conjunto, como la media del conjunto. Se obtienen series temporales a partir de datos de señales, lo que implica que las propiedades estadísticas como la media y la varianza son constantes en el tiempo, y que la serie no muestra tendencias claras ni patrones estacionales.
Estimación Espectral	Espectro de potencia Espectro de pedidos	<p>Método de Welch:</p> $P_{\text{Welch}}(f) = \frac{1}{N \cdot M} \left \sum_{i=1}^M X_i(f) \right ^2 \quad \text{Ecuación 8}$ <p>Donde: $P_{\text{Welch}}(f)$ es la densidad espectral de potencia en la frecuencia f, N es el tamaño de la ventana para dividir la señal en segmentos, M es el número total de segmentos superpuestos, $X_i(f)$ es la transformada de Fourier de la k-ésima ventana aplicada al segmento de la señal.</p> <p>Modelo autorregresivo:</p> $X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + \varepsilon_t \quad \text{Ecuación 9}$ <p>Donde: X_t es la variable en el tiempo t, c es una constante, $\phi_1, \phi_2, \dots, \phi_p$ son los coeficientes autorregresivos que representan la influencia de los valores anteriores en el valor actual, ε_t error estocástico en el tiempo t.</p>

Fuente: (MathWorks, 2019h).

4.4.3 Indicadores de condición

Según MathWorks (2019, p. 2), un paso clave en el desarrollo del algoritmo de mantenimiento predictivo es: identificar indicadores de condición características en los datos de su sistema cuyo comportamiento cambia de una manera predecible a medida que el sistema se degrada. Los indicadores de estado le ayudan a distinguir el funcionamiento correcto del defectuoso. Se puede obtener los indicadores de condición mediante el uso de características de dominio de tiempo, frecuencia y tiempo-frecuencia.

4.4.3.1 Características en el dominio del tiempo

Se puede crear características en el dominio del tiempo de una señal, características que pueden ser de series temporales, características basadas en modelos, características de máquinas giratorias, características no lineales, y en DFD se puede trabajar con funciones personalizadas por parte del usuario para mejorar los indicadores de condición (MathWorks, 2019b).

En la **Tabla 4** se muestran las funcionalidades que utiliza DFD para obtener características en el dominio del tiempo.

Tabla 4. Funciones de DFD para obtener características en el dominio del tiempo.

	Funciones	Descripción
		<p>Media.</p> <p>Para la media se utiliza la Ecuación 6.</p> <p>Valor cuadrático medio (RMS):</p> $\text{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N [x(i)]^2}$ <p style="text-align: right;">Ecuación 10</p> <p>Donde: RMS es el valor eficaz de la señal, N es el número de muestras en la señal discreta, $x(i)$ son las muestras de la señal.</p> <p>Desviación estándar:</p> <p>Para la desviación estándar se utiliza la Ecuación 7.</p> <p>Curtosis:</p>
Funciones en el dominio del tiempo	Características de la señal.	$\text{Kur} = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^4}{\left(\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2\right)^2} - 3$ <p style="text-align: right;">Ecuación 11</p>
		<p>Donde: n es el número de elementos de la muestra, x_i elementos individuales del conjunto, \bar{x} es el promedio de los datos, 3 es para el ajuste de la distribución normal y pueda dar 0.</p> <p>Coefficiente de asimetría (skewness):</p> $\text{Sk} = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^3}{\left(\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2\right)^{3/2}}$ <p style="text-align: right;">Ecuación 12</p> <p>Donde: n es el número de elementos de la muestra, x_i elementos individuales del conjunto, \bar{x} es el promedio de los datos.</p> <p>Valor pico o valor máximo:</p> $x_p = \max x_i $ <p style="text-align: right;">Ecuación 13</p>

Donde: x_i representa a elementos individuales del conjunto.

Valor de impulso:

$$x_{IF} = \frac{x_p}{RMS} \quad \text{Ecuación 14}$$

Donde: x_p es el valor pico o máximo y RMS es el valor cuadrático medio.

Funciones basadas en modelos.

Los modelos matemáticos de señales en el dominio del tiempo representan la dinámica subyacente de los mecanismos que generaron las señales. Los modelos autorregresivos son particularmente útiles para aplicaciones como maquinaria rotativa, que producen picos agudos en sus espectros de potencia. Las características extraídas de estos modelos pueden reflejar diferencias de comportamiento que surgen de condiciones saludables o defectuosas.

Fuente: (MathWorks, 2019b).

4.4.3.2 Características en el dominio de tiempo-frecuencia

“Las características de tiempo-frecuencia caracterizan señales cuyas frecuencias cambian en el tiempo (es decir, no son estacionarias). Estas señales pueden surgir de maquinaria con hardware degradado o defectuoso” (MathWorks, 2019b).

Se puede ver la **Tabla 5** con las funcionalidades que tiene la aplicación.

Tabla 5. Funcionalidades de dominio tiempo-frecuencia.

Funciones	Características
Funciones de espectrograma	Curtosis espectral
	<ul style="list-style-type: none"> Factor de cresta:
	$FC = \frac{x_p}{u} \quad \text{Ecuación 15}$
	Donde: x_p representa el valor pico o el valor máximo, u es el valor cuadrático medio.
	<ul style="list-style-type: none"> Factor de impulso:
El valor de impulso utiliza la Ecuación 14 para los cálculos.	
<ul style="list-style-type: none"> Factor de aclaramiento: 	
$FA = \frac{u}{RMS} \quad \text{Ecuación 16}$	
<ul style="list-style-type: none"> Valor pico: 	
El valor pico utiliza la Ecuación 13 para los cálculos.	
	Entropía Espectral

Funciones basadas en la descomposición del modo empírico (del inglés empirical mode decomposition, EMD).	Se utiliza las funciones de modo intrínseco (del inglés intrinsic mode functions, IMF). Se utiliza EMD para descomponer y simplificar señales complicadas en un número finito de funciones de modo intrínseco, se utilizan las ecuaciones de características espectral.
--	---

Fuente: (MathWorks, 2019b).

4.4.3.3 Características en el dominio de la frecuencia

Las características en el dominio de la frecuencia, se aplican a cualquier espectro, se utiliza más a máquinas rotativas; engranajes, rodamientos y todo lo que tenga que ver con la frecuencia. Algunas de las funcionalidades para el análisis de estas señales, en la que se puede ver en la herramienta son las que aparecen en la **Tabla 6**:

Tabla 6. Características en el dominio de la frecuencia.

Funciones	Características
Características espectrales.	Picos Espectrales: Se utiliza para encontrar los picos que se puede encontrar en las señales: Amplitud de pico. Frecuencia de pico. Poder de Banda.
Características espectrales basadas en bandas de falla.	Etas características son muy utilizadas en máquinas rotativas, como, por ejemplo: rodamientos, engranajes y también se puede personalizar las funciones de fallas con diferentes configuraciones.

Fuente: (MathWorks, 2019b).

4.4.3.4 Selección de características

La extracción de características es lo primero que se suele hacer en la predicción. Sin embargo, no todas las características son utilizables. Utilizar muchas características requiere de más recursos computacionales y también se tiene el problema del sobreajuste. Lo importante es seleccionar las más importantes para su entrenamiento. Se puede elegir el método de clasificación de las características, estos son: clasificación supervisada y no supervisada, generalmente se han utilizado lo métodos de dos clases y los de multiclase, en la **Tabla 7** se observa los métodos con sus funcionalidades.

Tabla 7. Selección de características.

MÉTODO	FUNCIONALIDADES
Supervisado	Métodos de dos clases: T-Test: Evalúa si la media de una muestra es significativamente diferente de un valor conocido o hipotético, ver la Ecuación 17 .

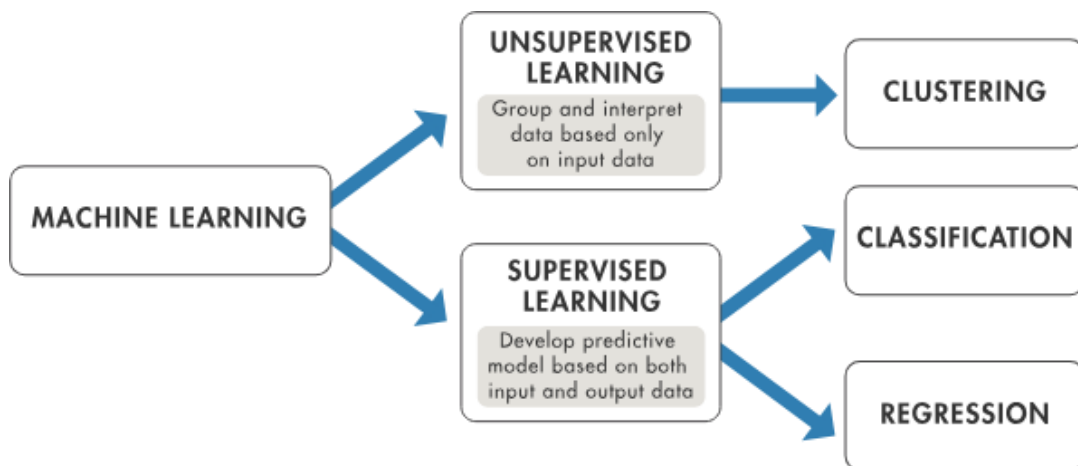
MÉTODO	FUNCIONALIDADES
	$t_{\text{test}} = \frac{u - \mu_0}{\sigma/\sqrt{n}}$ <p style="text-align: right;">Ecuación 17</p> <p>Donde; \bar{X} es la media de la muestra, μ_0 es el valor hipotético, s es la desviación estándar de la muestra, n es el tamaño de la muestra.</p> <ul style="list-style-type: none"> • ROC • Bhattacharyya • Entropía • Wilcoxon <p>Métodos multiclase:</p> <p>One-way ANOVA</p> <p>Kruskall Wallis</p> <p>Puntuación Laplaciano</p> <p>Varianza</p>
No supervisado	

Fuente: (MathWorks, 2019b).

4.4.4 Entrenar modelos

En la predicción, se ha utilizado el aprendizaje supervisado el cual utiliza dos técnicas: la de clasificación y de regresión. La elección de esta técnica depende de la situación particular, es decir una respuesta con dos o más categorías, se utiliza el entrenamiento por clasificación. Y si por el contrario se quiere predecir un cambio en la variable continua, se suele utilizar la regresión. En la **Figura 8** se puede ver el diagrama de elección para el entrenamiento.

Figura 8. Técnicas de aprendizaje automático supervisado y no supervisado.

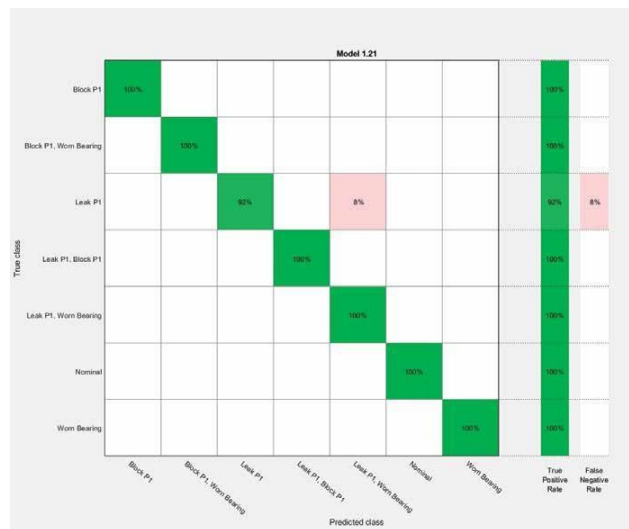


Fuente: (MathWorks, 2019h).

Cuando se tiene los indicadores de estado, se puede entrenar un modelo de aprendizaje automático con las características extraídas y comprobar la precisión del modelo entrenado con una matriz de confusión (MathWorks, 2019, p. 22).

La **Figura 9** muestra los resultados de uno de los clasificadores de mejor rendimiento con las características extraídas.

Figura 9. Matriz de confusión.



Nota. Precisión del modelo en una matriz de confusión. Tomado de mantenimiento predictivo: indicadores de condición Matlab, MathWorks, 2019.

En la **Tabla 8** se puede ver los tipos de modelos que se utiliza para el entrenamiento en aprendizaje automático, junto con las ecuaciones.

Tabla 8. Tipos de modelos en la predicción por clasificación.

MODELO	DESCRIPCIÓN
Árboles de decisión	<p>Se basa en condiciones lógicas, con sus nodos y ramas, puede ser: árbol fino, medio, y mediano, ejemplo entre dos clases:</p> <p>Si $X_1 \leq 5$ y $X_2 \leq 3$, entonces Clase = A</p> <p>Si $X_1 > 5$ y $X_2 \leq 3$, entonces Clase = B</p> <p>Utilizada para encontrar la combinación lineal de características que mejor separan dos o más clases. Las ecuaciones utilizadas son las siguientes:</p> <p>Discriminante lineal:</p>
Análisis discriminante	$P(x y = k) = \frac{1}{\sqrt{(2\pi)^D \Sigma }} e^{-\frac{1}{2}(x-u)^T \Sigma^{-1}(x-u)}$ <p style="text-align: right;">Ecuación 18</p> <p>donde $P(x y = k)$ es una función discriminante, x es el vector de características de entrada, D es el número de características, u es el vector de todas las clases combinadas, Σ es la matriz de covarianza común para todas las clases.</p> <p>Discriminante cuadrático:</p>

MODELO	DESCRIPCIÓN
Clasificadores de regresión logística	$P(x y = k) = \frac{1}{\sqrt{(2\pi)^{D/2} \Sigma_k ^{1/2}}} e^{-\frac{1}{2}(x-u)^T \Sigma_k^{-1}(x-u_k)}$ <p style="text-align: right;">Ecuación 19</p> <p>donde $P(x y = k)$ es una función discriminante, x es el vector de variables predictoras, A es la matriz de coeficientes cuadráticos, b es la constante de intersección.</p> <p>Regresión logístico binario GLM:</p> <p>Se trata de un modelo simple, lineal binario generalizado, que describe la respuesta media en función de la combinación lineal de los predictores.</p> $g(u) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k$ <p style="text-align: right;">Ecuación 20</p> <p>Donde: $g(u)$ es la función enlace, u es la media de la distribución de probabilidad condicional, $\beta_0, \beta_1, \beta_k$ son los coeficientes a estimar, x_1, x_2, x_k son las variables independientes. La función enlace $g(u)$ relaciona la media de la distribución de probabilidad con la combinación lineal de las variables independientes.</p>
	$g(u) = \ln\left(\frac{u}{1-u}\right)$ <p style="text-align: right;">Ecuación 21</p> <p>Gaussiano Naive Bayes:</p> <p>El clasificador Naive Bayes Gaussiano es una variante del algoritmo Naive Bayes que asume que las características continuas de las muestras se distribuyen de acuerdo con una distribución gaussiana (o distribución normal), la ecuación es la siguiente:</p>
	$P(y x_1, x_2, \dots x_n) = \frac{P(y) \cdot P(x_1 y) \cdot P(x_2 y) \dots P(x_n y)}{P(x_1) \cdot P(x_2) \dots P(x_n)}$ <p style="text-align: right;">Ecuación 22</p> <p>Donde: $P(y x_1, x_2, \dots x_n)$ es la probabilidad posterior de la clase y dado el vector de características $x_1, x_2, \dots x_n$, $P(y)$ es la probabilidad a priori de la clase y, $P(x_i y)$ es la probabilidad de la característica x_i dada la clase y. En el caso del Naive Bayes Gaussiano, se asume que x_i sigue una distribución gaussiana para cada clase y.</p>
Clasificadores naive de bayes	<p>Kernel Naive Bayes</p> <p>Es un clasificador Naive Bayes que utiliza la distribución kernel para los predictores numéricos y la distribución multinomial para los categoriales.</p>
	<p>Lineal SVM:</p> <p>La ecuación para la máquina de vectores de soporte (del inglés support vector machine, SVM) se basa en encontrar el hiperplano óptimo que maximiza el margen entre las clases en un espacio de características. La forma general de la ecuación para el hiperplano de un SVM lineal es:</p>
Máquina de vectores de soporte (SVM)	$f(x) = wx + b$ <p style="text-align: right;">Ecuación 23</p> <p>Donde: $f(x)$ es la función de decisión, w es el vector de pesos (coeficientes) que define la orientación del hiperplano, x es el vector de características de la instancia de entrada, b es el término de sesgo (bias).</p> <p>Cuadrático SVM:</p>

MODELO	DESCRIPCIÓN
	<p>La SVM cuadrática es una extensión de la SVM lineal que permite modelar relaciones no lineales entre las características y las clases al introducir términos cuadráticos en la función de decisión. La forma general de la ecuación es:</p> $f(x) = w^T x + b + \sum_{i=1}^n \alpha_i (k(x_i, x) + \rho) \quad \text{Ecuación 24}$ <p>Donde: n es el número de vectores de soporte, α_i son los coeficientes de los vectores de soporte, $k(x_i, x)$ es una función kernel que calcula la similitud entre dos instancias, ρ es el término de sesgo del kernel.</p> <p>Cúbica SVM:</p> <p>En SVM, el kernel es una función que se utiliza para transformar los datos de entrada en un espacio de mayor dimensionalidad. El kernel cúbico es un tipo de kernel polinomial que eleva los atributos a la tercera potencia, y se define cómo:</p> $k(x_i, x_j) = (x_i x_j + c)^3 \quad \text{Ecuación 25}$ <p>Donde x_i, y x_j son vectores de características, y c es un término de compensación.</p> <p>Gaussiano fino SVM:</p> <p>La ecuación del kernel Gaussiano (también conocido como kernel radial o RBF) en SVM es:</p> $k(x_i, x_j) = e^{-\frac{\ x_i - x_j\ ^2}{2\sigma^2}} \quad \text{Ecuación 26}$ <p>Donde: σ (sigma) es un parámetro que controla la amplitud del kernel gaussiano.</p> <p>Regresión logístico eficiente:</p> <p>La ecuación de regresión logística es un modelo utilizado para la clasificación binaria. Se utiliza para estimar la probabilidad de que una observación pertenezca a una de las dos clases, y se expresa de la siguiente manera:</p> $P(y = 1 x) = \frac{1}{1 + e^{-z}} \quad \text{Ecuación 27}$ <p>Donde: $P(y = 1 x)$ es la probabilidad de que la observación pertenezca a la clase positiva (clase 1) dadas las características x, z es la función lineal de las características x y los coeficientes.</p> <p>El algoritmo k-Nearest Neighbors (k-NN) no tiene una "ecuación" como tal, ya que no implica un modelo matemático con parámetros ajustables como otros algoritmos de aprendizaje supervisado. En lugar de eso, k-NN es un algoritmo de aprendizaje basado en instancia que clasifica nuevos puntos de datos basándose en la proximidad a los puntos de datos de entrenamiento.</p>
Clasificadores lineales entrenados eficientemente	
Clasificadores de vecinos más cercanos (KNN)	de
Clasificadores de Aproximación del Kernel	de Kernel SVM Regresión logística SVM

Fuente: (MathWorks, 2019b).

5. Metodología

5.1 Equipos y materiales

5.1.1 Equipos

- Computadora persona Dell (procesador Intel inside i5).

5.1.2 Materiales

Recursos tecnológicos

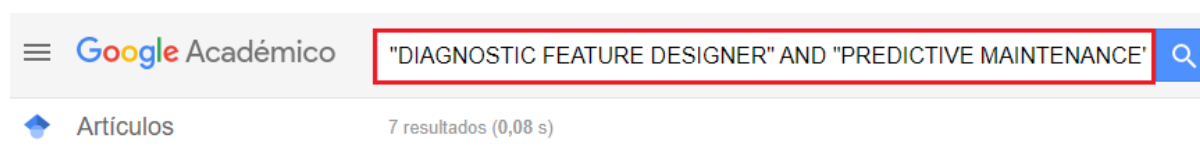
- Internet
- Matlab, versión estudiantil.

5.2 Procedimiento

5.3 Primero objetivo

Para el desarrollo del primer objetivo el cual consiste en “Caracterizar y documentar las funcionalidades de DIAGNOSTIC FEATURE DESIGNER en el contexto del diagnóstico de fallos en plantas eólicas”, se llevó a cabo una revisión de literatura que comenzó con la búsqueda de información relacionada con el uso de la herramienta DFD y el análisis de datos en plantas eólicas. Se empleó un script de búsqueda en Google Scholar utilizando los siguientes términos clave: "DIAGNOSTIC FEATURE DESIGNER" AND "PREDICTIVE MAINTENANCE" AND "MATLAB" AND "WIND TURBINES". Con todo esto el motor de búsqueda de Google Scholar el cuál arrojó un total de 7 resultados, se procedió a la selección de los documentos relevantes mediante la lectura de los resúmenes (abstracts), con el objetivo de determinar su relación con DFD y los temas específicos de interés. En la **Figura 10** se observa el método de búsqueda que se empleó para este trabajo.

Figura 10. Método de búsqueda en Google Académico.



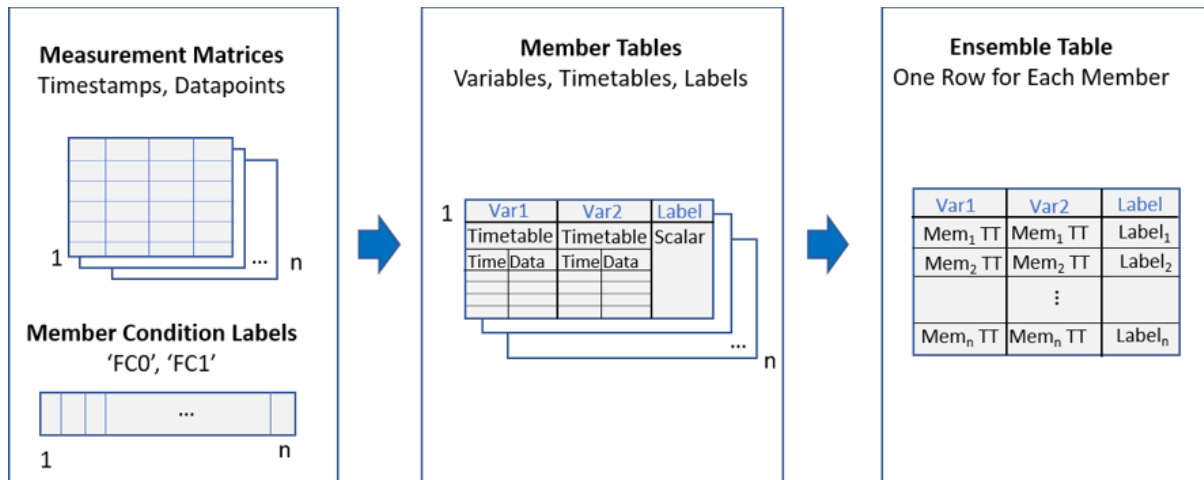
Posteriormente, se realizó un análisis de la documentación disponible en MathWorks, ya que es la empresa dueña de DFD y en la cual se encuentra la gran parte de la información. Se focalizó en los siguientes temas específicos para este trabajo:

5.3.1 Preprocesamiento de datos.

Se consultaron ejemplos disponibles en (MathWorks, 2019g), para ordenar y limpiar los datos, así como para seleccionar variables importantes y etiquetarlas. Se utilizó como guía el ejercicio presentado en (MathWorks, 2019f) para configurar la matriz antes de importar los

datos a DFD. La **Figura 11** ilustra un ejemplo de preparación de datos para su importación a DFD.

Figura 11. Ejemplo de preparación de datos para importar a DFD.



Fuente: (MathWorks, 2019b).

5.3.2 Importar datos.

Se consultó la documentación de (MathWorks, 2019b), para entender cómo llevar los datos a DFD, lo que facilita la elección y preparación de variables, incluyendo variables dependientes e independientes como el tiempo, variables de condición como los fallos en las máquinas estas variables pueden ser del tipo categóricas, numéricas y de tipo texto.

5.3.3 Procesamiento de datos.

Se optó por la interpolación para completar datos faltantes entre fechas y se emplearon métodos como Welch's Method y Autoregressive Model para la estimación espectral, utilizando la documentación de (MathWorks, 2019).

5.3.4 Generación de características.

Se utilizó la documentación de (MathWorks, 2019), para seleccionar indicadores de condición en diferentes dominios, incluyendo el tiempo, la frecuencia y el tiempo-frecuencia.

5.3.5 Selección de características.

Se recurrió nuevamente a la documentación de (MathWorks, 2019), para seleccionar las características más relevantes, utilizando el T-test para dos clases.

5.3.6 Exportar datos.

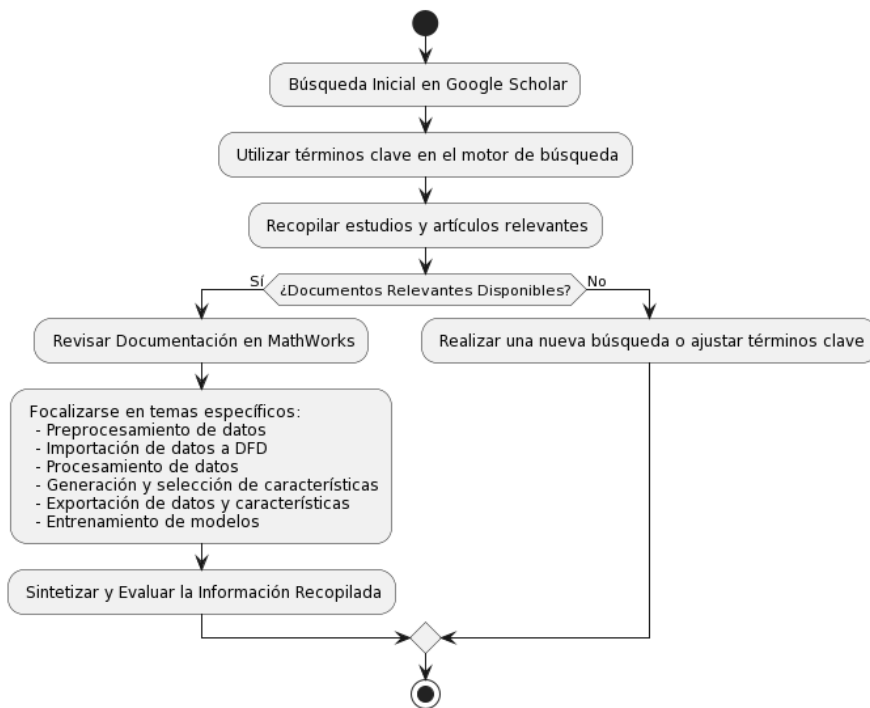
Se consultó la documentación de (MathWorks, 2019), para entender cómo exportar las características a diferentes entornos de trabajo, ya sea en código o como una matriz.

5.3.7 Entrenar modelos.

Aunque existe material de revisión en (MathWorks, 2023a), sobre la Aplicación Classification Learner que proporciona mucha información para entrenar los modelos, se prefirieron los webinars en video ofrecidos por la misma referencia. Estos últimos ofrecen explicaciones más específicas y claras, lo que facilita la comprensión del proceso de entrenamiento de modelos.

Se examinaron ejemplos y tutoriales en MathWorks para comprender mejor cómo aplicar el DFD en el diagnóstico de fallos en plantas eólicas. Además, se exploraron las funciones principales de MATLAB y se observó los webinars grabados en (MathWorks, 2021) centrados en el uso de esta herramienta. Se analizaron detalladamente estos recursos para obtener información práctica y ejemplos específicos de su aplicación en el diagnóstico de fallos. En la **Figura 12** se puede observar el diagrama de flujo de esta metodología.

Figura 12. Metodología para documentar las funcionalidades de DFD.



5.4 Segundo objetivo

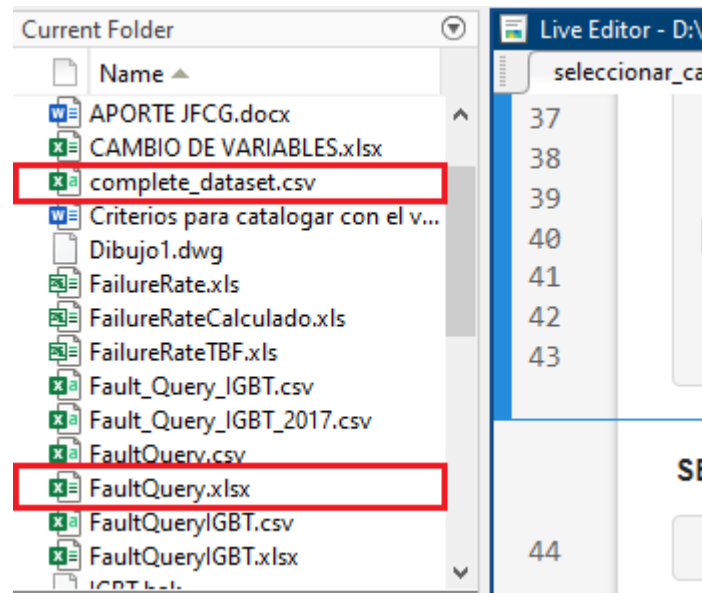
Para alcanzar el segundo objetivo de “Desarrollar modelos de diagnóstico de fallos en plantas eólicas utilizando DIAGNOSTIC FEATURE DESIGNER”, se emplearon los datos provenientes de la Central Eólica Villonaco. En el **Anexo 1** se detallan las variables capturadas por los sensores durante el monitoreo. Durante todo el proceso, se aplicó un enfoque experimental, combinado con la teoría de prueba y error. Esto implicó una constante retroalimentación y toma de decisiones en la creación de modelos de predicción. A

continuación, se describen los pasos seguidos para desarrollar este proceso y así obtener modelos de diagnóstico utilizando la aplicación DFD.

5.4.1 *Recolección de datos.*

Para poder utilizar los archivos, es necesario primero organizar todos los archivos proporcionados dentro del computador. Estos archivos, inicialmente, son del tipo csv, xlsx y txt. Sin embargo, no todos los archivos son utilizables. En la **Figura 13**, se puede observar un ejemplo de archivos guardados en una carpeta dentro del ordenador, mostrados en el área de trabajo de Matlab. Los archivos marcados son empleados para el preprocesamiento de los datos proporcionados por CELEC.

Figura 13. Carpeta de archivos de destino en Matlab.



5.4.2 *Preprocesamiento de los datos.*

A continuación, una vez que las matrices han sido exportadas a Matlab, se procede a seleccionarlas por categoría utilizando la función 'find' con el propósito de elegir un único aerogenerador. Además, se seleccionan tres temperaturas del convertidor para su análisis. Esta tarea se realiza con el fin de facilitar el análisis y ahorrar recursos computacionales. Para llevar a cabo la limpieza de los datos en busca de valores atípicos, faltantes o la selección de variables importantes, se ha diseñado un código, el cual se puede encontrar detallado en el **Anexo 2**.

Se comparó entre el archivo de datos de los sensores, “complete_dataset.csv”, y el historial de fallos mostrados con anterioridad, “FaultQuery.xlsx”. Esto se realizó para buscar alguna relación entre los fallos y las fechas en que ocurrieron en el aerogenerador. El código utilizado se encuentra en el **Anexo 3**. Una vez completadas las tareas de limpieza, selección y búsqueda de datos atípicos, tal como se muestra en resultados de la sección de

preprocesamiento e importe de datos, se ha desarrollado un algoritmo para crear un ensamblaje por fechas. Estas fechas se clasificaron en días con una frecuencia de muestreo de 10 minutos, comprendido entre 2014 y 2021. Para obtener más información sobre el preprocesamiento y las funciones utilizadas, se recomienda consultar la **Tabla 9**.

Tabla 9. Funciones utilizadas en el preprocesamiento.

Tarea de Preprocesamiento	Funciones Utilizadas
Ordenar filas en una matriz	Sortrows, find
Limpiar datos faltantes y rellenar datos	Fillmissing, linear, filloutliers, previous, quartiles
Ensamble de matriz por miembros	Zeros, numel, unique, size, lenght, all, array2timetable, table, cellfun, ver Anexo 9 .
Eliminar fechas que se repiten	All, diff.

En la **Figura 14** se logra ver las temperaturas seleccionadas del convertidor, las cuales fueron útiles para comparar los fallos del aerogenerador en el tiempo, desde el inicio hasta el fin de la falla. En el algoritmo del **Anexo 3** se puede ver la comparación y en relación a estos fallos asignamos los números [1, 0] para un estado de anomalía y normal, respectivamente.

Figura 14 Temperaturas seleccionadas del convertidor.

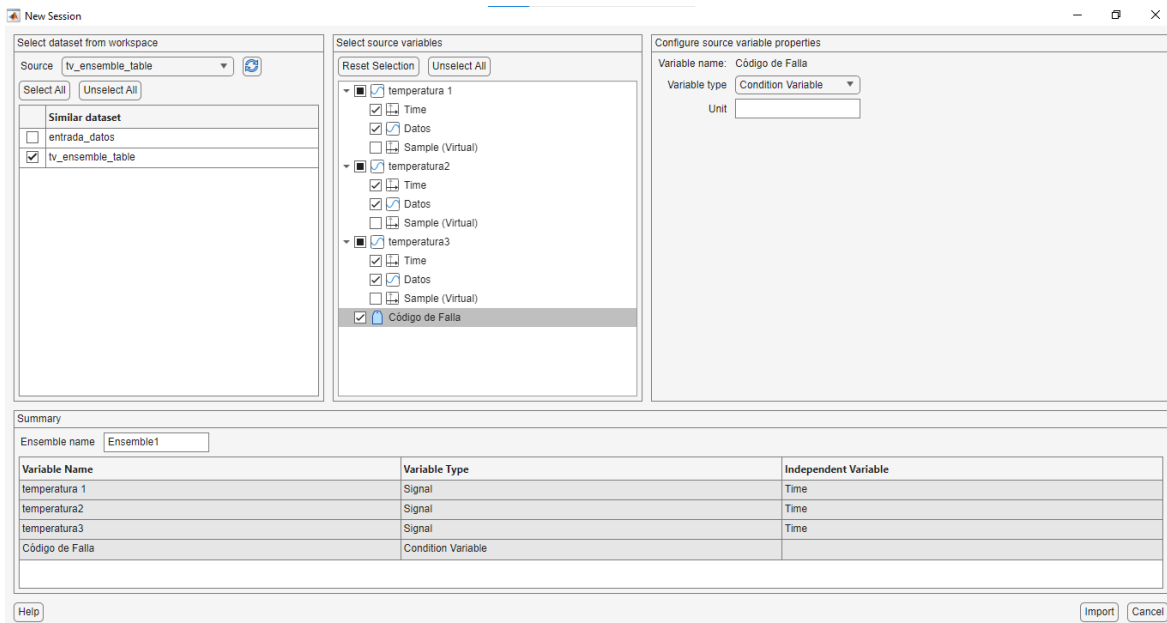
1 timestamp	2 pitch_converter_temperature_1_max	3 pitch_converter_temperature_2_max	4 pitch_converter_temperature_3_max
2014-01-01 00:00:00	18.4000	18.3000	18.3000
2014-01-01 00:10:00	18.4000	18.3000	18.3000
2014-01-01 00:20:00	18.3000	18.3000	18.3000
2014-01-01 00:30:00	18.3000	18.2000	18.3000
2014-01-01 00:40:00	18.3000	18.2000	18.3000
2014-01-01 00:50:00	18.3000	18.2000	18.3000
2014-01-01 01:00:00	18.3000	18.2000	18.3000

	FaultStartTime	WTG	FaultEndTime	FaultLength_s	FaultLength	FaultDescription	Codigo_Fallo
1	20-Apr-2015 15:29:42.060	S01	20-Apr-2015 18:51:40.543	12118	3.3660	(107)Error_profibus_20# station power supply	0
2	16-May-2015 12:39:16.330	S01	16-May-2015 12:50:06.360	650	0.1810	(420)Error_converter IGBT cooling 1# fan feedback loss	1
3	28-May-2015 04:57:14.733	S01	28-May-2015 06:12:13.717	4499	1.2500	(435)Error_converter not ready	1
4	28-May-2015 13:22:26.077	S01	28-May-2015 13:41:33.170	1147	0.3190	(52)Error_grid_LVRT over time	0
5	27-Jul-2015 06:24:39.463	S01	27-Jul-2015 06:37:08.587	749	0.2080	75	0
6	04-Aug-2015 10:13:25.933	S01	10-Aug-2015 01:27:26.337	486841	135.2340	(217)Error_pitch blade angle comparing	0
7	10-Aug-2015 02:31:24.247	S01	10-Aug-2015 02:32:08.730	44	0.0120	(217)Error_pitch blade angle comparing	0
8	12-Aug-2015 03:00:41.323	S01	12-Aug-2015 03:03:08.497	147	0.0410	(217)Error_pitch blade angle comparing	0
9	12-Aug-2015 03:19:23.997	S01	12-Aug-2015 03:21:51.683	148	0.0410	(217)Error_pitch blade angle comparing	0
10	12-Aug-2015 03:29:37.447	S01	12-Aug-2015 03:32:08.120	151	0.0420	(217)Error_pitch blade angle comparing	0
11	12-Aug-2015 03:34:42.510	S01	12-Aug-2015 03:37:13.040	151	0.0420	(217)Error_pitch blade angle comparing	0
12	21-Aug-2015 17:03:38.697	S01	21-Aug-2015 18:34:22.213	5444	1.5120	(120)Error_profibus_41# station diagnostic	0
13	28-Aug-2015 08:56:27.387	S01	28-Aug-2015 10:18:51.010	4944	1.3730	(92)Error_safety system_overspeed module	0

5.4.3 Importar datos a DFD

Para importar la tabla de ensamble `tv_ensemble_table`, que es un archivo preparado para ser importado a DFD, primero debemos seleccionar el programa correspondiente en las herramientas de Matlab. Luego, seleccionamos las variables de temperatura junto con sus códigos de falla, los cuales fueron descritos anteriormente. En la **Figura 15** se muestra la ventana de configuración de los datos antes de transferirlos a DFD.

Figura 15. Ventana para la selección y configuración de las variables.

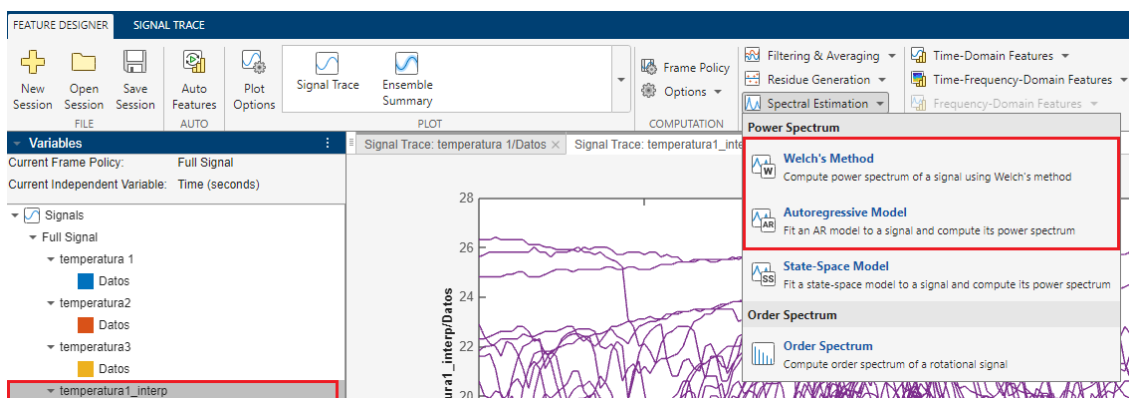


Nota. Datos de temperaturas del convertidor configurados para importar a la aplicación DFD.

5.4.4 Procesar datos en DFD

Después de obtener las variables de temperatura en DFD, se aplicó interpolación para realizar el preprocesamiento de las señales. La interpolación se empleó para completar algunos datos; por ejemplo, cada día debe contener 144 datos. Con la interpolación, podemos llenar los datos faltantes utilizando los valores siguientes (next). Para generar señales en el dominio de la frecuencia, utilizamos el método de Welch, el cual mide la densidad espectral. Además, empleamos el modelo Autorregresivo para modelar la dependencia temporal en las series temporales y prever valores futuros. En la **Figura 16**, se pueden observar estas opciones disponibles en DFD. El código utilizado para estas tareas se encuentra detallado en el **Anexo 4**, junto con las ecuaciones mostradas en la **Tabla 3**.

Figura 16. Métodos de welch & autorregresivo.

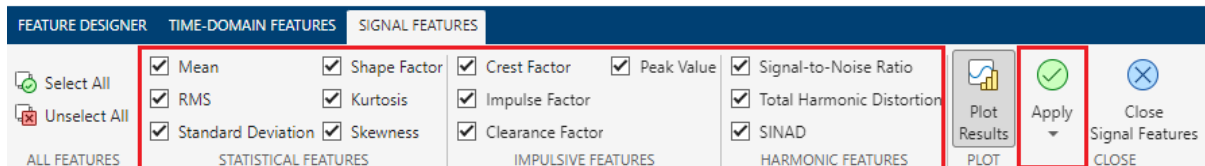


5.4.5 Indicadores de condición.

5.4.5.1 Características en el dominio del tiempo:

Para generar indicadores de condición, utilizamos las características que aparecen en el dominio del tiempo (Signal Features). Esta opción, que está disponible en el programa, permite el uso de los estadísticos descritos en la **Tabla 4**, con cada una de sus ecuaciones. En la **Figura 17**, podemos ver la representación los estadísticos en DFD.

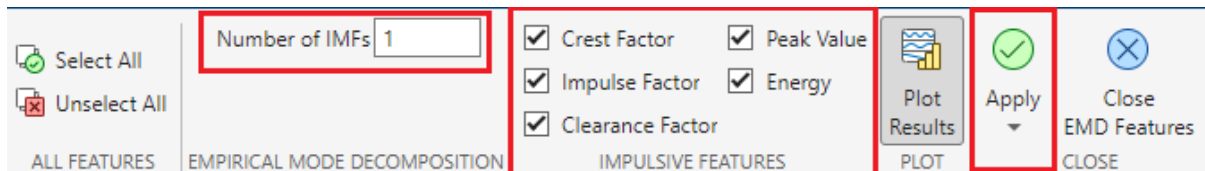
Figura 17. Estadísticos en el dominio del tiempo.



5.4.5.2 Características en el dominio tiempo-frecuencia:

Para encontrar las características en el dominio del tiempo-frecuencia, primero utilizamos la señal procesada en el dominio del tiempo, “temperatura1_interp”, mostrada en la **Figura 16**. De todo el conjunto que ofrece la herramienta elegimos la descomposición del modo empírico (EMD). En la **Tabla 5** se describe más a detalle esta descomposición, en la **Figura 18** se puede ver la opción para realizar este tipo de análisis.

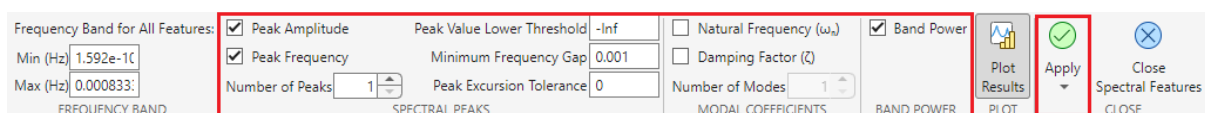
Figura 18. Descomposición de modo empírico.



5.4.5.3 Características en el dominio de la frecuencia:

Para generar características en el dominio de la frecuencia, empleamos las señales obtenidas mediante el método de Welch y el modelo autorregresivo temperatura1_ps y temperatura1_ps_1, respectivamente. En la **Tabla 6** se presenta una descripción más detallada de los estadísticos utilizados. Además, **Figura 19** se observan estos estadísticos empleados para el análisis espectral.

Figura 19. Estadísticos de la frecuencia.

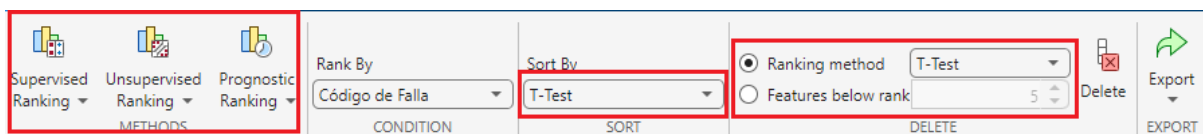


El código utilizado para las características se lo puede ver en el **Anexo 4**.

5.4.6 Selección de características.

Una vez que se ha identificado el conjunto de características, es necesario seleccionar las más relevantes. Para ello, se recurre a algunos de los métodos proporcionados por la herramienta. Dado que este estudio aborda el mantenimiento predictivo de dos clases, se opta por el método supervisado, método que se utiliza cuando ya se tiene un historial en los datos, en este caso fallos en el aerogenerador. Se utiliza el método de cálculo estadístico T-test en la selección de las mejores características para el entrenamiento. En la **Tabla 7** se detallan las ecuaciones utilizadas en el proceso de selección de características, mientras que en la **Figura 20** se muestra la ventana con las opciones disponibles en DFD, observar el código diseñado en el **Anexo 5**.

Figura 20. Opciones de selección de características en DFD.

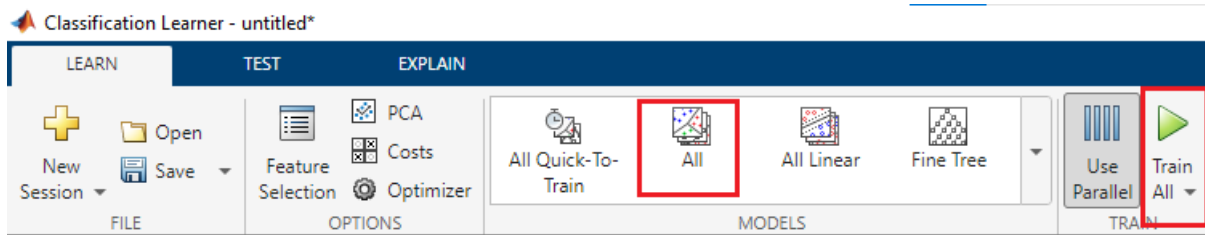


5.4.7 Entrenar modelos

Para entrenar los modelos, empleamos las características previamente seleccionadas utilizando la técnica supervisada T-test. Posteriormente, exportamos estos datos al Classification Learner, programa que nos facilitó el entrenamiento de modelos predictivos. Se seleccionó todos los modelos del programa disponibles con la finalidad de encontrar el modelo de mejor precisión. En la **Figura 21**, se puede observar cómo seleccionamos las opciones de entrenamiento. Además, en la **Tabla 8** se detallan las ecuaciones utilizadas por el programa, mientras que en el **Anexo 6** se presenta el algoritmo empleado por el programa para entrenar modelos.

Para entrenar los modelos, utilizamos las características previamente seleccionadas mediante la técnica supervisada del T-test. Posteriormente, exportamos estos datos al **Classification Learner**, un programa que facilita el entrenamiento de modelos predictivos. Seleccionamos todos los modelos disponibles en el programa para determinar cuál de ellos ofrecía la mejor precisión. En la **Figura 21**, se muestra el proceso de selección de las opciones de entrenamiento. Además, en la **Tabla 8** se detallan las ecuaciones utilizadas por el programa, mientras que en el **Anexo 6** se presenta el algoritmo empleado por el programa para el entrenamiento de los modelos.

Figura 21. Ventana de selección de modelos de aprendizaje (Classification Learner).



En la **Figura 22** se muestra toda la metodología desarrollada para generar el proceso descrito con anterioridad, los pasos pueden cambiar de acuerdo a la mejoría de los resultados en el modelo final.

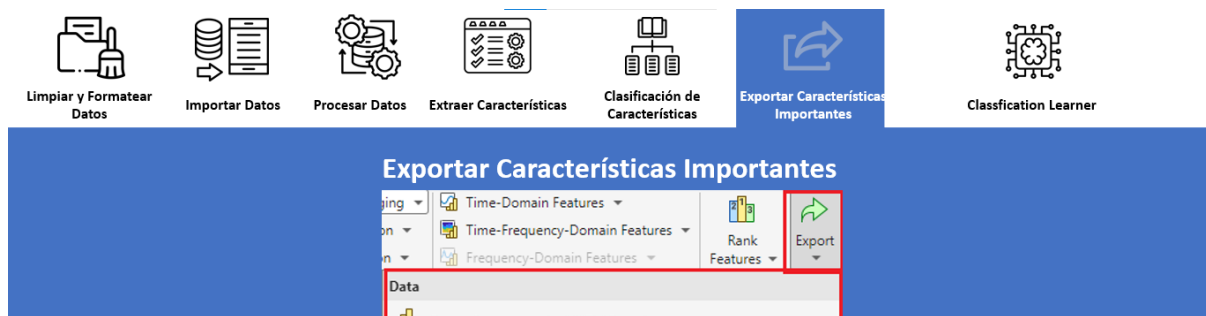
Figura 22. Metodología para el desarrollo de modelos de diagnóstico



5.5 Tercer objetivo

Para “Generar guías de uso e interpretación de resultados para DFD en el contexto de plantas eólicas”, se utilizó las ventanas diseñadas en forma secuencial para cada uno de los procesos. Estas ventanas se diseñaron utilizando íconos, con el fin que sea más comprensible a la hora de presentar los resultados en el análisis de señales de un aerogenerador. Los resultados fueron trabajados en la herramienta DFD, y se lo puede ver en el diseño mostrado en la **Figura 23**.

Figura 23. Esquema del manual de resultados.



6. Resultados

6.1 Primer objetivo

El primer objetivo se trata de “Caracterizar y documentar las funcionalidades de DFD en el contexto del diagnóstico de fallos en plantas eólicas”. Para ello, se ha recopilado material de diversas fuentes, que incluyen videos, artículos científicos y sitios web. En **Tabla 10**, se presenta la documentación de cada una de las funcionalidades junto con su descripción correspondiente.

Tabla 10. Documentación y características de DFD.

Referencias videos, webinars	Funcionalidad	Descripción
(MathWorks, 2019).	Preprocesar los datos	Documentación para comprender la organización de los datos proporcionados, limpieza de datos (incluyendo datos faltantes y por completar) y detección de datos atípicos.
(MathWorks, 2019i, 2023b)	Importar datos	Documentación para importar datos preprocesados y estructurados a DFD, incluyendo códigos de fallos. Los datos consisten en tres temperaturas con códigos de fallo [0,1].
(MathWorks, 2019b, 2022)	Procesar datos en	Documentación necesaria en MathWorks para la creación de nuevas variables mediante interpolación, ya sea señales promediadas en el tiempo o señales espectrales.
(MathWorks, 2019b, 2019i, 2022, 2023c)	Generar Características	Documentación necesaria para generar características o indicadores de condición mediante histogramas en el dominio del tiempo, tiempo-frecuencia y frecuencia, así como métricas no lineales y espectrales
(MathWorks, 2019b, 2022, 2023c)	Clasificación de características	Documentación para seleccionar el método de selección de características adecuado, es decir, elegir las correctas para el modelo de diagnóstico de fallas. Entre estas opciones se encuentran: Clasificación supervisada: ANOVA unidireccional, T-test, Bhattacharya, Kruskal-Wallis, Entropía, Curva característica operativa del receptor, Wilcoxon, Prueba T. Clasificación no supervisada: puntuación Laplaciano, varianza."
(MathWorks, 2019b, 2019i, 2022, 2023c)	Entrenar Modelos y exportar modelos	Documentación para exportar y entrenar los modelos en la aplicación CLASSIFICATION LEARNER, además de validar la precisión del modelo mediante porcentaje.

6.2 Segundo objetivo

6.2.1 Recolección de datos

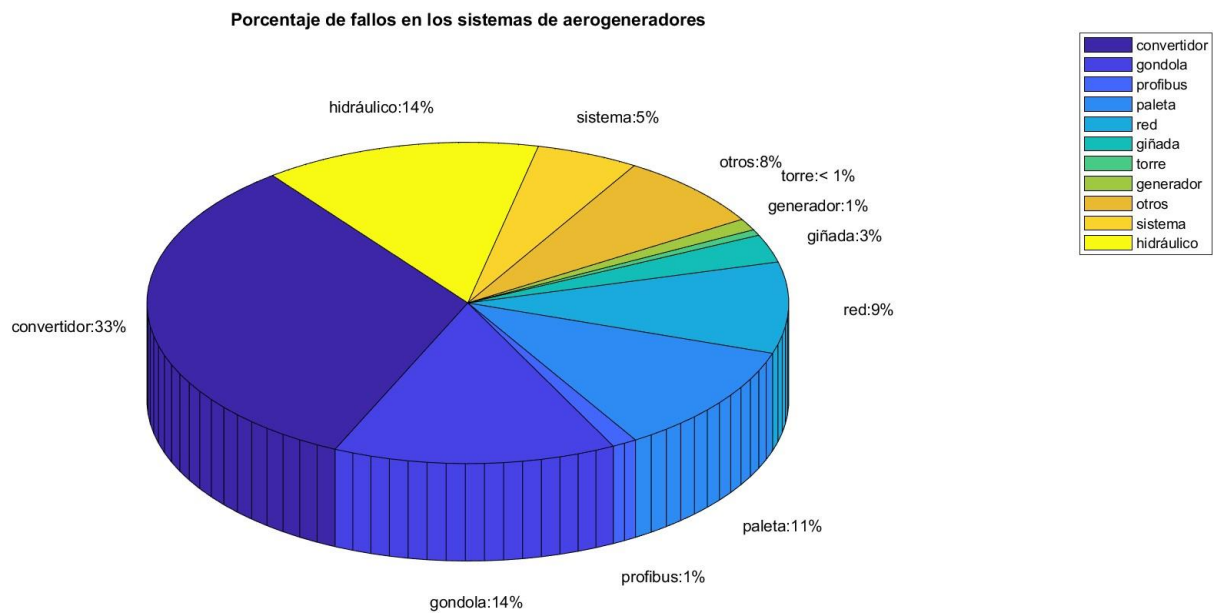
Se ha seleccionado el archivo “complete_dataset.csv” para importarlo a Matlab. Este archivo contiene 70 variables, lo que equivale a una matriz de tamaño [383974, 70]. Todas estas variables son de tipo double. El código utilizado para importar este archivo se encuentra en el **Anexo 7**. Además, también se incluye el archivo FaultQuery.xlsx (Consulta de falla), que consta de 99 categorías. Cada vez que ocurre una falla, estas categorías nos indican un código de error. Todo el historial de los códigos de falla ha sido analizado en Matlab, y se pueden observar estos fallos en la **Figura 24**.

Figura 24. Lista de fallos de los aerogeneradores.

(1)Error_Nacelle acceleration...	853
(10)Error_Nacelle temperature...	1
(107)Error_profibus_20# stati...	7
(115)Error_profibus_10# stati...	40
(117)Error_profibus_20# stati...	1
(120)Error_profibus_41# stati...	24
(121)Error_profibus_42# stati...	4
(122)Error_profibus_43# stati...	2
(131)Error_wind vane	690
.	
.	
.	
.	
(97)Error_safety system_PLC e...	8
(98)Error_safety system_safet...	4
30	166
73	18
75	173
76	10
762	92

Se realizó un análisis de los distintos componentes que integran un aerogenerador mediante un gráfico de pastel, con el fin de identificar aquellos que presentan una mayor cantidad de fallos. En la **Figura 25** se aprecia que el convertidor registra un 33% de fallos, una cifra superior a la de los otros sistemas. Por consiguiente, se enfocó el estudio en el convertidor y se seleccionaron tres temperaturas obtenidas a través de los sensores instalados en dicho componente. Estas variables específicas se detallan en el **Anexo 8**.

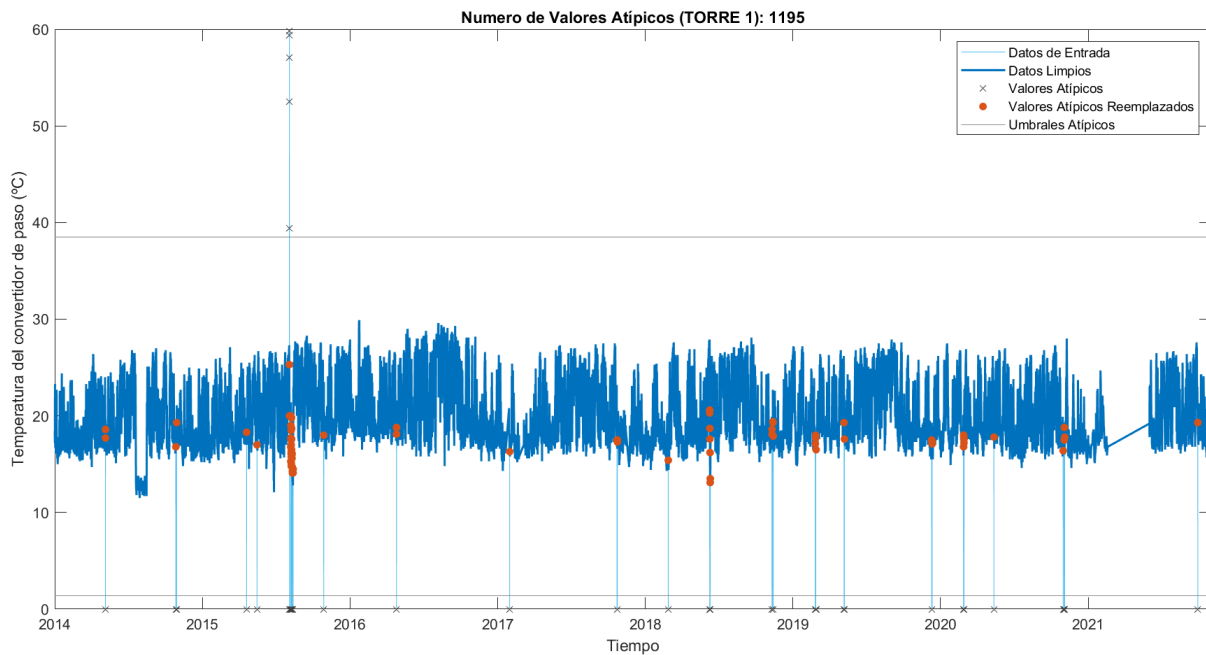
Figura 25. Porcentaje de fallo en los aerogeneradores.



6.2.2 Preprocesamiento e importación de datos

Una vez seleccionadas las tres temperaturas del convertidor, mediante el algoritmo de preprocesamiento se logró gestionar tanto los datos faltantes como los valores atípicos. La **Figura 26** se observa que la señal de temperatura presenta picos altos, con temperaturas superiores a los 50-60 °C, así como valores de temperatura por debajo de 0 °C. En este aerogenerador, no se encontraron valores faltantes tipo NaN. Para tratar los datos faltantes, se implementó la interpolación lineal, mientras que para los valores atípicos se emplearon el valor previo y los cuartiles, utilizando un umbral de 3. Se utiliza los métodos descritos en la **Tabla 1** y **Tabla 2**, con la **Ecuación 1** y **Ecuación 3**, que son utilizadas en el algoritmo de preprocesamiento.

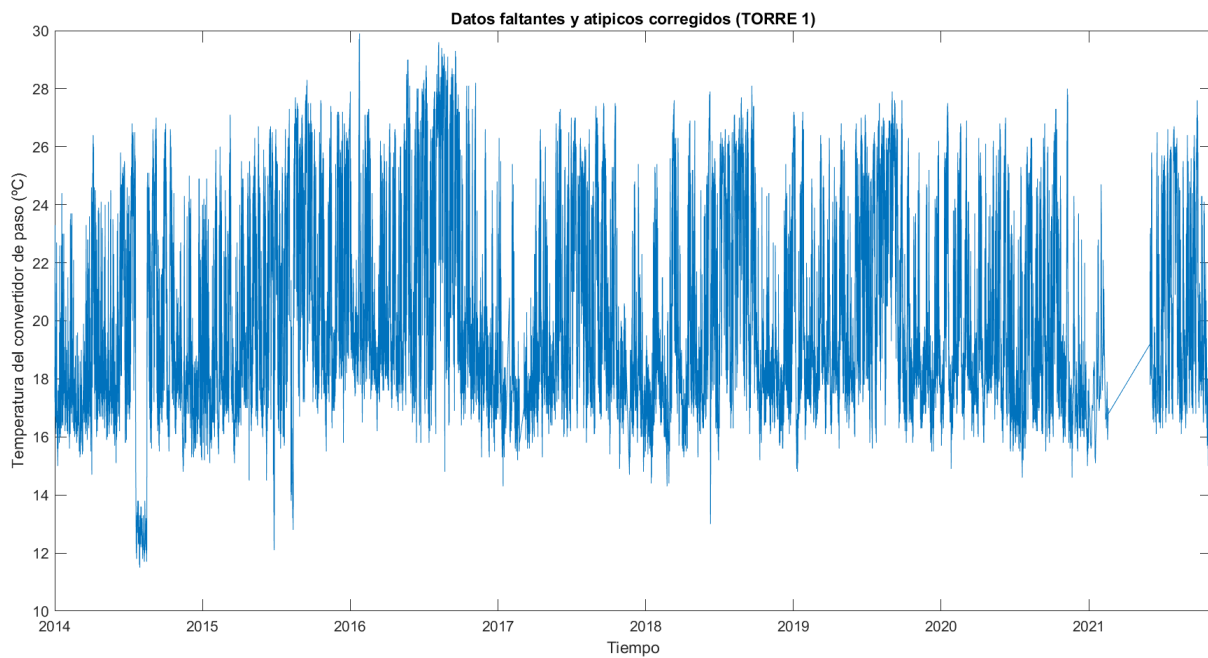
Figura 26. Datos limpios faltantes y atípicos



En la **Figura 27**, se presentan los resultados del procesamiento de los datos faltantes y atípicos. En (MathWorks, 2019d), se emplea la interpolación y el uso del valor previo para completar los datos faltantes y corregir los datos atípicos. Aunque existen otros métodos disponibles, se optó por la interpolación y el uso del valor previo con el fin de optimizar los recursos de procesamiento. Las temperaturas con un valor de cero y las temperaturas superiores al umbral de 3 se sustituyeron por el valor previo. Esta acción se llevó a cabo debido a la falta de coherencia en los datos, lo cual podría afectar el análisis en los datos.

Los datos anteriores presentaban datos atípicos, datos incoherentes, las temperaturas no pueden ser muy bajas, ahora se puede ver datos reemplazados más cercanos a las temperaturas de un convertidor.

Figura 27. Resultados preprocesamiento de datos aerogenerador.

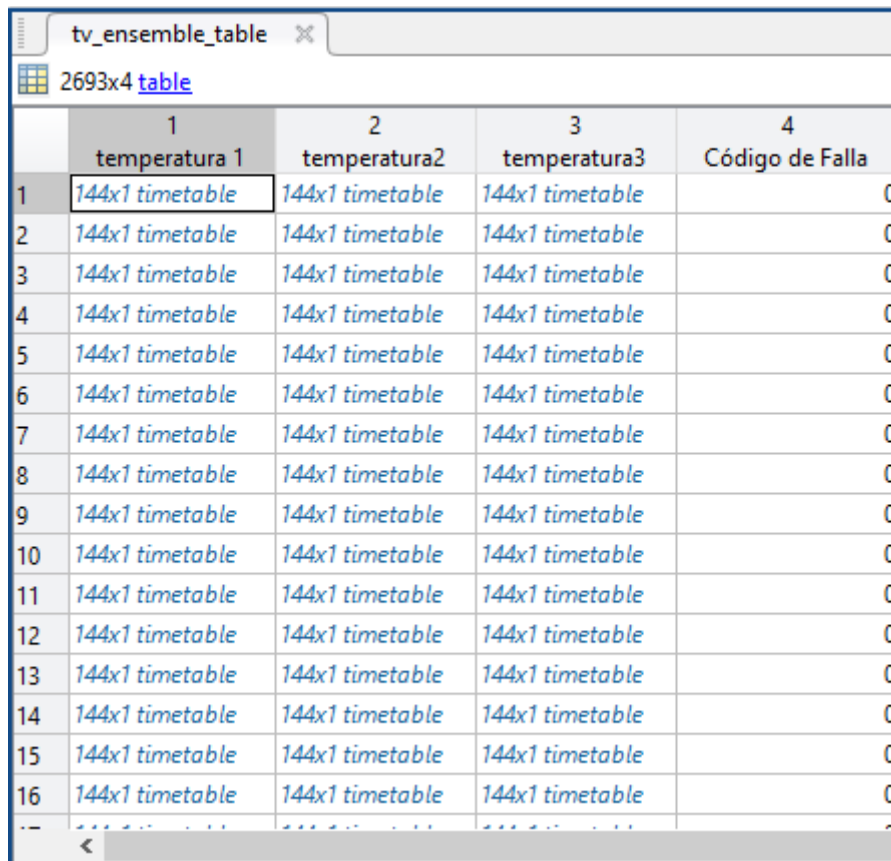


Fuente: Autor

Una vez que se realizó un preprocesamiento previo, se procede a preparar la matriz que incluye las temperaturas seleccionadas previamente. Estas temperaturas se organizan por días, dado que el sensor capturó mediciones cada 10 minutos. El algoritmo utilizado se detalla en el **Anexo 9**. Además, en la **Figura 28** se ilustra cómo se clasificaron los datos por día, asignándoles un código de fallo de [0 1]. Un valor de 0 indica un día en estado normal, mientras que un valor de 1 indica un día con un fallo detectado.

Esta matriz procesada puede importarse al DFD. Es importante destacar que el tamaño de esta matriz es [2693, 4], donde cada fila representa un miembro, y cada miembro contiene datos correspondientes a 144 mediciones por día. Esta estructura se presenta en forma de tabla, siendo cada entrada de la tabla un tipo de datos timetable.

Figura 28. Ensamble de tablas dentro de una matriz.



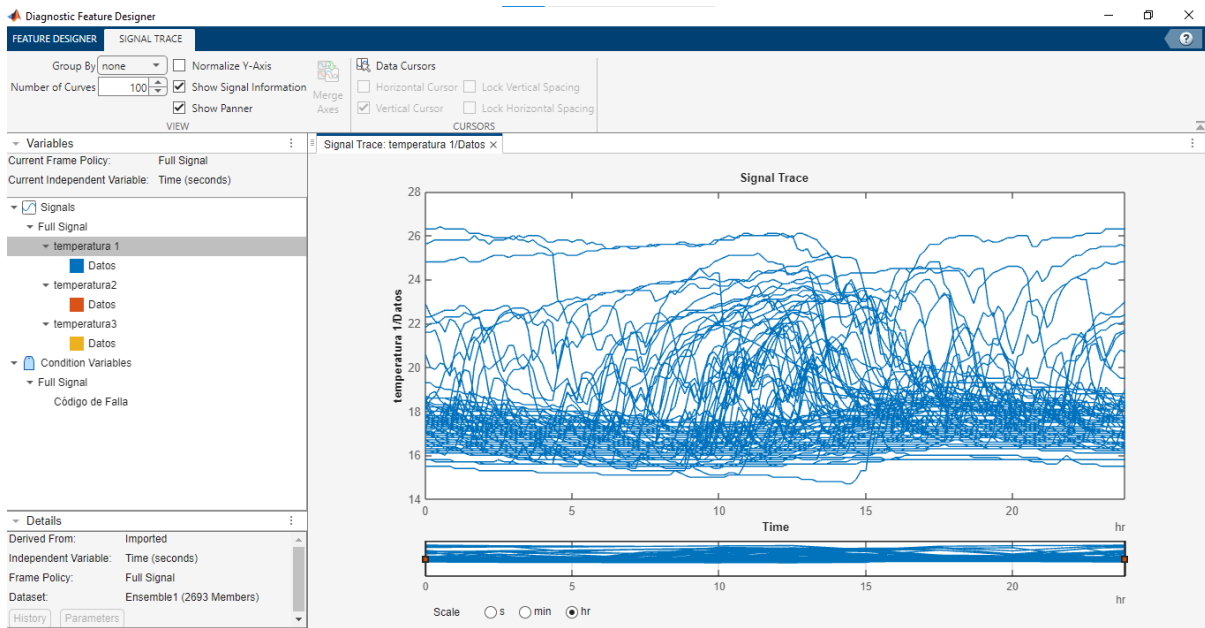
	1 temperatura 1	2 temperatura2	3 temperatura3	4 Código de Falla
1	144x1 timetable	144x1 timetable	144x1 timetable	0
2	144x1 timetable	144x1 timetable	144x1 timetable	0
3	144x1 timetable	144x1 timetable	144x1 timetable	0
4	144x1 timetable	144x1 timetable	144x1 timetable	0
5	144x1 timetable	144x1 timetable	144x1 timetable	0
6	144x1 timetable	144x1 timetable	144x1 timetable	0
7	144x1 timetable	144x1 timetable	144x1 timetable	0
8	144x1 timetable	144x1 timetable	144x1 timetable	0
9	144x1 timetable	144x1 timetable	144x1 timetable	0
10	144x1 timetable	144x1 timetable	144x1 timetable	0
11	144x1 timetable	144x1 timetable	144x1 timetable	0
12	144x1 timetable	144x1 timetable	144x1 timetable	0
13	144x1 timetable	144x1 timetable	144x1 timetable	0
14	144x1 timetable	144x1 timetable	144x1 timetable	0
15	144x1 timetable	144x1 timetable	144x1 timetable	0
16	144x1 timetable	144x1 timetable	144x1 timetable	0

Nota: El ensamble de las temperaturas es por día, junto con su código de fallo [0,1] cada 10 minutos.
Fuente: Autor

En la **Figura 11** se presenta la interfaz de la ventana de importación de DFD. Una vez configurada, permite la importación y análisis de las funcionalidades del programa. En esta ventana, es posible seleccionar un conjunto de datos y las variables con las que se trabajará en el programa, incluyendo variables independientes y variables de condición. En este caso específico, se eligió la variable "tv_ensemble_table", que contiene las tres temperaturas previamente procesadas. Finalmente, al hacer clic en "importar", se procede a la ventana de procesamiento.

En la interfaz de DFD, se encuentran varias herramientas disponibles para el procesamiento de las tres temperaturas. Sin embargo, se decidió enfocarse únicamente en una de las temperaturas (temperatura 1), ya que los datos de las otras dos temperaturas son similares. En la **Figura 29**, se aprecian las diversas funcionalidades que ofrece la aplicación, que incluyen procesamiento de datos, generación de características, clasificación de las características más importantes y exportación de características.

Figura 29. Ventana de interfaz DFD.

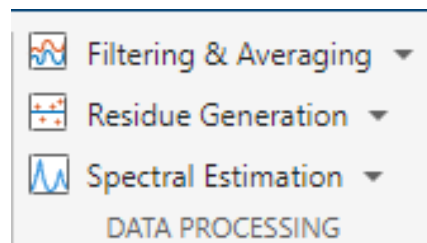


Nota: Señales de los datos de temperatura durante un día vistos en DFD.

6.2.3 Procesar los datos

En la herramienta, dentro de la sección de procesamiento de datos (Data Processing), se ofrece una variedad de opciones para el procesamiento. En nuestro estudio, optamos por la interpolación. En la **Figura 30** se presentan las distintas alternativas de procesamiento de datos disponibles.

Figura 30. Formas de procesamiento de los datos.



En el procesamiento de los datos, se implementó la interpolación para completar los datos faltantes entre fechas. Además, se eliminaron las matrices con menos de 10 filas, estos datos no aparecían en la matriz porque los sensores, en ocasiones, no capturan bien las señales. Este problema se identificó al encontrar mensajes de errores en el programa, sobre todo al realizar análisis espectral. Entonces para prevenir posibles errores durante el análisis de señales dentro de DFD se eliminó estos datos menores a 10 filas, considerando que, para un día completo en el ensamblaje, "tv_ensemble_table", se requieren de 144 filas. También se

llevó a cabo un análisis espectral utilizando el método de Welch y el modelo autorregresivo. Estos análisis son fundamentales para identificar características en el dominio de la frecuencia.

En la **Figura 31** se muestra una parte de la tabla resultante del procesamiento, donde se pueden observar las siguientes columnas: "Interpolación" (temperatura1_interp), "Método de Welch" (temperatura1_ps), y "Modelo autorregresivo" (temperatura1_ps_1).

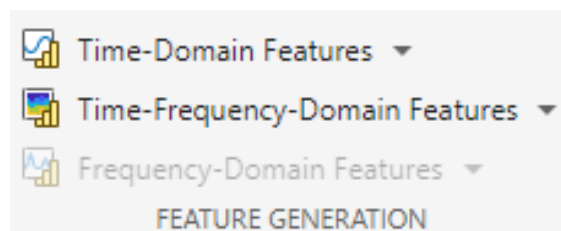
Figura 31. Procesamiento: interpolación, análisis espectral (Welch & Autorregresivo).

1	2	3	4	5	6	7
temperatura 1	temperatura2	temperatura3	temperatura1_interp	temperatura1_ps	temperatura1_ps_1	Código de Falla
143x1 timetable	143x1 timetable	143x1 timetable	144x2 table	129x2 table	45x2 table	0
143x1 timetable	143x1 timetable	143x1 timetable	144x2 table	129x2 table	84x2 table	0
129x1 timetable	129x1 timetable	129x1 timetable	144x2 table	129x2 table	37x2 table	0
144x1 timetable	144x1 timetable	144x1 timetable	144x2 table	129x2 table	96x2 table	0

6.2.4 Indicadores de condición

Dentro de la aplicación, se encuentra una herramienta denominada "Generación de Características" (Feature Generation). Una vez que se ha completado el procesamiento de las señales, es posible realizar análisis en diferentes dominios: tiempo, tiempo-frecuencia y frecuencia. En la **Figura 32** presenta las diversas funcionalidades del programa para generar indicadores de condición.

Figura 32. Funcionalidad del programa para los indicadores de condición.

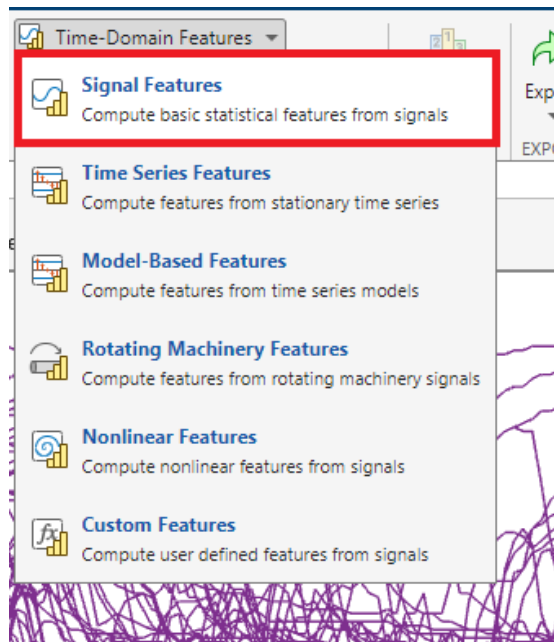


Nota: Panel de Herramientas para la generación de características en DFD

6.2.5 Características en el dominio del tiempo

Una vez procesados los datos de temperatura mediante interpolación, se emplearon las características en el dominio del tiempo (Time-Domain Features). Aquí se incluyen estadísticas específicas correspondientes al dominio temporal. Para este estudio, se seleccionaron las características de señales indicadas en la **Figura 33**.

Figura 33. Características de las señales (Características en el dominio del tiempo).



Los resultados del cálculo de estas características se presentan en forma de histogramas. Las ecuaciones de cálculo correspondientes se detallan en la **Tabla 4**. En la **Figura 34** y **Figura 35** la barra azul indica el estado normal [0], mientras que la barra naranja representa el estado de fallo [1]. Se observa la dispersión de los datos en cada gráfica, evidenciando así la contribución de cada uno de estos estadísticos al análisis en el dominio del tiempo.

Figura 34. Histograma de características en el tiempo.

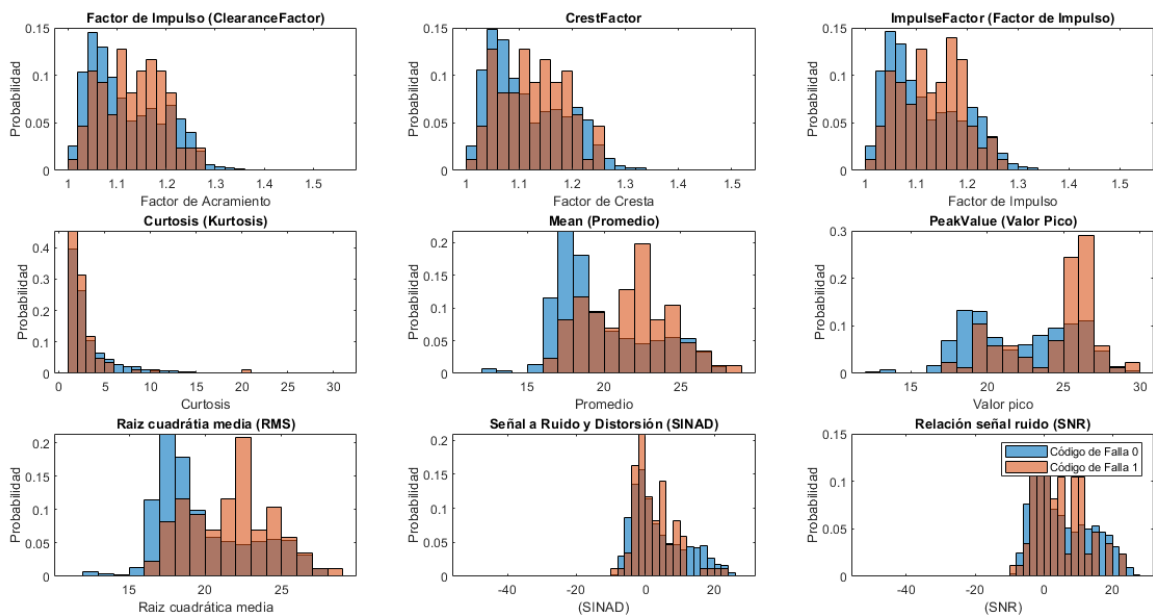
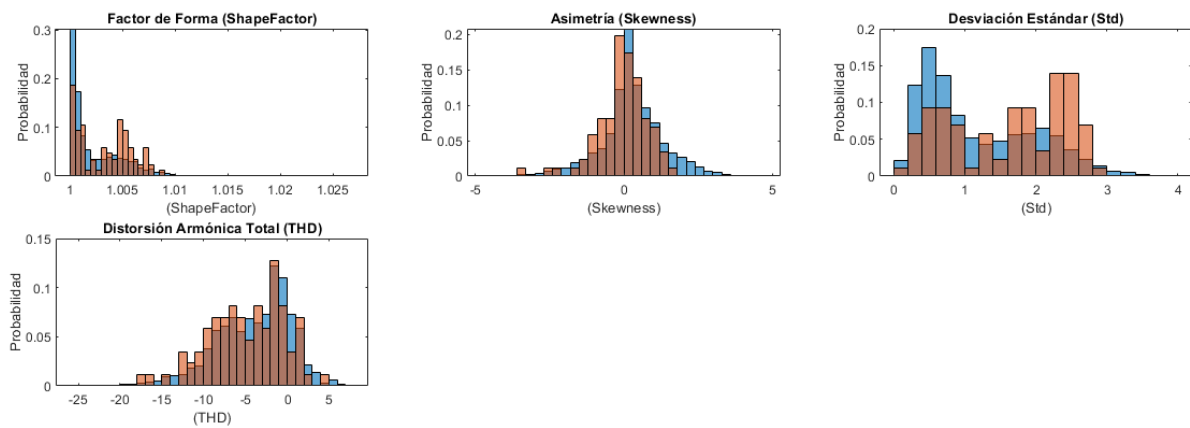


Figura 35. Histograma 2 de las características en el tiempo.



El análisis que se realiza a cada uno de estos estadísticos se lo de detalla en la **Tabla 11**, basado en la temperatura 1 del convertidor.

Tabla 11. Análisis de los resultados estadísticos en el dominio del tiempo.

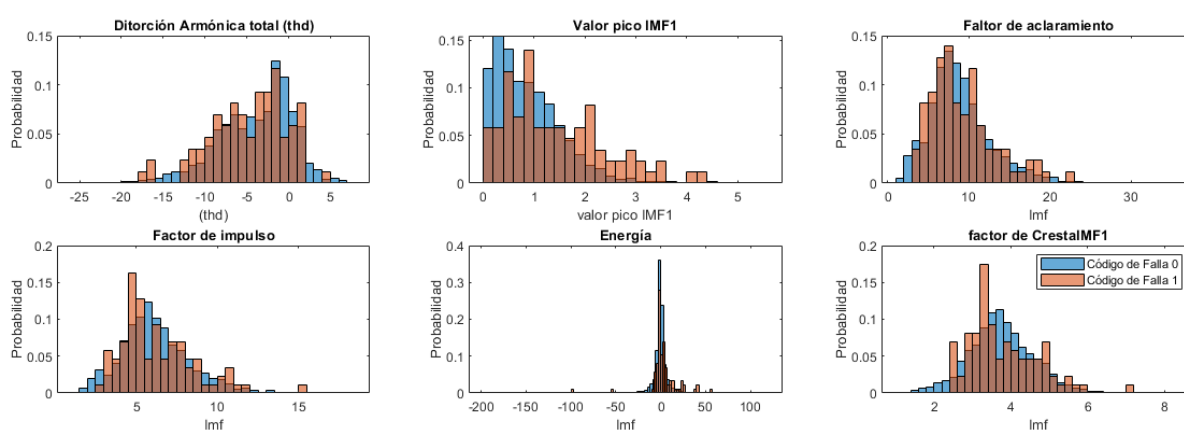
Estadístico	Resultados
Valor máximo o valor pico (Peak Value)	Al analizar el histograma, se evidencia que los valores máximos de temperatura registrados durante el día, en el rango de 26-27 °C, alcanzan una probabilidad del 29.06% para el estado normal [0]. Por otro lado, las temperaturas comprendidas entre 18-19 °C muestran una probabilidad del 13.34% para el estado de falla [1].
Factor de Aclaramiento (Clearance Factor)	Este factor tiene una probabilidad de 14.53 % para los datos normales [0] y 12.79% para los datos de fallo [1].
Raíz Cuadrática Media (RMS)	Este histograma muestra que la mayoría de los datos de temperatura se concentran en el rango de 17 a 18 °C, con una probabilidad del 21.40% para los datos correspondientes al estado normal [0]. Por otro lado, el 20.29% de los datos de temperatura se sitúan en el intervalo de 22 a 23 °C para el estado de fallo [1].
Factor de Cresta (Crest Factor)	El factor que evalúa la relación entre el valor pico de la señal y la raíz cuadrada de la media de los cuadrados de la señal, indica que el 14.84% de los datos están dentro del intervalo [1.04-1.06] para los datos normales [0], con una probabilidad del 12.79% para el mismo intervalo en los datos con fallo [1]. Esto sugiere una uniformidad en la señal, ya que exhibe un factor bajo.

Estadístico	Resultados
Factor de Impulso (Impulse Factor)	El factor compara el valor pico de las temperaturas con el valor medio de la señal, y este factor dentro del rango [1.04-1.06] posee una probabilidad del 14.61% en estado normal, mientras que para el rango [1.16-1.18], la probabilidad es del 13.95% en datos de fallo. El histograma indica que los datos se concentran principalmente en el rango de 17 a 18 °C, con una probabilidad del 21.78% para los datos normales [0], mientras que para el estado de fallo [1], la concentración se observa en el intervalo de temperatura de 22 a 23 °C, con un 19.76%.
Promedio (Mean)	La curtosis muestra una distribución de cola del 45.35% para los datos de fallo, en comparación con el 40% de los datos en condiciones normales.
Curtosis (Kurtosis)	Teniendo en cuenta que el SINAD mide la relación entre la señal, el ruido y la distorsión, podemos observar que una alta probabilidad en los puntos 0 y -2 sugiere una calidad de señal relativamente baja, esto se aplica para los dos estados.
Señal a Ruido y Distorsión (SINAD)	En esta relación se evidencia que la probabilidad de los datos está concentrada en un factor de 0 y 2. Por lo tanto, se interpreta que la señal no es óptima debido a su bajo factor.
Relación de Señal Ruido (SNR)	En esta distribución, se nota que la probabilidad de los datos se encuentra sesgada hacia la derecha y es positiva con respecto al 0. Por lo tanto, podemos concluir que representan temperaturas elevadas, tanto en condiciones normales como en estado de fallo.
Factor Forma (Shape Factor)	La distribución de los datos muestra una asimetría hacia la izquierda, siendo esta mucho más evidente en los datos de fallo, mientras que para los datos normales la distribución tiende a ser más simétrica.
Asimetría (Skewness)	La dispersión de probabilidad es del 17% en el estado de fallo, mientras que en el estado normal es del 13%. Se observa que los datos exhiben una dispersión hacia la izquierda, lo que significa que son predominantemente negativos en ambos estados, tanto normal como de fallo, representando aproximadamente el 12.79% del total.
Desviación Estándar (Std)	
Distorsión Armónica Total (THD)	

6.2.6 Características en el dominio del tiempo-frecuencia

Para analizar las características en el dominio tiempo-frecuencia, se ha elegido emplear los datos interpolados en conjunto con la Descomposición del Modo Empírico (DME), la cual se traduce en funciones de Modo Intrínseco (FMI). La **Figura 36** se presenta el histograma de las características tales como: distorsión armónica total, valor pico, factor de aclaramiento, factor de Liquidación, factor de cresta y energía. Consultar la **Tabla 5** que acompaña este análisis, así como las ecuaciones empleadas en el código para la obtención de estos histogramas.

Figura 36. Histogramas de las características en el dominio del tiempo-frecuencia.



Para el análisis de estas características se ha diseñado la **Tabla 12**, con los resultados de estas características.

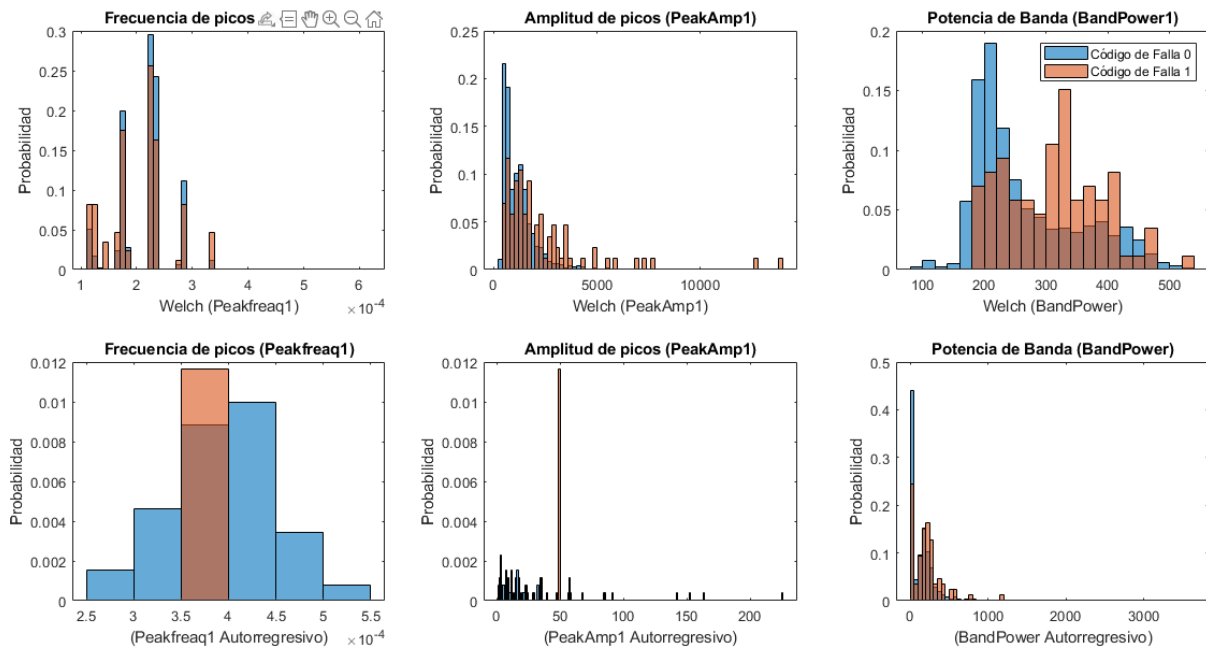
Tabla 12. Análisis de los resultados estadísticos en el dominio del tiempo-frecuencia.

Estadísticos	Resultados
Energía (EnergyIMF1)	La descomposición de la señal en un solo modo intrínseco IMF1 revela una distribución de datos que se concentra más hacia la derecha en el estado de fallo, mientras que, para los datos normales, la energía se distribuye de manera uniforme.
Valor pico (PeakValueIMF1)	Se observa que los datos se distribuyen hacia la derecha del 0 en ambos estados, lo que sugiere una mayor tendencia de la señal en la primera descomposición.
Factor de cresta (CrestFactorIMF1)	El factor de cresta para esta descomposición presenta una acumulación en 4 con una probabilidad de 16.27% para datos de falla y del 11.04% para datos de estado normal.
Factor de impulso (ImpulseFactorIMF1)	Para esta descomposición, los valores se distribuyen alrededor de 5 con una probabilidad del 15% y 12% para los estados de falla y normal, respectivamente.
Factor de aclaramiento (ClearanceFactorIMF1)	La distribución de datos para esta descomposición oscila entre 7, 8, 9 y 10, con una probabilidad del 13% para ambos estados, tanto normal como de fallo.

6.2.7 Características en el dominio de la frecuencia

Para analizar las características en el dominio de la frecuencia, se utilizan dos enfoques principales: el procesamiento de Welch, conocido como Método de Welch, y el modelo autorregresivo, denominado Modelo Autorregresivo. Las ecuaciones necesarias para estos cálculos se detallan en la **Tabla 3**. Además, se muestran los resultados de los histogramas correspondientes a cada una de estas características en la **Figura 37**.

Figura 37. Características en el dominio de la frecuencia.



En la **Tabla 13** se muestran los resultados correspondientes a cada una de las características en función de la frecuencia, comparando los dos métodos: Welch y Autorregresivo.

Tabla 13. Análisis de los resultados estadísticos en el dominio de la frecuencia.

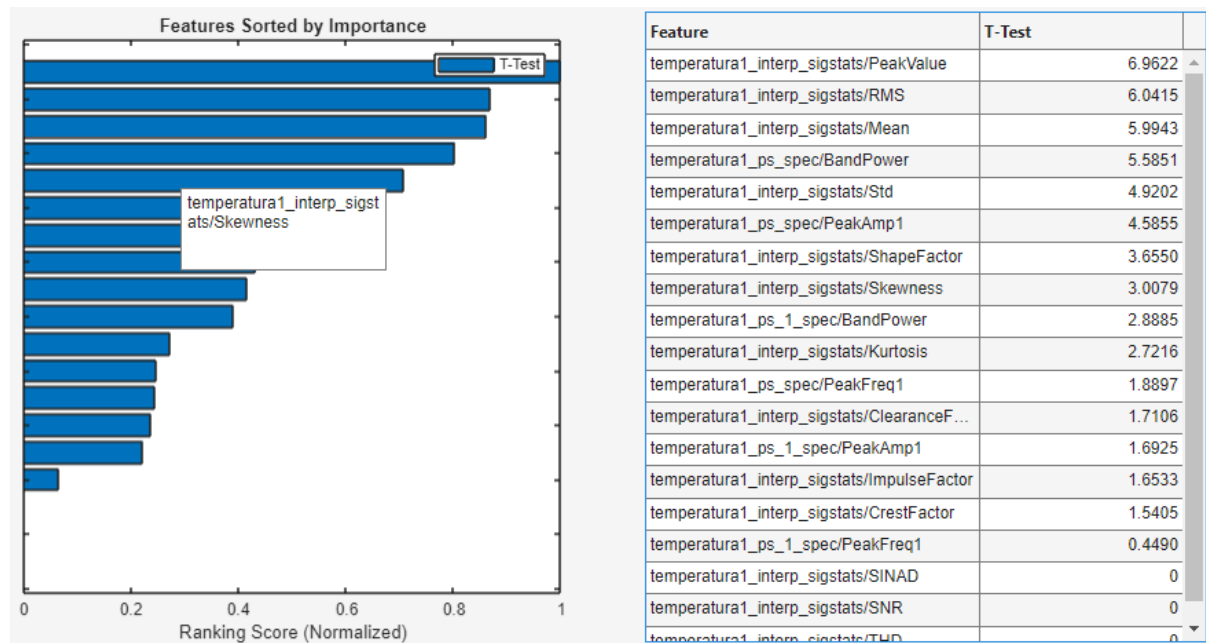
Funciones	Características
Frecuencia de picos (Peakfreak1)	Como es frecuencia de picos se puede ver la notable diferencia entre los dos métodos de Welch y Autorregresivo, y estos picos están muy dispersos para un análisis correcto.
Amplitud de picos (PeakAmp1)	Estos histogramas muestran una distribución desigual para los dos métodos de procesamiento, lo cual también se reflejará en las características.
Potencia de banda (BandPower1)	La potencia para los dos métodos es mucho más clara, significa que se puede encontrar alguna anomalía en la distribución que está alrededor de los 500.

6.2.8 Ranking de las características

Se seleccionaron las características más relevantes entre los dominios del tiempo, tiempo-frecuencia y frecuencia. En este análisis, que involucra datos de temperatura y dos estados de fallo en el convertidor del aerogenerador, se optó por un enfoque supervisado debido a sus dos clases [0, 1]. En la **Figura 38**, se presentan las características más significativas. La función utilizada para identificarlas es el método de T-test. Por ejemplo, la característica "valor pico" ($\text{temperatura1_interp_sigstats/PeakValue}$) encabeza la lista con el puntaje más alto en el

método de selección T-test. A continuación, se muestran las demás características calculadas en el programa.

Figura 38. Ranking de características: Análisis supervisado



6.2.9 Entrenar modelos

Se seleccionaron las cinco características más relevantes del análisis de Ranking Supervisado: (temperatura1_interp_sigstats/Mean, temperatura1_interp_sigstats/PeakValue, temperatura1_interp_sigstats/RMS, temperatura1_interp_emdfeat/PeakValueIMF1, temperatura1_ps_spec/BandPower). Estas características fueron exportadas al programa Classification Learner, donde se procedió a entrenar los modelos. Se dividió el conjunto de datos en una proporción del 30% para datos de prueba y el resto para entrenamiento. De todos estos modelos, el modelo de árbol obtuvo un resultado de precisión del **97.27%**. En la **Tabla 14**, se presentan algunos de los modelos seleccionados junto con su precisión calculada por el programa.

La precisión de los modelos dependerá de las características analizadas en DFD, así como de la cantidad de datos y del método de selección de las características.

Tabla 14. Resultados de los modelos entrenados en Classification Learner.

MODELOS	PRECISIÓN
Discriminante del subespacio	96,8%
SVM Kernel, Kernel de regresión logística, coseno KNN, basto KNN, gaussiano fino SVM, basto gaussiano SVM, Regresión Logística Binaria GLM,	96,7%
Árbol fino	96,5%
Árbol Medio	95,4%
Árbol basto, discriminante cuadrática	95,3%

6.3 Tercer Objetivo

6.3.1 Guías para interpretar resultados.

Para comprender los resultados de la aplicación DFD, se han diseñado ventanas que muestran la secuencia del proceso, detallando cada uno de los pasos junto con las funciones principales utilizadas en el programa. Además, se incluyen los algoritmos empleados a lo largo de todo el proceso. Se trata de ser lo más explícito posible para que el usuario pueda entender la herramienta.

6.3.2 Pasos para el preprocesamiento de datos (limpieza de datos)

Se ha desarrollado un código en Matlab para realizar la limpieza de datos, cuyos pasos se detallan en la **Figura 39** y **Figura 40**. Se emplean diversas funciones, entre las que se incluyen `sortrows`, `fillmissing` y `filloutliers`. Asimismo, se identificaron días no registrados por el sensor, los cuales son eliminados mediante el uso de las funciones `unique`, `dateshift`, `hitscounts`, `find` y `ismember` en Matlab.

Figura 39. Pasos para encontrar datos atípicos y faltantes.

Limpiar y Formatear Datos

Paso 1
Seleccionamos las variables importantes como: el número de torre, y variables de las 3 temperaturas del convertidor, se usa la función: *find()*.

Paso 2
Usamos la función: *fillmissing()*; la función se utiliza para rellenar los datos faltantes en los vectores de temperatura, utilizando la técnica de interpolación lineal.

Paso 3
Usamos la función: *filloutliers()*, para encontrar valores atípicos y sustituirlos por ejemplo: temperaturas en 0° y temperaturas superiores a los 30° se utiliza la técnica del valor previo junto con cuartiles.

Figura 40. Pasos en la construcción de la tabla de importación.

Limpiar y Formatear Datos

Paso 4
Se construye un algoritmo de bucle *for* que itere cada día, de esta manera se agrupa por miembro compuesto de una tabla de 144 datos. En la variable *timetable* se encuentran los datos en función del tiempo.

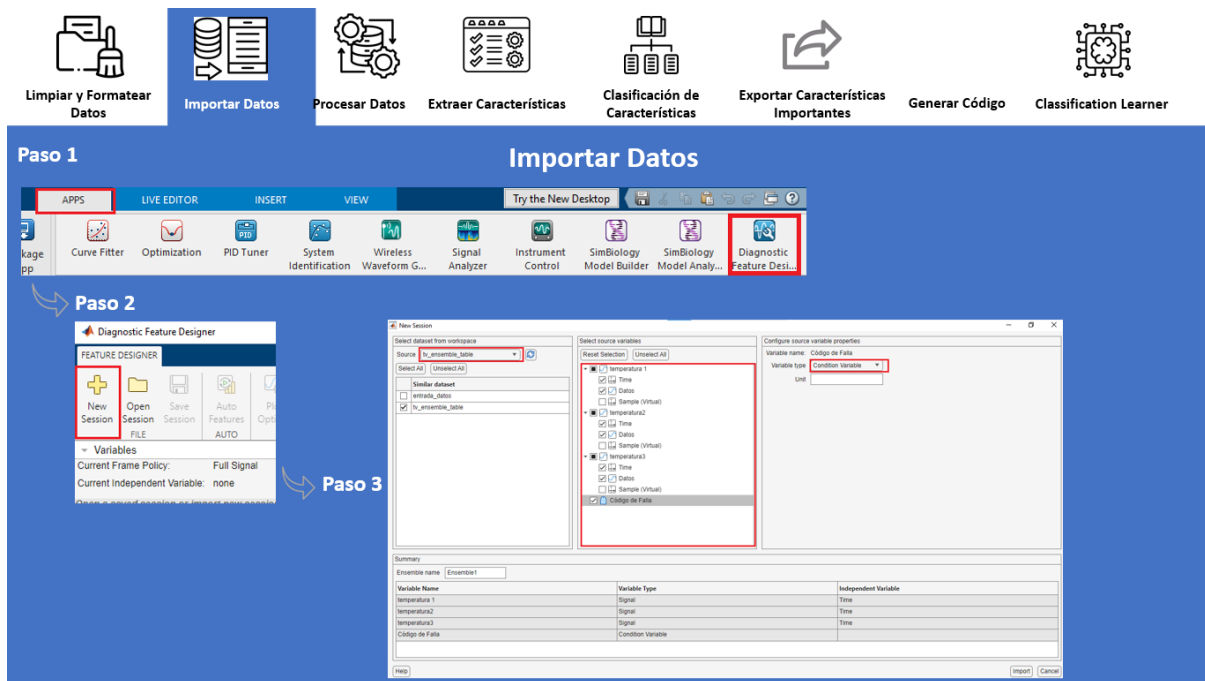
	1	2	3	4
	temperatura 1	temperatura2	temperatura3	Código de Falla
1	144x1 timetable	144x1 timetable	144x1 timetable	0
2	144x1 timetable	144x1 timetable	144x1 timetable	0
3	144x1 timetable	144x1 timetable	144x1 timetable	0
4	144x1 timetable	144x1 timetable	144x1 timetable	0
5	144x1 timetable	144x1 timetable	144x1 timetable	0
6	144x1 timetable	144x1 timetable	144x1 timetable	0
7	144x1 timetable	144x1 timetable	144x1 timetable	0
8	144x1 timetable	144x1 timetable	144x1 timetable	0
9	144x1 timetable	144x1 timetable	144x1 timetable	0
10	144x1 timetable	144x1 timetable	144x1 timetable	0
11	144x1 timetable	144x1 timetable	144x1 timetable	0
12	144x1 timetable	144x1 timetable	144x1 timetable	0

Time	1	2
00:00:00	18.4000	
00:10:00	18.4000	
00:20:00	18.3000	
00:30:00	18.3000	
00:40:00	18.3000	
00:50:00	18.3000	
01:00:00	18.3000	
01:10:00	18.3000	
01:20:00	18.3000	
01:30:00	18.3000	
01:40:00	18.2000	
01:50:00	18.2000	

6.3.3 Pasos para importar datos

Después de preparar la tabla, esta puede importarse a DFD. en la **Figura 41** se muestra la ventana con las diversas opciones que ofrece la interfaz. Se han extraído las temperaturas de la ventana de trabajo (*tv_ensemble_table*).

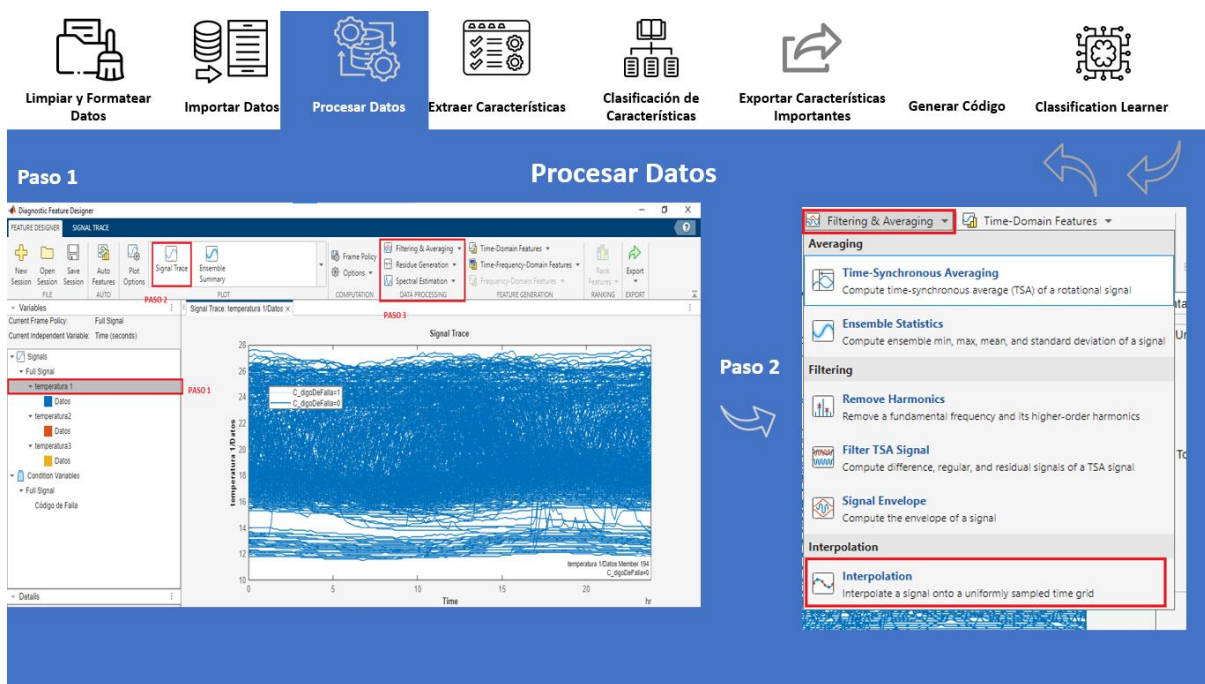
Figura 41. Pasos para importar la tabla de las temperaturas



6.3.4 Pasos en el procesamiento de datos

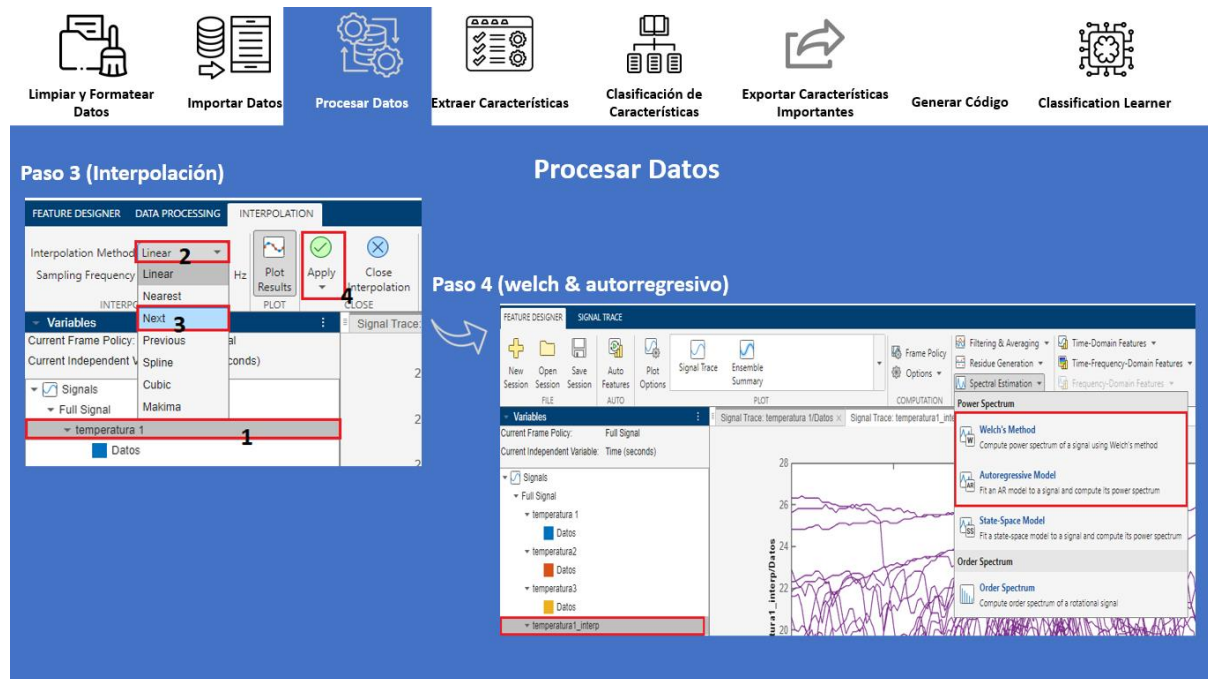
Los pasos a seguir en el procesamiento de datos o en el tratamiento de señales se pueden observar en la **Figura 42**. Existen varios métodos de tratamiento de señales en DFD, tales como la interpolación, el filtrado y el promediado de señales, entre otros. Para nuestro análisis, se han seleccionado la interpolación y los métodos de Welch y autorregresivo.

Figura 42. Pasos en la interpolación de los datos.



Se emplea la función interp1 junto con el método 'next' para la interpolación, mientras que para los métodos de Welch y Autorregresivo se utilizan las funciones welch y ar, respectivamente. Ver el **Anexo 4** y sigue los pasos descritos en la **Figura 43**.

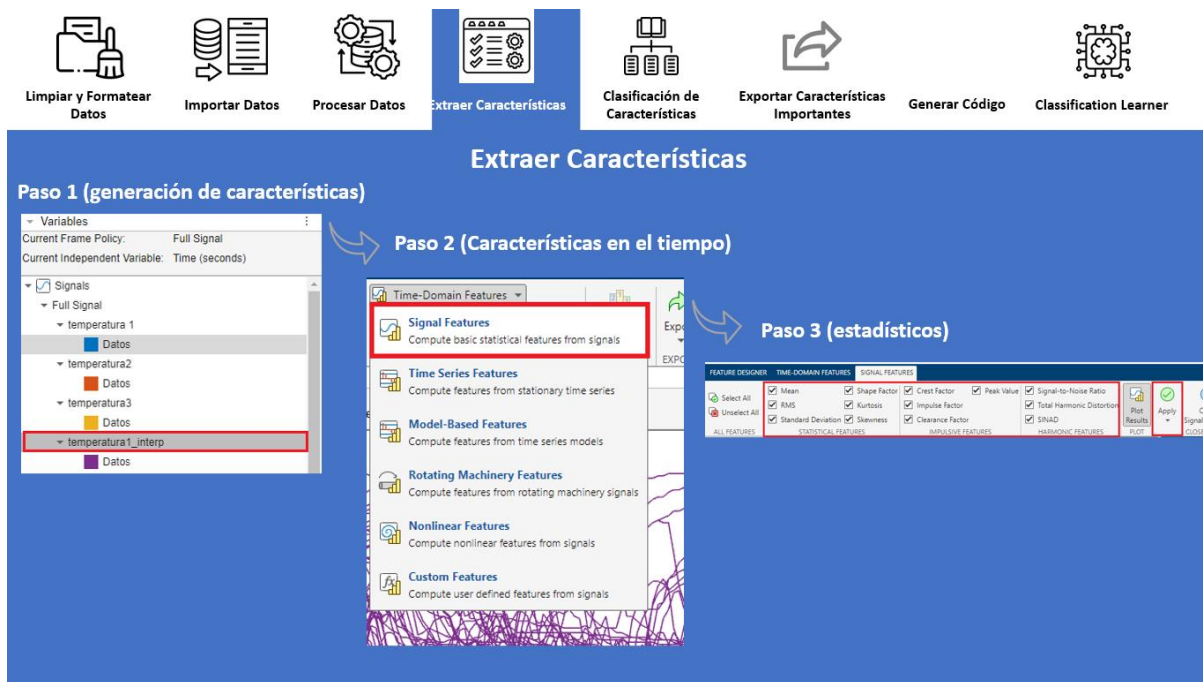
Figura 43. Pasos métodos Welch & Autorregresivo.



6.3.5 Pasos para extraer características (dominio en el tiempo)

Los pasos necesarios para obtener los indicadores de condición se detallan en la **Figura 44**, Se comienza seleccionando la variable procesada en el recuadro correspondiente, por ejemplo, (temperatura1_interp). Luego, se elige "Signal Features" en el recuadro mostrado y se seleccionan los estadísticos deseados. Estos se marcan en las pestañas correspondientes y, finalmente, se aplica la acción pulsando el botón "Aplicar" (Apply). Entre las funciones utilizadas por el programa se encuentran: funitconv, trapz, array2table, mean, max, rms, peakAmp, peakFreq, kurtosis, signad, snr, skewness, std y thd. Para más detalles, consulte el **Anexo 4**.

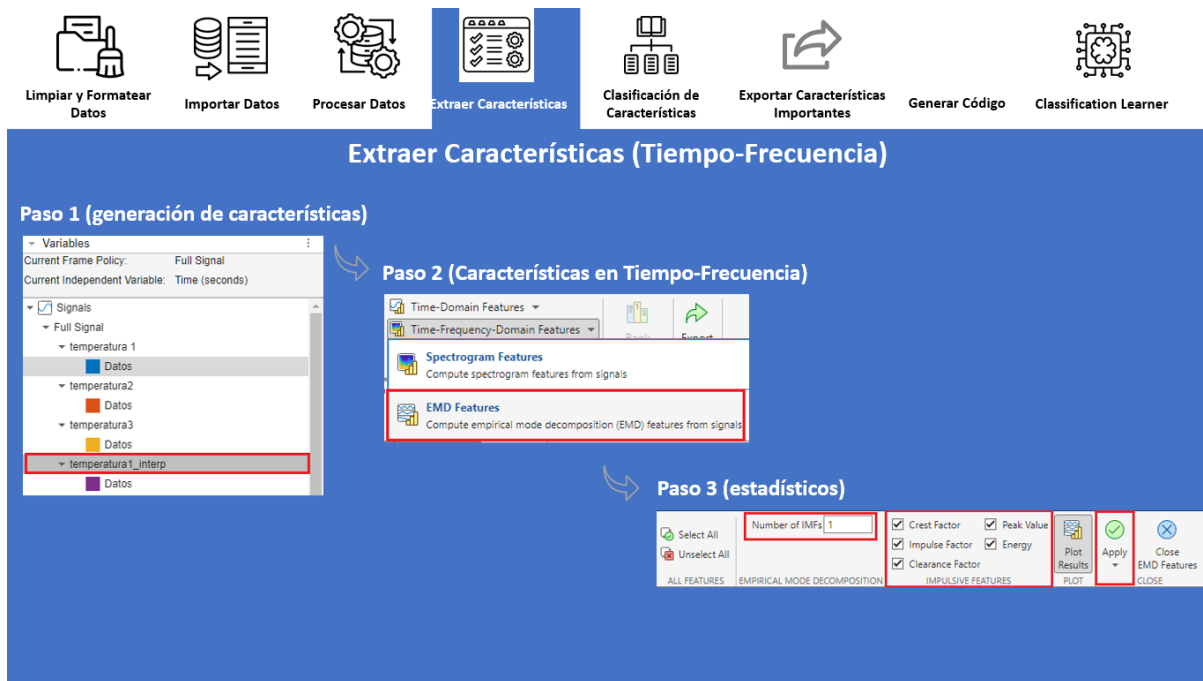
Figura 44. Pasos para encontrar las características en el dominio del tiempo



6.3.6 Pasos para extraer características (dominio en el tiempo-frecuencia)

Para obtener las Características en el Dominio del Tiempo-Frecuencia, seguimos los pasos detallados en la **Figura 45**. Utilizamos la misma señal procesada (temperatura1_interp). Luego, procedemos a la Descomposición Modal Empírica (EMD Features) y seleccionamos una descomposición. El usuario puede ajustar los parámetros según considere conveniente para el análisis. Posteriormente, hacemos clic en los recuadros correspondientes para marcar con pestañas los estadísticos seleccionados. Las funciones utilizadas por el programa para calcular estas características son las siguientes: emd, peak2rms, max y sum. Para más detalles, consulte el **Anexo 4**.

Figura 45. Pasos para obtener dominio tiempo-frecuencia.



6.3.7 Pasos para extraer características (dominio en la frecuencia)

Para obtener las características en el dominio de la frecuencia, se siguen los pasos detallados en la **Figura 46**. Además, se seleccionan las señales procesadas previamente con los métodos de Welch y Autorregresivo. Luego, se eligen las opciones para obtener características en la barra desplegable. En este estudio, se optó por las características espectrales (Spectral Features). Finalmente, se seleccionan los estadísticos deseados en los recuadros correspondientes. Las funciones utilizadas incluyen findpeaks, peakAmb, funitconv y trapz, las cuales se aplican a ambas señales espectrales. Para más detalles, consulte el **Anexo 4**.

Figura 46. Pasos para obtener las características en el dominio de la frecuencia

6.3.8 Selección de características (ranking features).

Para la clasificación de características, se siguen los pasos detallados en la **Figura 47**, Se debe elegir entre la selección supervisada y no supervisada. Dado que se trata de dos clases, se optó por el método supervisado (T-test). La función utilizada por el programa es rankfeatures con el método ttest. El código correspondiente se puede encontrar en el **Anexo 5**. La calidad de las características seleccionadas dependerá del usuario.

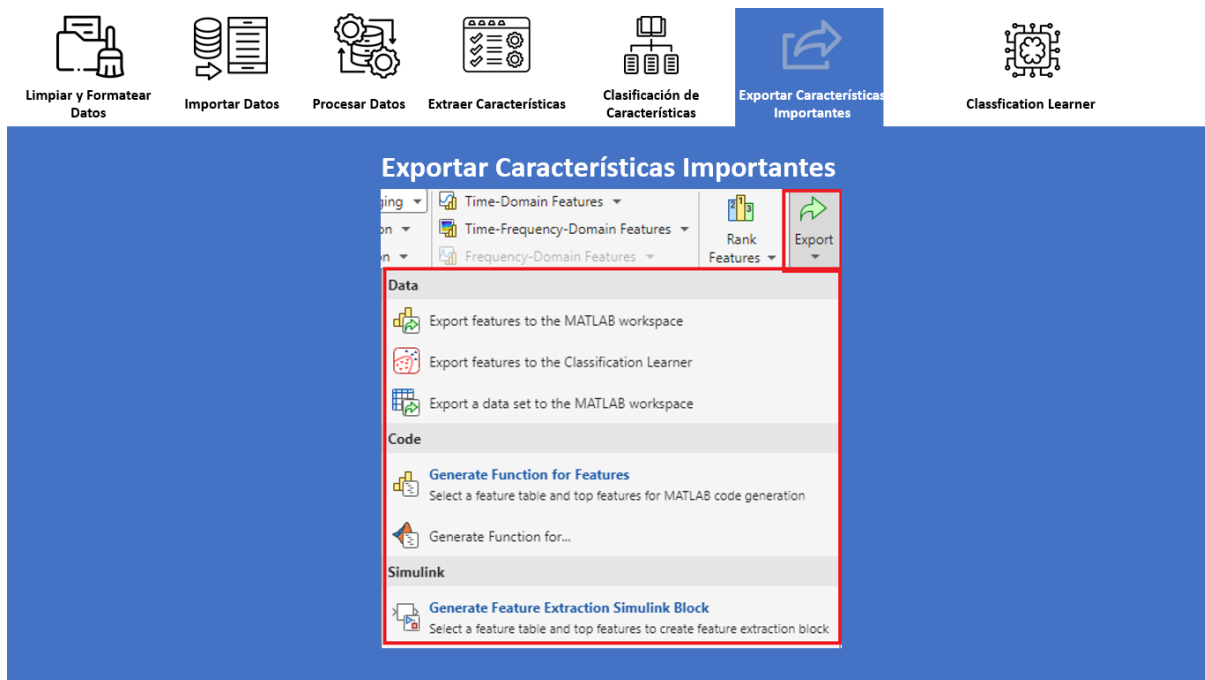
Figura 47. Método de selección de características.

Feature	T-Test
temperatura1_interp_sigstats/PeakValue	6.9622
temperatura1_interp_sigstats/RMS	6.0415
temperatura1_interp_sigstats/Mean	5.9943
temperatura1_ps_spec/BandPower	5.5851
temperatura1_interp_emdfeat/PeakValueIMF1	5.0382
temperatura1_interp_sigstats/Std	4.9202
temperatura1_ps_spec/PeakAmp1	4.5855
temperatura1_interp_sigstats/ShapeFactor	3.6550
temperatura1_interp_sigstats/Skewness	3.0079
temperatura1_ps_1_spec/BandPower	2.8885
temperatura1_interp_sigstats/Kurtosis	2.7216
temperatura1_ps_spec/PeakFreq1	1.8897
temperatura1_interp_emdfeat/EnergyIMF1	1.7768
temperatura1_interp_sigstats/ClearanceFactor	1.7106
temperatura1_ps_1_spec/PeakAmp1	1.6925
temperatura1_interp_sigstats/ImpulseFactor	1.6533
temperatura1_interp_sigstats/CrestFactor	1.5405
temperatura1_interp_emdfeat/ClearanceFact...	0.4789

6.3.9 Exportar características importantes

Al presionar sobre el botón "Exportar" (Export), podemos generar código a través del programa y llevarlo a la ventana de trabajo de Matlab. También se ofrece la opción de seleccionar una tabla con las características más importantes según considere el usuario. En la **Figura 48** se muestran las diferentes formas de exportar las características.

Figura 48. Ventana de exportación de las características seleccionadas.



6.3.10 Programa (classification learner)

Para entrenar los modelos, es necesario utilizar el programa CLASIFICATION LEARNER. Una vez que se cuenta con el código o la tabla de características, se procede en Matlab seleccionando CLASIFICATION LEARNER desde la sección de aplicaciones (APPS). Esto abrirá una ventana en la que se elige la opción "From Workspace" para importar los datos. Posteriormente, se selecciona la variable de respuesta, que en este caso es 0 y 1. Finalmente, se ejecuta esta sección seleccionando "Start Session". Los pasos para importar las características importantes se muestran en la **Figura 49**.

Una vez dentro del programa, podemos seleccionar los modelos que deseamos entrenar. Al hacer clic en el botón "Todo" (All), seleccionamos todos los modelos disponibles. En esta sección, se entrenan todos los modelos y podemos observar la precisión de cada uno en porcentaje. Para seguir estos pasos, consulte la **Figura 50**.

Figura 49. Pasos para importar las características a classification learner

Paso 1

Classification Learner

Paso 2

Paso 3 (Ventana de Importe Características)

Classification Learner - untitled*

LEARN TEST EXPLAIN

From Workspace
Start a new session by selecting data from the workspace and specifying a validation scheme

From File
Start a new session by importing data from a file and specifying a validation scheme

New Session from Workspace

Data set: 2010C11_10K14

Response: From data set variable: 00004

Feature	Type	Range
temperature_1_inner_1_0_1	double	12.7 - 20.9
temperature_1_inner_1_0_2	double	11.0665 - 20.5568
temperature_1_inner_1_0_3	double	11.0572 - 20.5548
temperature_1_inner_1_0_4	double	62.2011 - 552.884
temperature_1_inner_1_0_5	double	0.0205427 - 19.5206
temperature_1_inner_1_0_6	double	0.344724 - 3.93989
temperature_1_inner_1_0_7	double	1 - 1.62586
temperature_1_inner_1_0_8	double	4.68865 - 4.62364
temperature_1_inner_1_0_9	double	0.00010719 - 3620.97
temperature_1_inner_1_0_10	double	1.02281 - 10.2281

Validation Scheme: Cross Validation

Test: Start Session

Figura 50 Entrenamiento de modelos con su Precisión.

Clasificación Learner

Paso 1

Classification Learner - untitled*

LEARN TEST EXPLAIN

Train All

Paso 2

Model	Accuracy (Validation)	Features
2.15 SVM	96.8%	10/10 features
2.16 SVM	96.8%	10/10 features
2.17 KNN	94.1%	10/10 features
2.18 KNN	96.8%	10/10 features
2.19 KNN	96.8%	10/10 features
2.20 KNN	96.8%	10/10 features
2.21 KNN	96.8%	10/10 features
2.22 KNN	96.7%	10/10 features
2.23 Ensemble	96.8%	10/10 features
2.24 Ensemble	96.7%	10/10 features

Model 2.18

True Class \ Predicted Class	0	1
0	96.9%	
1	3.2%	100.0%

PPV: 96.9%, FDR: 3.2%

7. Discusión

En comparación con las búsquedas realizadas en la documentación, específicamente en Google Scholar, se encuentra que seguir la documentación proporcionada por MathWorks es mucho más conveniente. Esta última ofrece una cantidad sustancial de información detallada sobre su herramienta, lo cual se presenta como una ventaja significativa en comparación con la limitada información disponible en los artículos académicos. Esto resulta especialmente útil para comprender la funcionalidad específica de la herramienta.

Por lo expuesto de (Martínez, 2007, p. 17), donde se subraya la importancia de revisar los datos en busca de valores faltantes, atípicos y para organizar adecuadamente el conjunto de datos, se puede afirmar que los datos recopilados de los aerogeneradores cumplen con estas características. Se ha enfocado el análisis en un solo aerogenerador debido a restricciones computacionales y se han seleccionado variables de temperatura junto con los códigos de falla. Estos datos fueron preparados externamente a la aplicación, eligiendo las temperaturas del convertidor, motivado por el hecho de que en el diagrama de pastel estas temperaturas presentaron el mayor porcentaje de fallos, alrededor del 33%. Este proceso puede considerarse como una etapa de preprocesamiento, que incluyó tareas de limpieza y detección de datos atípicos. Durante la limpieza, no se encontraron datos de tipo NaN, sin embargo, se identificaron valores atípicos, como temperaturas extremadamente bajas (0°C) y extremadamente altas (superiores a 30° y 40°C). Además, antes de importar los datos, se realizó un ensamblaje necesario para garantizar un manejo adecuado de los mismos en el programa.

Según lo que menciona Mathworks (2011, p. 5), el mantenimiento predictivo ofrece la posibilidad de preprocesar los datos, organizarlos y encontrar características que permitan detectar fallos en las máquinas. En esta investigación, centrada en los aerogeneradores y llevada a cabo mediante (DFD), se realizó un proceso iterativo en busca de mejorar el modelo. Inicialmente, se obtuvo un modelo de árbol con una precisión del 97.27%. Sin embargo, al entrenar el modelo en diferentes ocasiones, se observaron variaciones en estos porcentajes debido a la calidad del procesamiento, esto tiene que ver mucho con a la naturaleza de los datos. Dentro del programa se trabajó con una sola temperatura, porque las 3 temperaturas son similares, además se facilita el uso de memoria al trabajar tan solo con una temperatura, con la señal interpolada se pudo prácticamente completar datos de tiempo que el sensor por alguna razón no pudo capturar, se eliminó días menores a 10 datos por fila, porque son muy pocos a la hora de obtener señales espectrales, en DFD existen muchas otras formas de procesar las

señales, pero en esta investigación se optó por utilizar la interpolación junto con los métodos de Welch y Autorregresivo.

Los indicadores de estado desempeñan un papel fundamental en la distinción entre el funcionamiento normal y defectuoso, tal como lo señala MathWorks (2019, p. 2), Esto se evidencia en los resultados de los indicadores de condición del convertidor, los cuales se muestran en la distribución de los datos a través de histogramas. Es notable observar que, especialmente en el dominio del tiempo y en los valores pico de temperatura, como el "peakvalue", las probabilidades en la distribución de los datos son bastante similares. A pesar de ello, se puede observar que el estado normal tiene una acumulación ligeramente mayor.

Durante la selección de características, siguiendo las recomendaciones de (MathWorks, 2019b, 2022, 2023c), se encuentran disponibles varios métodos para la clasificación. Sin embargo, se sugiere optar por la clasificación supervisada cuando se trata de distinguir entre dos estados. En nuestro análisis, hemos empleado la técnica de selección supervisada T-test, que proporciona una puntuación en orden descendente. Además, se ha observado que los indicadores RMS y el promedio también juegan un papel crucial en este método. Esto se relaciona con la manera en que la técnica evalúa cada indicador, permitiendo una comparación más precisa en relación con el estado del sistema.

En última instancia, lo que realmente se busca es un modelo que pueda ser entrenado y que, lógicamente, ofrezca una precisión considerable. De hecho, la tarea de desarrollar un modelo de este tipo puede resultar complicada y difícil de programar. Sin embargo, Matlab ofrece una herramienta que nos facilita esta tarea: CLASIFICATION LEARNER, como se menciona en un video de (MathWorks, 2021), Lo importante es evaluar la precisión de nuestros indicadores de condición seleccionados en el DFD. Con esta herramienta, pudimos entrenar rápidamente el modelo y, al utilizar la matriz de confusión, alcanzamos una precisión del 96.87%. La idea es continuar mejorando la precisión de los modelos a medida que se hace retroalimentación.

El manual elaborado tiene como objetivo guiar al usuario a través de los pasos necesarios para obtener resultados y comprender algunas de las funcionalidades que ofrece el (DFD). Esta guía no solo se ha desarrollado para facilitar el entendimiento de DFD, sino que también forma parte integral de la investigación, con el propósito de comprender el mantenimiento predictivo en aerogeneradores. Durante el proceso de diseño del manual, se ha buscado crear un gráfico secuencial que detalle los pasos realizados en el proceso. Se ha puesto especial énfasis en que esta representación sea comprensible para el usuario, asegurando así una experiencia clara y fluida al seguir las instrucciones proporcionadas.

8. Conclusiones

- La documentación proporcionada por MathWorks, junto con los videos, tutoriales y revistas disponibles, es de gran ayuda para comprender las funcionalidades de DFD. Los ejemplos proporcionados por la empresa son indispensables para la codificación y adaptación a los requerimientos de diagnóstico de fallos en los aerogeneradores.
- Mediante la selección y el procesamiento adecuado de los datos, se puede comprender mejor las señales. Al elegir los datos de temperatura del convertidor, se pueden observar ciertos picos, como temperaturas que caen a 0 °C y valores superiores a 40 °C. Al sustituir estos valores mediante la técnica de llenado con el valor previo (previous), se logra mantener la integridad de los datos en lugar de simplemente eliminarlos.
- En el análisis de datos, siempre existirán datos faltantes. En este análisis de un aerogenerador, no se encontraron datos del tipo NaN, pero sí hubo fechas en las que el sensor no pudo capturar datos. Las funciones de eliminación con restricciones fueron de gran ayuda para corregir y evitar errores en el procesamiento. En este caso, con una frecuencia de muestreo de 10 minutos, se esperaban 144 datos por día. Sin embargo, no todos los días presentaron datos completos. La interpolación permitió corregir en parte esta problemática, asegurando la integridad y continuidad de los datos.
- La selección de los métodos de procesamiento de señales espectrales se basó en el propósito general, esto quiere decir que no se tomaron en cuenta otro tipo de procesamiento porque la mayoría es para maquinaria rotativa como: engranajes, cojinetes o máquinas que tengan movimiento circular, nuestro análisis es en base a temperaturas de circuitos electrónicos de potencia, los mismo aplica para las características en el dominio del tiempo y tiempo-frecuencia.
- La selección supervisada es ideal para las dos clases de 0 y 1, al elegir el método T-test se hace decisiones de prueba para dos muestras, entonces de esta manera clasifica las mejores características que se utilizarán para el entrenamiento de los modelos.
- De todos los modelos que se entrenaron el modelo de árbol alcanzó el 97.27 %, entonces esto da entender que es muy confiable en la predicción, con esto ya se podría sacar la vida útil, pero en esta investigación solo se tomó en cuenta la utilidad de DFD.

9. Recomendaciones

- Es necesario tener algo de conocimiento de lo que es un mantenimiento predictivo y estadística. Solo de esta manera se podrá comprender de la mejor manera los temas tratados en este proceso de investigación. También conocer algo sobre análisis de señales acompañado del dominio del tiempo, tiempo-frecuencia y frecuencia.
- Es importante no importar al programa datos en bruto sin hacer una limpieza y selección previa, esto puede ocasionar problemas. En el análisis de datos, sobre todo en frecuencias, las transformaciones se las hace con la transformada de Fourier y las ventanas están diseñadas de acuerdo al periodo de los datos.
- No es estrictamente necesario que la selección de nuestro ensamble sea por días dependiendo de lo que se tiene, se puede optar por meses, minutos, segundos, o como se quiera clasificar los datos de entrada, al final lo que se desea es tener la legibilidad de los datos y tener la estructura definida.
- Es necesario considerar el número de picos al obtener las características en el dominio de la frecuencia, el usuario las puede modificar de acuerdo a la prueba error que se realice en el estudio de las señales.
- La precisión de los modelos puede cambiar considerablemente, entonces no se debe caer en la confusión de los porcentajes mostrados, porque los modelos dependen de la cantidad de datos que se tengan disponibles para el entrenamiento y de las características seleccionadas.

10. Bibliografía

- Goldwind. (2008). *Manual de funcionamiento del sistema generador de turbinas eólicas serie Goldwind de 1500 kW*. www.goldwind.cn
- González, J. A. C., Pérez, R. C., Santos, A. C., & Gil, M. A. C. (2009). *Centrales de energías renovables*.
- Martínez, C. (2007). “*Uso de las Técnicas de Preprocesamiento de Datos e Inteligencia Artificial (Lógica Difusa) en la Clasificación / Predicción del Riesgo Bancario*” *Caso de Estudio : La Banca Comercial*.
- Mathworks. (2011). Mantenimiento predictivo. In *Redes de Ingeniería* (Vol. 2, Issue 1). <https://doi.org/10.14483/2248762x.7176>
- Mathworks. (2018). *Mastering Machine Learning A Step-by-Step Guide with MATLAB*. 22.
- MathWorks. (2019a). *Diagnostic Feature Designer*. <https://la.mathworks.com/help/predmaint/ref/featuredesigner-app.html>
- MathWorks. (2019b). *Diagnostic Feature Designer*. MathWorks. https://la.mathworks.com/help/predmaint/ref/featuredesigner-app.html#mw_6fee5e96-2094-4072-b579-27367d522391
- MathWorks. (2019c). *Diagnostic Feature Designer*. <https://la.mathworks.com/help/predmaint/ref/featuredesigner-app.html>
- MathWorks. (2019d). *fillmissing*. MathWorks. <https://la.mathworks.com/help/matlab/ref/fillmissing.html#bvc6et8>
- MathWorks. (2019e). *Import and Visualize Ensemble Data in Diagnostic Feature Designer*.
- MathWorks. (2019f). *Importar Datos a Diagnostic Feature Designer*. MathWorks. <https://la.mathworks.com/help/predmaint/ug/import-data-into-diagnostic-feature-designer.html>
- MathWorks. (2019g). *Limpiar datos confusos y localizar extremos usando las tareas de Live Editor*. https://la.mathworks.com/help/matlab/data_analysis/cleandatawithliveeditortasks.html
- MathWorks. (2019h). *Limpiar Datos Faltantes*. MathWorks. <https://la.mathworks.com/help/matlab/ref/cleanmissingdata.html>
- Mathworks. (2019). *Predictive Maintenance: Extracting Condition Indicators with MATLAB*.
- MathWorks. (2019i, April 23). *Cómo usar la aplicación “Diagnostic Feature Designer” para la extracción de características*. MATLAB.

<https://www.youtube.com/watch?v=oDd7aEmRNpI&list=PLFWOKW0A15QSOntQ1Ughh9GwjH9ONr6A8&index=3&t=2s>

MathWorks. (2021). *Mantenimiento predictivo con MATLAB: un enfoque basado en datos.*

https://la.mathworks.com/support/search.html/videos/predictive-maintenance-with-matlab-a-data-based-approach-1637776205693.html?fq%5B%5D=asset_type_name:video&fq%5B%5D=category:predmaint/index&page=1

MathWorks. (2022, April 5). *Mantenimiento predictivo con MATLAB: un enfoque basado en datos.* MATLAB En Español.

<https://www.youtube.com/watch?v=6t913oHwiu4&list=PLFWOKW0A15QSOntQ1Ughh9GwjH9ONr6A8&index=2&t=1077s>

MathWorks. (2023a). *Classification Learner.*

<https://la.mathworks.com/help/stats/classificationlearner-app.html>

MathWorks. (2023b). *Organizar datos del sistema para DIAGNOSTIC FEATURE*

DESIGNER. MathWorks. <https://la.mathworks.com/help/predmaint/ug/organize-system-data-for-diagnostic-feature-designer.html>

MathWorks. (2023c, October 12). *Identificación de fallas del motor mediante el aprendizaje automático para el mantenimiento predictivo.* Matlab.

<https://www.youtube.com/watch?v=JwZ5ffZk-fM&t=607s>

MathWorks. (2023d, November 8). *Explore datos de conjunto y compare funciones mediante el Diseñador de funciones de diagnóstico.*

Singh, R. R., Bhatti, G., Kalel, D., Vairavasundaram, I., & Alsaif, F. (2023). Building a Digital Twin Powered Intelligent Predictive Maintenance System for Industrial AC Machines. *Machines*, 11(8). <https://doi.org/10.3390/machines11080796>

Yule, L., Harris, N., Hill, M., Zaghari, B., & Grundy, J. (2024). Temperature Hotspot Detection on Printed Circuit Boards (PCBs) Using Ultrasonic Guided Waves—A Machine Learning Approach. *Sensors*, 24(4). <https://doi.org/10.3390/s24041081>

Zhao, S., Blaabjerg, F., & Wang, H. (2021). An overview of artificial intelligence applications for power electronics. In *IEEE Transactions on Power Electronics* (Vol. 36, Issue 4, pp. 4633–4658). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/TPEL.2020.3024914>

11. Anexos

Anexo 1. Datos de los sensores de los aerogeneradores.

	VarName1	timestamp	wind_speed_avg	wind_speed_max	wind_speed_min	grid_active_power_avg	grid_active_power_max	grid_active_power_min	generator_
1	0	2017-04-16 00:...	12.0800	15.7500	6.9900	1.0405e+03	1.4751e+03	736.7000	
2	1	2017-04-16 00:...	11.4600	15.3200	6.3600	929.7700	1.3257e+03	580.4800	
3	2	2017-04-16 00:...	11.9000	16.6200	7.5400	924.6600	1.3445e+03	548.8900	
4	3	2017-04-16 00:...	11.6800	15.0700	6.8100	937.6300	1.2506e+03	614.6300	
5	4	2017-04-16 00:...	10.5500	13.5200	6.9900	821.7000	1.2737e+03	582.1900	
6	5	2017-04-16 00:...	10.4000	13.8700	7.2900	743.3700	997.0600	491.7000	
7	6	2017-04-16 01:...	11.5200	14.7000	7.0400	788.9400	1.1635e+03	513.0400	
8	7	2017-04-16 01:...	12.0400	14.6700	8.8100	970.6700	1.2498e+03	712.8000	
9	8	2017-04-16 01:...	12.8600	15.8700	8.1600	1.1292e+03	1.4222e+03	821.2100	
10	9	2017-04-16 01:...	13.7900	18.3000	9.2600	1.2945e+03	1.5562e+03	949.2600	
11	10	2017-04-16 01:...	14.2400	18	8.9100	1.3447e+03	1.5588e+03	964.6200	
12	11	2017-04-16 01:...	13.9700	18.5200	8.1100	1.3675e+03	1.5733e+03	997.0600	
13	12	2017-04-16 02:...	14.4800	19.4000	9.1900	1.4148e+03	1.5690e+03	1.0048e+03	
14	13	2017-04-16 02:...	14.9700	18.1500	10.2200	1.4315e+03	1.5665e+03	857.9200	
15	14	2017-04-16 02:...	15.2000	19.2000	9.2900	1.3991e+03	1.5690e+03	979.9900	
16	15	2017-04-16 02:...	14.6800	18.3700	9.0400	1.3432e+03	1.5665e+03	986.8200	
17	16	2017-04-16 02:...	13.4000	18.1000	7.9600	1.1878e+03	1.5571e+03	802.4300	
18	17	2017-04-16 02:...	13.1400	17.0500	9.0600	1.1818e+03	1.5579e+03	828.8900	
19	18	2017-04-16 03:...	14.0300	17.9200	8.9100	1.2799e+03	1.5690e+03	845.9700	
20	19	2017-04-16 03:...	12.6400	18.1200	4.9100	1.1699e+03	1.5699e+03	705.9700	
21	20	2017-04-16 03:...	13.1000	16.7500	6.6100	1.1804e+03	1.5451e+03	729.0200	
22	21	2017-04-16 03:...	12.6600	17.2500	4.8100	1.1465e+03	1.5656e+03	664.1400	
23	22	2017-04-16 03:...	13.7300	17.9700	6.5400	1.2745e+03	1.5682e+03	836.5800	
24	23	2017-04-16 03:...	14.0100	19	7.6600	1.3039e+03	1.5699e+03	711.0900	
25	24	2017-04-16 04:...	13.3900	17.5200	8.4100	1.2389e+03	1.5673e+03	696.5800	
26	25	2017-04-16 04:...	14.3800	17.2500	9.8400	1.2905e+03	1.5622e+03	956.9400	
27	26	2017-04-16 04:...	14.7500	19.8000	9.1400	1.3644e+03	1.5673e+03	898.8900	
28	27	2017-04-16 04:...	15.0900	18.9200	9.7400	1.3806e+03	1.5656e+03	997.0600	
29	28	2017-04-16 04:...	15.7600	19.4000	10.7900	1.4485e+03	1.5724e+03	928.7700	
30	29	2017-04-16 04:...	15.9700	20.1800	10.0200	1.4807e+03	1.5716e+03	853.6500	

Anexo 2. Código preprocesamiento de datos.

```
1  Preprocesamiento de los Datos Torre 1 Aerogenerador
2  clc;clear
3  load TORRE_1.mat
4  TORRE_1;
5  Torre=TORRE_1(:,[2 49 50 51 55]);
6  Torre=sortrows(Torre) %Ordenar filas de matriz o tabla.
7
8  Limpiar Datos Faltantes
9  %Completar las entradas que faltan
10 [T1,missingIndices] = fillmissing(Torre,"linear",...
11     "DataVariables",["pitch_converter_temperature_1_max",...
12     "pitch_converter_temperature_2_max","pitch_converter_temperature_3_max"
13 ]);
14 ];
15
16 %Mostrar resultados
17 clf
18 plot(T1.timestamp,T1.pitch_converter_temperature_1_max,"Color",[0 114
19 189]/255,...
20     "LineWidth",1.5,"DisplayName","Cleaned data")
21 hold on
22
23 plot(T1.timestamp(missingIndices(:,2)),T1.pitch_converter_temperature_1
24 _max(missingIndices(:,2)),".",...
25     "MarkerSize",12,"Color",[217 83 25]/255,...
26     "DisplayName","Filled missing entries")
27 title("Número de entradas completas que faltan (TORRE 1): " +
28 nnz(missingIndices(:,2)))
29
30 hold off
31 legend('Datos Limpios')
32 ylabel("pitch converter temperature 1 max")
33 xlabel('TIEMPO')
34 clear missingIndices
35
36 Limpiar Datos Atípicos
37 % Rellenar Datos Atípicos
38 [T1_1,outlierIndices,thresholdLow,thresholdHigh] =
39 filloutliers(T1,"previous","quartiles","ThresholdFactor",3,...
40     "DataVariables",["pitch_converter_temperature_1_max",...
41     "pitch_converter_temperature_2_max","pitch_converter_temperature_3_max"
42 ]);
43 ];
44
45 % Mostrar Resultados
46 clf
47
48 plot(T1.timestamp,T1.pitch_converter_temperature_1_max,"Color",[77 190
49 238]/255,...
50     "DisplayName","Datos de Entrada")
51
52 hold on
53 plot(T1.timestamp,T1_1.pitch_converter_temperature_1_max,"Color",[0 114
54 189]/255,...
```

```

55     "LineWidth",1.5,"DisplayName","Datos Limpios")
56
57 % Plot outliers
58 plot(T1.timestamp(outlierIndices(:,2)),T1.pitch_converter_temperature_1
59 _max(outlierIndices(:,2)),...
60     "x","Color",[64 64 64]/255,"DisplayName","Valores Atípicos")
61 title("Numero de Valores Atípicos (TORRE 1): " +
62 nnz(outlierIndices(:,2)))
63
64 % Plot filled outliers
65 plot(T1.timestamp(outlierIndices(:,2)),T1_1.pitch_converter_temperature
66 _1_max(outlierIndices(:,2)),".",...
67     "MarkerSize",12,"Color",[217 83 25]/255,"DisplayName","Valores
68 Atípicos Reemplazados")
69 % Plot outlier thresholds
70 plot([xlim missing xlim],...
71     [thresholdLow.pitch_converter_temperature_1_max*[1 1] NaN
72 thresholdHigh.pitch_converter_temperature_1_max*[1 1]],...
73     "Color",[145 145 145]/255,"DisplayName","Umbrales Atípicos")
74
75 hold off
76 legend
77 xlabel('Tiempo')
78 ylabel("pitch converter temperature 1 max (°C)")
79 clear outlierIndices thresholdLow thresholdHigh
80
81 plot(T1_1.timestamp,T1_1.pitch_converter_temperature_1_max)
82 title('Datos faltantes y Atipicos corregidos (TORRE 1)')
83 xlabel('Tiempo')
84 ylabel("pitch converter temperature 1 max (°C)")

```


Anexo 3. Designación de código (0-1) Fault Query para el convertidor.

```
1 DESIGNACIÓN DE CÓDIGO 0-1 FAULT QUERY PARA EL CONVERTIDOR
2 for i=1:size(selec_tor1.FaultDescription)
3     if selec_tor1.FaultDescription(i) == "(435)Error_converter not
4 ready" | selec_tor1.FaultDescription(i)=="(451)Error_converter DC link
5 voltage high" | selec_tor1.FaultDescription(i)=="(447)Error_converter
6 grid side IGBT over current"...
7         | selec_tor1.FaultDescription(i)=="(442)Error_converter
8 IGBT ok signal loss" |
9 selec_tor1.FaultDescription(i)=="(450)Error_converter DC link voltage
10 low" | selec_tor1.FaultDescription(i)=="(449)Error_converter chopper
11 over current"...
12         | selec_tor1.FaultDescription(i)=="(446)Error_converter
13 step up circuit over current" |
14 selec_tor1.FaultDescription(i)=="(420)Error_converter IGBT cooling 1#
15 fan feedback
16 loss"|selec_tor1.FaultDescription(i)=="(451)Error_converter DC link
17 voltage high"...
18         | selec_tor1.FaultDescription(i)=="(448)Error_converter
19 grid side voltage high"
20     codigo(i)=1;
21     else
22         codigo(i)=0;
23     end
24 end
25 codigo=categorical(codigo');
26 t=array2table(codigo,"VariableNames",{ 'Codigo_Fallo' });
27 selec_tor1=[selec_tor1 t];
28
29 %dat2recor=selec_tor1(find(selec_tor1.FaultStartTime>fecha &
30 selec_tor1.FaultStartTime<fecha_fin),:)
31
32 figure
33 escalera1=[];
34 escalera2=[];
35 % Iterar a través de los intervalos de fallo y construir las
36 coordenadas
37 for i = 1:length(selec_tor1.FaultStartTime)
38     escalera1 = [escalera1,
39 selec_tor1.FaultStartTime(i),selec_tor1.FaultEndTime(i)];
40     escalera2 = [escalera2, selec_tor1.Codigo_Fallo(i),
41 selec_tor1.Codigo_Fallo(i)];
42 end
43 % Graficar las escaleras
44 stairs(escalera1,escalera2, 'Color', 'black', 'LineWidth', 2);
45 % Configurar ejes y etiquetas
46 xlabel('Tiempo')
47 ylabel('Estado')
48 title('Gráfico de las fallas en el convertidor del Aerogenerador 1')
```

Anexo 4 Código para procesamiento & indicadores de condición.

```
1 function [featureTable,outputTable] = diagnosticFeatures(inputData)
2 %DIAGNOSTICFEATURES recreates results in Diagnostic Feature Designer.
3 %
4 % Input:
5 % inputData: A table or a cell array of tables/matrices containing
6 the
7 % data as those imported into the app.
8 %
9 % Output:
10 % featureTable: A table containing all features and condition
11 variables.
12 % outputTable: A table containing the computation results.
13 %
14 % This function computes signals:
15 % temperaturual_interp/Datos
16 %
17 % This function computes spectra:
18 % temperaturual_ps/SpectrumData
19 %
20 % This function computes features:
21 % temperaturual_interp_sigstats/Mean
22 % temperaturual_interp_sigstats/PeakValue
23 % temperaturual_interp_sigstats/RMS
24 % temperaturual_interp_emdfeat/PeakValueIMF1
25 % temperaturual_ps_spec/BandPower
26 %
27 % Organization of the function:
28 % 1. Compute signals/spectra/features
29 % 2. Extract computed features into a table
30 %
31 % Modify the function to add or remove data processing, feature
32 generation
33 % or ranking operations.
34
35 % Auto-generated by MATLAB on 18-Jan-2024 15:26:42
36
37 % Create output ensemble.
38 outputEnsemble =
39 workspaceEnsemble(inputData, 'DataVariables', ["temperatura
40 1"; "temperatura2"; "temperatura3"]);
41
42 % Reset the ensemble to read from the beginning of the ensemble.
43 reset(outputEnsemble);
44
45 % Append new signal or feature names to DataVariables.
46 outputEnsemble.DataVariables =
47 unique([outputEnsemble.DataVariables;"temperaturual_interp";"temperatur
48 al_ps";"temperaturual_interp_sigstats";"temperaturual_interp_emdfeat";"t
49 emperaturual_ps_spec"], 'stable');
50
51 % Set SelectedVariables to select variables to read from the ensemble.
52 outputEnsemble.SelectedVariables = "temperatura 1";
53
54 % Compute sampling frequency for interpolation.
```

```

55 member1 = read(outputEnsemble);
56
57 time = readMemberData(member1,"temperatura 1/Time");
58 time = time2num(time,"seconds");
59 temperatur1_interp_Fs = effectivefs(time);
60 reset(outputEnsemble)
61
62 % Loop through all ensemble members to read and write data.
63 while hasdata(outputEnsemble)
64     % Read one member.
65     member = read(outputEnsemble);
66
67     % Get all input variables.
68     temperatur1 = readMemberData(member,"temperatura
69 1",["Time","Datos"]);
70
71     % Initialize a table to store results.
72     memberResult = table;
73
74     %% Interpolation
75     try
76         % Cálculo de Interpolación
77         time = temperatur1.Time;
78
79         % Obtener periodo de muestreo
80         samplePeriod_numeric = 1/temperatur1_interp_Fs;
81         if isduration(time) || isdatetime(time)
82             samplePeriod = seconds(samplePeriod_numeric);
83         else
84             samplePeriod = samplePeriod_numeric;
85         end
86
87         if isdatetime(time)
88             timeOrigin = datetime(0,1,1,0,0,0);
89             ivStart = min(time) - timeOrigin;
90             ivEnd = max(time) - timeOrigin;
91         else
92             ivStart = min(time);
93             ivEnd = max(time);
94         end
95
96         gridStartIdx = ceil(ivStart/samplePeriod);
97         gridEndIdx = floor(ivEnd/samplePeriod);
98
99         ivGrid = (gridStartIdx:gridEndIdx)*samplePeriod;
100        if isdatetime(time)
101            ivGrid = ivGrid + timeOrigin;
102        end
103
104        % Interpolación
105        val = interp1(time,temperatur1.Datos,ivGrid,'next',NaN);
106        temperatur1_interp =
107        table(ivGrid,val,'VariableNames',{'Time','Datos'});
108        catch
109            % Package computed signal into a table.
110

```

```

111     temperatur1_interp =
112 table(NaN,NaN, 'VariableNames', {'Time', 'Datos'});
113     end
114
115     % Append computed results to the member table.
116     memberResult = [memberResult, ...
117
118 table({temperatur1_interp}, 'VariableNames', {'temperatur1_interp'})];
119 %#ok<AGROW>
120
121     %% PowerSpectrum
122     try
123         % Get units to use in computed spectrum.
124         tuReal = "seconds";
125
126         % Compute effective sampling rate.
127         tNumeric = time2num(temperatur1_interp.Time, tuReal);
128         [Fs, irregular] = effectivefs(tNumeric);
129         %Ts = 1/Fs;
130
131         % Resample non-uniform signals.
132         x_raw = temperatur1_interp.Datos;
133         if irregular
134             x = resample(x_raw, tNumeric, Fs, 'linear');
135         else
136             x = x_raw;
137         end
138
139         % Set Welch spectrum parameters.
140         L = fix(length(x)/4.5);
141         noverlap = fix(L*50/100);
142         win = hamming(L);
143
144         % Compute the power spectrum.
145         [ps, f] = pwelch(x, win, noverlap, [], Fs);
146         w = 2*pi*f;
147
148         % Convert frequency unit.
149         factor = funitconv('rad/TimeUnit', 'Hz', 'seconds');
150         w = factor*w;
151         Fs = 2*pi*factor*Fs;
152
153         % Remove frequencies above Nyquist frequency.
154         I = w<=(Fs/2+1e4*eps);
155         w = w(I);
156         ps = ps(I);
157
158         % Configure the computed spectrum.
159         ps = table(w, ps, 'VariableNames', {'Frequency',
160 'SpectrumData'});
161         ps.Properties.VariableUnits = {'Hz', ''};
162         ps = addprop(ps, {'SampleFrequency'}, {'table'});
163         ps.Properties.CustomProperties.SampleFrequency = Fs;
164         temperatur1_ps = ps;
165     catch
166

```

```

167     temperatur1_ps = table(NaN, NaN, 'VariableNames',
168 {'Frequency', 'SpectrumData'});
169     end
170
171     % Append computed results to the member table.
172     memberResult = [memberResult, ...
173     table({temperatur1_ps}, 'VariableNames', {'temperatur1_ps'})];
174     %#ok<AGROW>
175
176     %% SignalFeatures
177     try
178         % Compute signal features.
179         inputSignal = temperatur1_interp.Datos;
180         Mean = mean(inputSignal, 'omitnan');
181         PeakValue = max(abs(inputSignal));
182         RMS = rms(inputSignal, 'omitnan');
183
184         % Concatenate signal features.
185         featureValues = [Mean, PeakValue, RMS];
186
187         % Package computed features into a table.
188         featureNames = {'Mean', 'PeakValue', 'RMS'};
189         temperatur1_interp_sigstats =
190 array2table(featureValues, 'VariableNames', featureNames);
191     catch
192         % Package computed features into a table.
193         featureValues = NaN(1,3);
194         featureNames = {'Mean', 'PeakValue', 'RMS'};
195         temperatur1_interp_sigstats =
196 array2table(featureValues, 'VariableNames', featureNames);
197     end
198
199     % Append computed results to the member table.
200     memberResult = [memberResult, ...
201     table({temperatur1_interp_sigstats}, 'VariableNames', {'temperatur1_in
202     terp_sigstats'})]; %#ok<AGROW>
203
204
205     %% EMDFeatures
206     try
207         % Compute Empirical Mode Decomposition based features.
208         inputSignal = temperatur1_interp.Datos;
209         % Compute Empirical Mode Decomposition of the signal.
210         outputEMD = emd(inputSignal, "MaxNumIMF", 1);
211
212         PeakValueIMF1 = max(abs(outputEMD(:,1)));
213
214         % Concatenate signal features.
215         featureValues = PeakValueIMF1;
216
217         % Package computed features into a table.
218         featureNames = {'PeakValueIMF1'};
219         temperatur1_interp_emdfeat =
220 array2table(featureValues, 'VariableNames', featureNames);
221     catch
222         % Package computed features into a table.

```

```

223     featureValues = NaN(1,1);
224     featureNames = {'PeakValueIMF1'};
225     temperatur1_interp_emdfeat =
226 array2table(featureValues, 'VariableNames', featureNames);
227     end
228
229     % Append computed results to the member table.
230     memberResult = [memberResult, ...
231
232 table({temperatur1_interp_emdfeat}, 'VariableNames', {'temperatur1_int
233 erp_emdfeat'})]; %#ok<AGROW>
234
235     %% SpectrumFeatures
236     try
237         % Compute spectral features.
238         % Get frequency unit conversion factor.
239         factor = funitconv('Hz', 'rad/TimeUnit', 'seconds');
240         ps = temperatur1_ps.SpectrumData;
241         w = temperatur1_ps.Frequency;
242         w = factor*w;
243         mask_1 = (w>=factor*6.510416666666667e-06) &
244 (w<=factor*0.0008333333333333333);
245         ps = ps(mask_1);
246         w = w(mask_1);
247
248         % Compute spectral peaks.
249         % [peakAmp,peakFreq] = findpeaks(ps,w/factor,'MinPeakHeight',-
250 Inf, ...
251 %
252 'MinPeakProminence',0,'MinPeakDistance',0.001,'SortStr','descend','NPe
253 aks',1);
254         % peakAmp = [peakAmp(:); NaN(1-numel(peakAmp),1)];
255         % peakFreq = [peakFreq(:); NaN(1-numel(peakFreq),1)];
256
257         % Extract individual feature values.
258         BandPower = trapz(w/factor,ps);
259
260         % Concatenate signal features.
261         featureValues = BandPower;
262
263         % Package computed features into a table.
264         featureNames = {'BandPower'};
265         temperatur1_ps_spec =
266 array2table(featureValues, 'VariableNames', featureNames);
267     catch
268         % Package computed features into a table.
269         featureValues = NaN(1,1);
270         featureNames = {'BandPower'};
271         temperatur1_ps_spec =
272 array2table(featureValues, 'VariableNames', featureNames);
273     end
274
275     % Append computed results to the member table.
276     memberResult = [memberResult, ...
277 table({temperatur1_ps_spec}, 'VariableNames', {'temperatur1_ps_spec'})
278 ]; %#ok<AGROW>

```

```

279
280     %% Write all the results for the current member to the ensemble.
281     writeToLastMemberRead(outputEnsemble,memberResult)
282 end
283
284 % Gather all features into a table.
285 selectedFeatureNames =
286 ["temperatural_interp_sigstats/Mean","temperatural_interp_sigstats/Pea
287 kValue","temperatural_interp_sigstats/RMS","temperatural_interp_emdfea
289 t/PeakValueIMF1","temperatural_ps_spec/BandPower"];
290 featureTable =
291 readFeatureTable(outputEnsemble,'Features',selectedFeatureNames);
292
293 % Set SelectedVariables to select variables to read from the ensemble.
294 outputEnsemble.SelectedVariables =
295 unique([outputEnsemble.DataVariables;outputEnsemble.ConditionVariables
296 ;outputEnsemble.IndependentVariables],'stable');
297
298 % Gather results into a table.
299 outputTable = readall(outputEnsemble);
300 end

```

Anexo 5 Selección de características.

```
1 Ranking de Caracteristicas
2 c=tabla_caracte.("Código de Falla");
3 tablacara2=tabla_caracte(:,2:end);
4
5 for i=1:width(tablacara2)
6     h(:,i)=tablacara2(:,i)';
7     [~,tt(i)] = rankfeatures(tablacara2(:,i)',c,'Criterion',"ttest");
8 end
9 % Encuentra los valores NaN
10 nan_indices = isnan(tt);
11
12 % Reemplaza los valores NaN con ceros
13 tt(nan_indices) = 0';
14
15 % Ordenamiento de las características
16 ranking_caracte=array2table(tt,"VariableNames",selecci_nombre_cara);
17 u=rows2vars(ranking_caracte);
18 v=sortrows(u,2,'descend');
19
20 %MEJORES CARACTERÍSTICAS
21 %Seleccionar las primeras características importantes
22 n_mejores_caracte = 10;
23 mejores_indices = v.OriginalVariableNames(1:n_mejores_caracte);
24 % Obtener las características correspondientes a los índices
25 seleccionados
26 mejores_caracte =[tabla_caracte(:,mejores_indices)
27 array2table(tabla_caracte.("Código de Falla"),"VariableNames",{ 'Código
28 de fallo'})]];
29 figure
30 barh(v.OriginalVariableNames,v.Var1) %Ordenar las características por
31 barras
```


Anexo 6 Código de entrenamiento de modelos.

```
1 function [trainedClassifier, validationAccuracy] =
2 trainClassifier(trainingData)
3 % [trainedClassifier, validationAccuracy] =
4 trainClassifier(trainingData)
5 % Returns a trained classifier and its accuracy. This code recreates
6 the
7 % classification model trained in Classification Learner app. Use the
8 % generated code to automate training the same model with new data, or
9 to
10 % learn how to programmatically train models.
11 %
12 % Input:
13 % trainingData: A table containing the same predictor and response
14 % columns as those imported into the app.
15 %
16 % Output:
17 %     trainedClassifier: A struct containing the trained classifier.
18 The
19 %     struct contains various fields with information about the
20 trained
21 %     classifier.
22 %     trainedClassifier.predictFcn: A function to make predictions on
23 new
24 %     data.
25 %
26 %     validationAccuracy: A double representing the validation
27 accuracy as
28 %     a percentage. In the app, the Models pane displays the
29 validation
30 %     accuracy for each model.
31 %
32 % Use the code to train the model with new data. To retrain your
33 % classifier, call the function from the command line with your
34 original
35 % data or new data as the input argument trainingData.
36 %
37 % For example, to retrain a classifier trained with the original data
38 set
39 % T, enter:
40 %     [trainedClassifier, validationAccuracy] = trainClassifier(T)
41 %
42 % To make predictions with the returned 'trainedClassifier' on new
43 data T2,
44 % use
45 %     [yfit,scores] = trainedClassifier.predictFcn(T2)
46 %
47 % T2 must be a table containing at least the same predictor columns as
48 used
49 % during training. For details, enter:
50 %     trainedClassifier.HowToPredict
51
52 % Auto-generated by MATLAB on 19-Jan-2024 14:09:43
53
54 % Extract predictors and response
```

```

55 % This code processes the data into the right shape for training the
56 % model.
57 inputTable = trainingData;
58 predictorNames = {'temperatural_interp_sigstats/Mean',
59 'temperatural_interp_sigstats/PeakValue',
60 'temperatural_interp_sigstats/RMS',
61 'temperatural_interp_emdfeat/PeakValueIMF1',
62 'temperatural_ps_spec/BandPower'};
63 predictors = inputTable(:, predictorNames);
64 response = inputTable('Código de Falla');
65 isCategoricalPredictor = [false, false, false, false, false];
66 classNames = [0; 1];
67
68 % Train a classifier
69 % This code specifies all the classifier options and trains the
70 classifier.
71 classificationTree = fitctree(...
72     predictors, ...
73     response, ...
74     'SplitCriterion', 'gdi', ...
75     'MaxNumSplits', 4, ...
76     'Surrogate', 'off', ...
77     'ClassNames', classNames);
78
79 % Create the result struct with predict function
80 predictorExtractionFcn = @(t) t(:, predictorNames);
81 treePredictFcn = @(x) predict(classificationTree, x);
82 trainedClassifier.predictFcn = @(x)
83     treePredictFcn(predictorExtractionFcn(x));
84 % Add additional fields to the result struct
85 trainedClassifier.RequiredVariables =
86 {'temperatural_interp_sigstats/Mean',
87 'temperatural_interp_sigstats/PeakValue',
88 'temperatural_interp_sigstats/RMS',
89 'temperatural_interp_emdfeat/PeakValueIMF1',
90 'temperatural_ps_spec/BandPower'};
91 trainedClassifier.ClassificationTree = classificationTree;
92 trainedClassifier.About = 'This struct is a trained model exported
93 from Classification Learner R2023b.';
94 trainedClassifier.HowToPredict = sprintf('To make predictions on a new
95 table, T, use: \n [yfit,scores] = c.predictFcn(T) \nreplacing ''c''
96 with the name of the variable that is this struct, e.g.
97 ''trainedModel''. \n \nThe table, T, must contain the variables
98 returned by: \n c.RequiredVariables \nVariable formats (e.g.
99 matrix/vector, datatype) must match the original training data.
100 \nAdditional variables are ignored. \n \nFor more information, see <a
101 href="matlab:helpview(fullfile(docroot, ''stats'', ''stats.map''),
102 ''appclassification_exportmodeltoworkspace'')">How to predict using an
103 exported model</a>.'.');
104
105 % Extract predictors and response
106 % This code processes the data into the right shape for training the
107 % model.
108 inputTable = trainingData;
109 predictorNames = {'temperatural_interp_sigstats/Mean',
110 'temperatural_interp_sigstats/PeakValue',

```

```

111 'temperatural_interp_sigstats/RMS',
112 'temperatural_interp_emdfeat/PeakValueIMF1',
113 'temperatural_ps_spec/BandPower'};
114 predictors = inputTable(:, predictorNames);
115 response = inputTable('Código de Falla');
116 isCategoricalPredictor = [false, false, false, false, false];
117 classNames = [0; 1];
118
119 % Perform cross-validation
120 partitionedModel = crossval(trainedClassifier.ClassificationTree,
121 'Kfold', 5);
122
123 % Compute validation predictions
124 [validationPredictions, validationScores] =
125 kfoldPredict(partitionedModel);
126
127 % Compute validation accuracy
128 validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun',
129 'ClassifError');

```

Anexo 7. Código de importe del archivo (complete_dataset.csv).

```
1 clc;clear
2 opts = delimitedTextImportOptions("NumVariables", 70);
3
4 % Specify range and delimiter
5 opts.DataLines = [2, Inf];
6 opts.Delimiter = ",";
7
8 % Specify column names and types
9 opts.VariableNames = ["VarName1", "timestamp", "wind_speed_avg",
10 "wind_speed_max", "wind_speed_min", "grid_active_power_avg",
11 "grid_active_power_max", "grid_active_power_min",
12 "generator_capacitors_temperature_max", "converter_reactive_power_avg",
13 "converter_reactive_power_max", "converter_reactive_power_min",
14 "generator_speed_avg", "generator_speed_max", "generator_speed_min",
15 "ambient_temperature_avg", "ambient_temperature_max",
16 "ambient_temperature_min", "blade_angle_avg", "blade_angle_max",
17 "blade_angle_min", "nacelle_position_avg", "grid_U1_avg",
18 "grid_U2_avg", "grid_U3_avg", "grid_I1_avg", "grid_I2_avg",
19 "grid_I3_avg", "acceleration_nacelle_max", "winding_temperature_max",
20 "topbox_temperature_max", "nacelle_temperature_max",
21 "igbt_temperature_max", "dc_link_capacitors_temperature_max",
22 "ac_inductor_temperature_max", "rectifier_temperature_max",
23 "chopper_igbt_temperature_max", "dc_inductor_temperature_max",
24 "step_up_igbt_temperature_max", "pitch_motor_temperature_1_max",
25 "pitch_motor_temperature_2_max", "pitch_motor_temperature_3_max",
26 "pitch_capacitor_temperature_1_max",
27 "pitch_capacitor_temperature_2_max",
28 "pitch_capacitor_temperature_3_max", "pitch_cabinet_temperature_1_max",
29 "pitch_cabinet_temperature_2_max", "pitch_cabinet_temperature_3_max",
30 "pitch_converter_temperature_1_max",
31 "pitch_converter_temperature_2_max",
32 "pitch_converter_temperature_3_max",
33 "pitch_power_supply_temperature_1_max",
34 "pitch_power_supply_temperature_2_max",
35 "pitch_power_supply_temperature_3_max", "operation_mode",
36 "data_energy_yield", "data_consumed_energy_yield",
37 "data_Power_on_time", "data_WTG_ok_time", "data_error_time",
38 "data_environment_ok_time", "data_environment_not_ok_time",
39 "data_service_time", "data_grid_control_standstill_time",
40 "data_power_production_time", "today_data_energy_yield",
41 "yesterday_data_energy_yield", "two_day_before_data_energy_yield",
42 "three_days_before_data_energy_yield", "Aero_Number"];
43 opts.VariableTypes = ["double", "datetime", "double", "double",
44 "double", "double", "double", "double", "double", "double", "double",
45 "double", "double", "double", "double", "double", "double", "double",
46 "double", "double", "double", "double", "double", "double", "double",
47 "double", "double", "double", "double", "double", "double", "double",
48 "double", "double", "double", "double", "double", "double", "double",
49 "double", "double", "double", "double", "double", "double", "double",
50 "double", "double", "double", "double", "double", "double", "double",
51 "double", "double", "double", "double", "double", "double", "double",
52 "double", "double", "double", "double", "double", "double", "double",
53 "double", "double", "double"];
54
```

```
55 % Specify file level properties
56 opts.ExtraColumnsRule = "ignore";
57 opts.EmptyLineRule = "read";
58
59 % Specify variable properties
60 opts = setvaropts(opts, "timestamp", "InputFormat", "yyyy-MM-dd
61 HH:mm:ss");
62
63 % Import the data
64 m1 = readtable("E:\Trabajo de Titulación\ARCHIVOS VILLONACO\Código
65 fallos\complete_dataset.csv", opts)
66
67 Clear temporary variables
68 clear opts
69 TORRE_11=m1(find(m1.Aero_Number==11),:)
```

Anexo 8. Variables de temperatura seleccionadas del convertidor.

	timestamp	pitch_converter_temperature_1_max	pitch_converter_temperature_2_max	pitch_converter_temperature_3_max	operation_mode	
1	2014-01-01 00:00:00	18.4000	18.3000	18.3000	5	▲
2	2014-01-01 00:10:00	18.4000	18.3000	18.3000	5	
3	2014-01-01 00:20:00	18.3000	18.3000	18.3000	5	
4	2014-01-01 00:30:00	18.3000	18.2000	18.3000	5	
5	2014-01-01 00:40:00	18.3000	18.2000	18.3000	5	
6	2014-01-01 00:50:00	18.3000	18.2000	18.3000	5	
7	2014-01-01 01:00:00	18.3000	18.2000	18.3000	5	
8	2014-01-01 01:10:00	18.3000	18.1000	18.3000	5	
9	2014-01-01 01:20:00	18.3000	18.1000	18.3000	5	
10	2014-01-01 01:30:00	18.3000	18.1000	18.3000	5	
11	2014-01-01 01:40:00	18.2000	18.1000	18.2000	5	
12	2014-01-01 01:50:00	18.2000	18.1000	18.2000	5	
13	2014-01-01 02:00:00	18.2000	18.1000	18.2000	5	
14	2014-01-01 02:10:00	18.1000	18.1000	18.1000	5	
15	2014-01-01 02:20:00	18.1000	18	18.1000	5	
16	2014-01-01 02:30:00	18.1000	18	18.1000	5	
17	2014-01-01 02:40:00	18.1000	18	18.1000	5	
18	2014-01-01 02:50:00	18.1000	18	18.1000	5	
19	2014-01-01 03:00:00	18.1000	18	18.1000	5	
20	2014-01-01 03:10:00	18.1000	18	18	5	
21	2014-01-01 03:20:00	18	17.9000	18	5	
22	2014-01-01 03:30:00	18	17.9000	18	5	
23	2014-01-01 03:40:00	18	17.8000	18	5	
24	2014-01-01 03:50:00	18	17.8000	17.9000	5	
25	2014-01-01 04:00:00	18	17.8000	17.9000	5	
26	2014-01-01 04:10:00	17.9000	17.8000	17.9000	5	
27	2014-01-01 04:20:00	17.8000	17.7000	17.8000	5	
28	2014-01-01 04:30:00	17.8000	17.7000	17.8000	5	
29	2014-01-01 04:40:00	17.8000	17.6000	17.8000	5	
30	2014-01-01 04:50:00	17.8000	17.6000	17.8000	5	
31	2014-01-01 05:00:00	17.8000	17.6000	17.7000	5	▼

Anexo 9 Preprocesamiento y preparación de la tabla para DFD.

```
1  Preprocesamiento para preparación de la tabla para la Aplicación
2  Diagnostic Feature Designer
3  resul = zeros(size(T1_1.timestamp, 1), 1);
4
5  for i=1:numel(selec_tor1.FaultStartTime) % numel: Número de elementos
6  del arreglo
7      resul=resul | (T1_1.timestamp>=selec_tor1.FaultStartTime(i)) &
8      (T1_1.timestamp<=selec_tor1.FaultEndTime(i) &
9      (selec_tor1.Codigo_Fallo(i)=="1"));
10 end
11
12 resultados=double(resul);
13 T1_1=[T1_1,array2table(resultados,"VariableNames',{'falla'}]);
14
15 % Encuentra las fechas únicas (días)
16 fechas_unicas2 = unique(dateshift(T1_1.timestamp, 'start', 'day'));
17 ceros_unos = zeros(size(fechas_unicas2));
18 % Itera a través de las fechas únicas (días)
19 for i = 1:numel(fechas_unicas2)
20     % fecha_actual = fechas_unicas(i);
21
22     % Filtra los datos para el día actual
23     datos_dia = T1_1.falla(dateshift(T1_1.timestamp, 'start', 'day') ==
24 fechas_unicas2(i));
25
26     % Si hay al menos un '1', asigna 1 en los resultados para este día
27     if any(datos_dia == 1)
28         ceros_unos(i) = 1;
29     end
30
31 end
32 cod_cero_uno=ceros_unos;
33
34 % Itera a través de las fechas únicas y calcula el promedio de cada
35 día.
36 for i = 1:length(fechas_unicas2)
37     fecha_actual = fechas_unicas2(i);
38
39     datos_dia = T1_1(T1_1.timestamp >= fecha_actual & T1_1.timestamp <
40 fecha_actual + days(1), :);
41 tabla_agrupada(i,:)=table({datos_dia});
42 end
43 % Itera sobre cada tabla y elimina las filas donde la altura es menor o
44 igual a 10.
45 tabla_individual = tabla_agrupada.Var1;
46     % Encuentra las filas donde la conversión a número fue exitosa y
47 el valor es menor o igual a 10.
48 filas_a_eliminar = cellfun(@(x) height(x) <= 10, tabla_individual);
49
50     % Elimina esas filas de la tabla original.
51 tabla_agrupada(filas_a_eliminar, :) = [];
52
53 Convertir una matriz en una tabla
54 [m,n]=size(tabla_agrupada.Var1{1});
```

```

55     tv_ensemble_table = table();
56     for j=1:size(tabla_agrupada,1)
57         T1 = tabla_agrupada.Var1{j}(:, [1 2]);
58         T2 = tabla_agrupada.Var1{j}(:, [1 3]);
59         T3 = tabla_agrupada.Var1{j}(:, [1 4]);
60         tiempo = T1.timestamp - T1.timestamp(1); % Resta con respecto a la
61 primera fecha
62         temp1 =
63 array2timetable(T1.pitch_converter_temperature_1_max, 'RowTimes', tiempo,
64 'VariableNames', {'Datos'}); %Convertir una matriz homogénea en un
65 horario
66         temp2 =
67 array2timetable(T2.pitch_converter_temperature_2_max, 'RowTimes', tiempo,
68 'VariableNames', {'Datos'});
69         temp3 =
array2timetable(T3.pitch_converter_temperature_3_max, 'RowTimes', tiempo,
'VariableNames', {'Datos'});
        tv_member =
table({temp1}, {temp2}, {temp3}, cod_cero_uno(j), 'VariableNames', {'tempera
tura 1', 'temperatura2', 'temperatura3', 'Código de Falla'});
        tv_ensemble_table(j,:) = tv_member;
    end

    %%Eliminar Valores Costantes si es todo el día
    for k=1:size(tv_ensemble_table)
miembrosConstantes =tv_ensemble_table("temperatura 1"){k};
mie(k,:) = all(diff(miembrosConstantes.Datos,2) == 0); %diff: calcula
diferencias entre elementos adyacentes; %all: Determinar si todos los
elementos de un arreglo son distintos a cero o true
    end
tv_ensemble_table(mie,:)=[];

```


Anexo 10. Certificado de traducción del resumen.

Certificado de Traducción al Idioma Inglés

René Mauricio Gómez González, Licenciado en Ciencias de la Educación mención idioma Inglés y Máster Univeritario en Enseñanza del Inglés, con C.I: 1105140865.

Certifico:

Que el texto traducido al idioma inglés que compone el **Resumen** del Trabajo de Titulación denominado: **Exploración y aplicación de "DIAGNOSTIC FEATURE DESIGNER" para desarrollar modelos de diagnóstico de fallos en plantas eólicas** correspondiente al **Sr. Wilmer Andrés Santos Ordoñez**, con número de cédula **1105424848**, fue realizado y verificado bajo mi supervisión.

Es todo en cuanto puedo indicar en honor a la verdad, facultando al interesado hacer uso del presente documento para los fines pertinentes.

Loja, 24 de julio de 2024



Firma: Lic. René Gómez Mg.

C.I: 1105140865

Celular: 0989419049