

TRABAJO ESPECIAL DE GRADO

MODELADO DE LA DEGRADACIÓN DEL DESEMPEÑO DE UNA TURBOMÁQUINA USANDO APRENDIZAJE AUTOMÁTICO

Presentado ante la Ilustre
Universidad Central de Venezuela
Por el Br. Ronny D. Portillo M.
Para optar al Título
de Ingeniero Químico

Caracas, Noviembre 2018

TRABAJO ESPECIAL DE GRADO

MODELADO DE LA DEGRADACIÓN DEL DESEMPEÑO DE UNA TURBOMÁQUINA USANDO APRENDIZAJE AUTOMÁTICO

TUTOR ACADÉMICO: Prof. Alí Gabriel Lara
TUTOR ACADÉMICO: Prof. Andrés Rosales

Presentado ante la Ilustre
Universidad Central de Venezuela
Por el Br. Ronny D. Portillo M.
Para optar al Título
de Ingeniero Químico

Caracas, Noviembre 2018

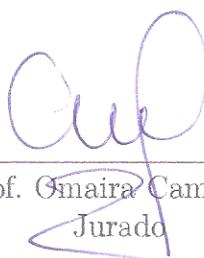
Caracas, Noviembre 2018

APROBACIÓN DEL JURADO

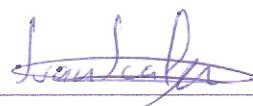
Los abajo firmantes, miembros del jurado designado por el Consejo de Facultad de Ingeniería de la Universidad Central de Venezuela, para evaluar el Trabajo Especial de Grado, presentado por el Bachiller Ronny Portillo, titulado:

"MODELADO DE LA DEGRADACIÓN DEL DESEMPEÑO DE UNA TURBOMÁQUINA USANDO APRENDIZAJE AUTOMÁTICO"

consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniero Químico, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.



Prof. Omaira Camacaro
Jurado



Prof. Juan Calderón
Jurado



Prof. Alí Gabriel Lara
Tutor Académico



Prof. Andrés Rosales
Tutor Académico



Caracas, 20 de noviembre del 2018

ACTA

MENCIÓN HONORÍFICA

Los miembros del jurado examinador del Trabajo Especial de Grado para optar al Título de Ingeniera Química presentado por la bachiller **Ronny David, Portillo Machado** (C.I. 21072426) y cuyo título es "**MODELADO DE LA DEGRADACIÓN DEL DESEMPEÑO DE UNA TURBOMÁQUINA USANDO APRENDIZAJE AUTOMÁTICO**" consideramos que merece ser reconocido con una Mención Honorífica.

Luego de finalizada la defensa pública y considerando que:

1. En el presente trabajo especial de grado, el Br. Portillo integró los conocimientos adquiridos en la carrera, específicamente del área de termodinámica, métodos numéricos, estadística y programación con los diversos algoritmos de machine learning para la evaluación de procesos industriales. Además puso en práctica conocimientos asociados a contenidos temáticos como el diseño de redes neuronales y estadística avanzada.
2. Su trabajo especial de grado inicia una nueva línea de investigación y desarrollo para la Escuela de Ing. Química de la UCV en la cual se puede modelar procesos industriales, basados en datos reales de producción, a fin de identificar problemas de operación, detección de fallas y proponer soluciones que permitan la optimización de dichos procesos.
3. Desarrolló una metodología donde, aplicando métodos de deep learning, se puede crear APIs para monitorear y evaluar procesos industriales que asienta las bases para la creación de soluciones industriales de alto nivel a la nación.
4. *El aporte fundamental del TEG en cuanto a la generación de conocimientos, se fundamenta en el hecho de haber demostrado que el uso de la tecnología de Machine Learning es aplicable al campo de mantenimiento predictivo de equipos industriales*

Acordado y firmado a los veinte días de noviembre de 2018

Prof. Omaira Camacaro
Jurado

Prof. Juan Calderón
Jurado

Prof. Ali Gabriel Lara
Docente Tutor

Prof. Andrés Rosales
Tutor académico

DEDICATORIA

A mi querida abuela Marcia Machado (mami) te extraño y te recuerdo con mucha ilusión, esto es para ti con todo el amor del mundo.

A Anibal Almeida (chicho) gracias por todo el apoyo que usted y su familia me brindaron durante mi estadía alrededor de ustedes.

*A los que tienen paciencia y viven en la conciencia
de que todo lo que buscan llegará tarde o temprano
si se preparan debidamente.*

AGRADECIMIENTOS

A mis padres y hermanos. Son la razón por la cual he luchado tanto durante esta carrera y me han ayudado mucho. Gracias!

A mis amigos. Llegar a este punto hubiera sido mucho más difícil si no hubiera contado con el apoyo de ustedes.

Al Ing. Alí Gabriel Lara por ayudarme durante todo este tiempo para desarrollar este trabajo y principalmente, por haberme mostrado este mundo que me apasiona.

Ronny D, Portillo M.

**MODELADO DE LA DEGRADACIÓN DEL DESEMPEÑO DE
UNA TURBOMÁQUINA USANDO APRENDIZAJE
AUTOMÁTICO**

Tutor Académico: Prof. Alí Gabriel Lara. Tutor Académico: Prof.

Andrés Rosales

**Tesis. Caracas, U.C.V. Facultad de Ingeniería. Escuela de Ingeniería
Química. Año 2018, n° pag. 108**

Palabras Claves: Desempeño de turbofan, Mantenimiento predictivo,
Detección de fallas, Degradación, Machine learning

Resumen.

Los costos relacionados con el mantenimiento de turbomáquinas y la necesidad de un alto rendimiento, han generado un interés considerable en las técnicas avanzadas de mantenimiento basadas en la condición y la salud de equipos. Este Trabajo Especial de Grado diseña un esquema de diagnóstico y pronóstico para estimar la detección de fallas y predecir el tiempo de vida útil remanente para motores aeronáuticos (turbofan), utilizando metodologías basadas en aprendizaje automático para garantizar la seguridad y el rendimiento del equipo. Algoritmos como, regresión logística, árboles de decisión, bosques aleatorios, vectores soportados lineales y K-vecinos, se emplean para la detección de fallas; así como también en la fase de predicción del tiempo de vida útil remanente se emplean algoritmos de regresión y redes neuronales artificiales. Las redes neuronales recurrentes con neuronas LSTM son usadas. Se pueden producir varias degradaciones en el motor que resultan en cambios en el rendimiento de sus componentes. Los deterioros del ventilador y del compresor de alta presión son modelados bajo diversas condiciones de degradación. Los algoritmos de clasificación, regresión y las redes neuronales propuestas se desarrollan y aplican para capturar la dinámica de estas degradaciones en el turbofan. La condición de salud del motor se predice debido a la aparición de estos deterioros.

ÍNDICE GENERAL

<i>Contenido</i>	<i>Pag.</i>
ÍNDICE DE FIGURAS	xiv
ÍNDICE DE TABLAS	xvi
CAPÍTULO I. INTRODUCCIÓN	1
CAPÍTULO II. FUNDAMENTO DE LA INVESTIGACIÓN	3
II.1. PLANTEAMIENTO DEL PROBLEMA	3
II.2. ANTECEDENTES	4
II.3. OBJETIVOS	8
II.3.1. Objetivo General	8
II.3.2. Objetivos Específicos	8
CAPÍTULO III. MARCO TEÓRICO	10
III.1. Motores aeronáuticos	10
III.1.1. Clasificación de los motores aeronáuticos	11
III.2. Turbofan	11
III.2.1. Principio de funcionamiento	13
III.3. Límites de operación	15
III.4. Degradación del desempeño de un compresor	16
III.4.1. Tipos de degradación	18

III.4.2. Mecanismos de deterioro	19
III.5. Aprendizaje automático (machine learning)	21
III.6. Entrenamiento de modelos de machine learning	22
III.7. Tipos de aprendizajes en machine learning	23
III.7.1. Aprendizaje automático supervisado	23
III.7.2. Aprendizaje automático no supervisado	24
III.7.3. Aprendizaje profundo (deep learning)	25
III.7.4. Clasificación binaria	26
III.7.5. Matriz de confusión	28
III.7.6. Modelos de regresión	28
III.7.7. Métricas de rendimiento de modelos	28
III.7.8. Factor del valor atípico local	32
III.7.9. DBSCAN	33
III.7.10. Eliminación recursiva de características (RFE)	33
III.7.11. Red neuronal recurrente con neurona LSTM	33
III.7.12. Vida útil remanente	35
III.7.13. Índice de degradación	35
CAPÍTULO IV. MARCO METODOLÓGICO	36
IV.1. Investigar sobre el mecanismo y el modelado del proceso de degradación del desempeño de una turbomáquina enfocado en aplicaciones de pronóstico.	36
IV.2. Desarrollar un modelo para detección de falla para una turbomáquina a partir de datos típicos de operación.	36

IV.3. Desarrollar un modelo para estimar la degradación del desempeño de una turbomáquina a partir de datos típicos de operación.	37
IV.4. Caracterizar el proceso de degradación empleando los parámetros: índice de degradación y tiempo de vida útil remanente.	37
CAPÍTULO V. RESULTADOS Y DISCUSIÓN	38
V.1. Investigar sobre el mecanismo y el modelado del proceso de degradación del desempeño de un turbofan enfocado en aplicaciones de pronóstico.	38
V.2. Desarrollar un modelo para detección de falla para un turbofan a partir de datos típicos de operación.	46
V.2.1. Adaptación de los datos	47
V.2.2. Análisis exploratorio de los datos	48
V.2.3. Eliminación de valores atípicos	55
V.2.4. Agrupamiento de los datos	59
V.2.5. Normalización de los datos	60
V.2.6. Selección de características	60
V.2.7. Modelo de clasificación para la detección de fallas	63
V.2.8. Desarrollar un modelo para estimar la degradación del desempeño de un turbofan a partir de datos típicos de operación.	76
V.2.9. Caracterizar el proceso de degradación empleando los parámetros: índice de degradación y tiempo de vida útil remanente.	85
CAPÍTULO VI. CONCLUSIONES Y RECOMENDACIONES	88
VI.1. CONCLUSIONES	88

VI.2. RECOMENDACIONES	89
BIBLIOGRAFÍA	90

ÍNDICE DE FIGURAS

<i>Figura</i>	<i>Pag.</i>
Figura N° 1. Motor aeronáutico	10
Figura N° 2. Partes de un Turbofan	12
Figura N° 3. Flujos de aire en un turbofan	12
Figura N° 4. Turbofan	14
Figura N° 5. Mapa de un compresor	16
Figura N° 6. Degradación del desempeño de un compresor	17
Figura N° 7. Posibles deterioros en turbo-maquinas	19
Figura N° 8. Compresor multi-etapa afectado por corrosión	20
Figura N° 9. Ensuciamiento que puede tener lugar en las palas del impulsor	20
Figura N° 10. Impulsor afectado por ensuciamiento	21
Figura N° 11. Representación básica del aprendizaje automático	22
Figura N° 12. Representación básica del aprendizaje supervisado	24
Figura N° 13. Representación básica del aprendizaje no supervisado	25
Figura N° 14. Representación de los niveles de la IA.	26
Figura N° 15. Representación gráfica de las RNA	26
Figura N° 16. Matriz de confusión para el ejemplo de exactitud.	29
Figura N° 17. Matriz de confusión para el ejemplo de precisión.	30
Figura N° 18. Área bajo la curva ROC.	31

Figura N° 19. El módulo de repetición en un LSTM contiene cuatro capas interactivas.	34
Figura N° 20. el rango P-F será la vida útil remanente.	35
Figura N° 21. Localización de los sensores típicos en un turbofan	39
Figura N° 22. Porcentaje de uso de la base de datos de la NASA en el estudio PHM.	40
Figura N° 23. Estructura datos de entrenamiento.	42
Figura N° 24. Estructura datos de prueba	43
Figura N° 25. Detección de fallas. Se indica mediante la estrella roja.	46
Figura N° 26. Sensores para la máquina 1.	49
Figura N° 27. Varianza de todos los sensores.	50
Figura N° 28. Distribución de los sensores.	51
Figura N° 29. Prueba de normalidad estadística (La imagen es similar para las pruebas de dos colas, sólo que aparecerían áreas amarillas y verdes en ambas colas).	52
Figura N° 30. Matriz de distribuciones (la mayoría de las características tienen una relación no lineal con respecto al RUL).	53
Figura N° 31. Matriz de correlación del archivo 1.	54
Figura N° 32. Serie de tiempo del sensor 7 de la máquina 1.	55
Figura N° 33. Serie de tiempo del sensor 7 con el score del LOF.	56
Figura N° 34. Serie de tiempo del sensor 7 señalando los valores atípicos.	57
Figura N° 35. Serie de tiempo del sensor 2 señalando los valores atípicos.	57
Figura N° 36. Serie de tiempo del sensor 7 sin valores atípicos.	58
Figura N° 37. Agrupamiento del DBSCAN.	59

Figura N° 38. Gráfica de la selección de características para los cuatro archivos.	61
Figura N° 39. Con la selección de características se obtienen atributos que representan de manera más clara y con una alta precisión a los datos originales.	62
Figura N° 40. Ilustración del recorrido en el área de Pronósticos y Diagnósticos de la salud.	63
Figura N° 41. Matriz de confusión para Regresión Logística.	66
Figura N° 42. Curva ROC y Curva Precisión-Recuperación de la Regresión Logística.	67
Figura N° 43. Matriz de confusión para Árboles de decisión.	68
Figura N° 44. Curva ROC y Curva Precisión-Recuperación para Árboles de Decisión.	69
Figura N° 45. Matriz de confusión para los Bosques aleatorios.	69
Figura N° 46. Matriz de confusión para los vectores soportados.	70
Figura N° 47. Curva ROC y Curva Precisión-Recuperación para los vectores soportados lineales.	71
Figura N° 48. Matriz de confusión para los K-Vecinos.	72
Figura N° 49. Curva ROC y Curva Precisión-Recuperación para los K-Vecinos.	73
Figura N° 50. Curvas ROC y Curvas Precisión-Recuperación para todos los algoritmos.	74
Figura N° 51. Ilustración del recorrido en el área de Pronósticos y Diagnósticos de la salud.	76

Figura N° 52. Ilustración de las etapas de la degradación de un activo representando el rango del RUL.	76
Figura N° 53. Estructura de los datos para una Red Neuronal Recurrente	81
Figura N° 54. Gráfica de la pérdida o error de los datos de entrenamiento y los de validación de la RNN.	83
Figura N° 55. RUL vs Ciclos para una máquina escogida al azar de la base de datos de prueba.	85
Figura N° 56. Comportamiento del índice de degradación al evaluarla en distintos ciclos.	87

ÍNDICE DE TABLAS

<i>Tabla</i>	<i>Pag.</i>
Tabla N° 1. Conjunto de datos del simulador C-MAPSS para un turbofan.	41
Tabla N° 2. Estructura datos RUL (simplificado)	44
Tabla N° 3. Estructura de todos los dataset	44
Tabla N° 4. Estructura datos RUL	45
Tabla N° 5. Datos de entrenamiento.	47
Tabla N° 6. Condiciones de operación únicas.	48
Tabla N° 7. Datos con la nueva columna de las configuraciones de operación.	48
Tabla N° 8. Importancia de los sensores seleccionados	62
Tabla N° 9. Estructura datos de prueba asociados al archivo 1	64
Tabla N° 10. Resultados de la Regresión Logística	65
Tabla N° 11. Métricas para la Regresión Logística	66
Tabla N° 12. Resultados de los Árboles de Decisión	67
Tabla N° 13. Métricas para los Árboles de Decisión	68
Tabla N° 14. Resultados de los Bosques aleatorios	69
Tabla N° 15. Métricas para los Bosques Aleatorios	70
Tabla N° 16. Resultados de los Vectores Soportados	70
Tabla N° 17. Métricas para los Vectores soportados	71
Tabla N° 18. Resultados de K-Vecinos	72
Tabla N° 19. Métricas para K-Vecinos	73
Tabla N° 20. Métricas de todos los algoritmos	74

Tabla N° 21. Métricas obtenidas al usar los datos con todos los sensores (Sin reducción de dimensiones)	75
Tabla N° 22. Estructura de los datos de entrenamiento hasta este punto. .	77
Tabla N° 23. Estructura de los datos de entrenamiento con la columna RUL.	78
Tabla N° 24. Estructura de los datos de entrenamiento del archivo 1 con la columna RUL y RUL_norm.	78
Tabla N° 25. Resultados de los algoritmos de Regresión	79
Tabla N° 26. Métricas de todos los algoritmos de regresión	79
Tabla N° 27. Métricas de todos los algoritmos de regresión al usar la base de datos completa	80
Tabla N° 28. Métricas del entrenamiento de la RNN	83
Tabla N° 29. Resultados de la RNN	84
Tabla N° 30. Métricas del prueba de la RNN	84

CAPÍTULO I

INTRODUCCIÓN

En el campo de la ingeniería es importante comprender los diferentes sistemas y sus componentes, no solo en cómo se desempeñan actualmente, sino también en cómo su rendimiento se degrada con el tiempo. Lo anterior se extiende al campo de Pronósticos y Diagnósticos de la Salud (PHM, por sus siglas en inglés), que intenta predecir el futuro de un sistema o componente, basado en sus estados pasados y presentes. Un problema común en este campo es la estimación de la vida útil remanente, o la duración de la funcionalidad de un sistema o componente.

El mantenimiento basado en la condición es una estrategia que el campo PHM emplea para la toma de decisiones, realizando diagnósticos en tiempo real de fallas inminentes y el pronóstico de la salud futura del equipo, donde la decisión de realizar el mantenimiento se alcanza al observar la "condición" del sistema y sus componentes. La condición de un sistema se cuantifica mediante la obtención de datos de varios sensores en el sistema de forma periódica o incluso continua. El mantenimiento basado en la condición intenta evitar tareas de mantenimiento innecesarias realizando acciones de mantenimiento solo cuando hay evidencia de comportamientos anómalos de un activo físico. Es un proceso proactivo, que requiere el desarrollo de un modelo predictivo que pueda activar la alarma para el mantenimiento correspondiente. En general, el mantenimiento basado en condiciones es un proceso que implica dos tareas principales: diagnósticos y pronósticos. El diagnóstico es el proceso de identificación de fallas, mientras que el pronóstico es el proceso de predicción del tiempo hasta el fallo. El tiempo que queda antes de que el equipo falle se conoce como la vida útil remanente del equipo (RUL, por sus siglas en inglés).

Por esta razón, el uso de la información proveniente de sensores que recopilan constantemente datos sobre el estado de turbomaquinaria ha promovido el rápido desarrollo de soluciones basadas en datos en ingeniería. La identificación de patrones, anomalías y detección de fallas, con niveles aceptables de precisión y confiabilidad, son ejemplos de problemas complejos en esta área. Sin embargo, las técnicas de aprendizaje automático han demostrado su utilidad para ayudar en el análisis de los datos de salud de las turbomáquinas. Por lo tanto, el propósito de

este trabajo especial de grado es realizar un análisis detallado en la degradación de una turbomáquina empleando técnicas de aprendizaje automático.

Así pues, se emplearon algoritmos de aprendizaje automático enfocados a la clasificación de valores normales y valores "degradados" para detectar las fallas en el motor; así como también, en algoritmos de regresión y redes neuronales que permitieran predecir el tiempo de vida útil remanente.

CAPÍTULO II

FUNDAMENTO DE LA INVESTIGACIÓN

II.1 PLANTEAMIENTO DEL PROBLEMA

La seguridad, la economía y el rendimiento de la operación de una aeronave depende en gran medida de sus motores. El diagnóstico y el pronóstico de los motores aeronáuticos han sido un tema de interés en los últimos años debido a la creciente demanda de operaciones confiables de estos sistemas. En general, la detección y diagnóstico de fallas es una técnica para comprender si hay fallas presentes en el motor, determinar su ubicación y la gravedad de sus componentes. El pronóstico de fallas es la capacidad de predecir la salud futura del componente del sistema antes de que falle. El diagnóstico de fallas y el pronóstico dan como resultado importantes decisiones de mantenimiento basadas en la condición del equipo, para reducir los costos de mantenimiento debido a reemplazos innecesarios de componentes o paradas. En una industria aeronáutica, los costos relacionados con los motores turbofan implican una gran parte del costo operativo de una aeronave, por lo que el diagnóstico y pronóstico de fallas permite evitar los altos costos de una falla o revisión del sistema. El objetivo general del pronóstico es mejorar la confiabilidad, la seguridad y la disponibilidad.

Con base en lo anterior, el mantenimiento regular y el diagnóstico físico de fallas de una turbomáquina son los dos métodos tradicionales que se emplean para asegurar la operación continua de propulsión de una aeronave. Desafortunadamente, no pueden detectar fallas con anticipación para evitar y/o reparar cualquier defecto del equipo a tiempo.

Por esta razón, en este proyecto se plantea crear una herramienta útil para los analistas e ingenieros, que provea la información cuantitativa necesaria para predecir el estado de un turbofan y definir estrategias para asegurar su operación continua, combinando el uso de herramientas de machine learning que permitan diagnosticar y detectar las fallas, en conjunto con las redes neuronales artificiales para predecir el índice de degradación y tiempo de vida útil remanente.

II.2 ANTECEDENTES

En esta sección se presenta la revisión bibliográfica que se realizó para conocer los avances que se han hecho en el área de optimización en tiempo real aplicada en la reconciliación de datos de procesos industriales con la finalidad de actualizar los parámetros claves que describen dichos procesos.

Tabakoff et al. (1989) investigaron acerca de la degradación del rendimiento aerodinámico básico de un motor de turbina de gas, debido al flujo de partículas y determinaron que esta degradación se puede dividir en dos categorías:

1. Pérdida temporal del rendimiento durante el período de ingestión de partículas.
2. Pérdida permanente de rendimiento debido a la erosión de los álabe.

El enfoque correspondió al efecto de la erosión para compresores de una etapa y multi-etapa de la turbina, y para ello se valieron de pruebas de rendimiento de los compresores.

Los datos experimentales obtenidos de las pruebas de rendimiento de los compresores de una etapa antes y después de la erosión, se usaron para probar un modelo de falla para representar la erosión. Este modelo se implementó en un programa de apilamiento desarrollado para demostrar el efecto de la erosión en un compresor multi-etapa. Este método tiene las ventajas de introducir fallas en las etapas individuales, de modo que su efecto acumulativo en todo el compresor se puede ver mediante el apilamiento de etapas.

Como resultados, obtuvieron que la simulación indicó que la erosión ocasiona que disminuya tanto la presión como el caudal másico. En contraste, en los compresores de varias etapas encontraron que esta reducción era una función de la velocidad del compresor y la ubicación de la etapa donde se produce la erosión. Así mismo, la reducción en el índice de flujo másico aumenta con la velocidad y la primera etapa afecta el rendimiento general del compresor más que las etapas posteriores.

Lazzaretto y Toffolo (2001) presentaron un modelo de una turbina de gas tanto en condiciones de diseño (design) como fuera de diseño (off-design) para su uso en aplicaciones industriales, mediante la construcción de mapas de la máquina a través del uso de técnicas de escalamiento apropiadas y la aplicación de redes neuronales artificiales.

La metodología empleada consistió en 5 etapas:

- (1) Obtención del mapa de rendimiento del compresor axial de la turbina de gas mediante la literatura y de un modelo analítico capaz de predecir el comportamiento fuera de diseño.
- (2) Escalar el mapa generalizado del compresor a través de técnicas apropiadas de corrección usando factores de escalamiento (design model).
- (3) Establecer un modelo analítico para obtener las modificaciones que sufre el mapa del compresor a través de un simulador comercial (off-design).
- (4) Entrenar las redes neuronales con los resultados obtenidos del simulador comercial y analizar su capacidad como herramienta para predecir el rendimiento de la máquina.
- (5) Aplicar las redes neuronales ya entrenadas con un conjunto de datos conocidos y evaluar su capacidad de predicción de las variables de salida.

Como producto de la investigación, determinaron que el modelo analítico arrojó resultados precisos a pesar de que solo se consideraron las características termodinámicas de los componentes sin simular el campo de flujo interno. Este enfoque, sin embargo, no permite tener en cuenta los efectos de las variaciones de los parámetros internos en el rendimiento.

Las redes neuronales feed-forward empleadas demostraron capacidad para reproducir de manera efectiva los parámetros reales de trabajo con errores de predicción de aproximadamente 1% para todas las variables consideradas.

Li y Nilkitsaranont (2009). Proponen un enfoque de diagnóstico y pronóstico mediante la incorporación del conocimiento pronostico-estadístico y de técnicas de regresión, que incluye modelos lineales y cuadráticos, para predecir la vida útil remanente de una turbina de gas. Además, emplearon una "prueba de compatibilidad" estadística para determinar el punto de transición de una regresión lineal a una regresión cuadrática.

Para que el enfoque descrito en este estudio sea aplicable, aplicaron las siguientes suposiciones:

- (1) Solo analizaron la degradación suave asociada con el cambio de rendimiento (como el ensuciamiento y la erosión) que se desarrolla gradualmente a lo largo del tiempo.
- (2) La degradación del rendimiento de la turbina de gas sigue un patrón de fallas

donde, esta tasa, permanece constante por un período de tiempo seguido por una tasa de falla creciente. Esta suposición también cubre el escenario donde solo ocurre una tasa de falla constante o una tasa de falla creciente.

Los resultados del trabajo permitieron determinar que el modelo de regresión combinado es capaz de proporcionar un buen ajuste a los datos históricos de salud del equipo con patrones de falla variables y suministrar una predicción satisfactoria de la degradación del motor, así como también, se demostró que una "prueba de compatibilidad" estadística es útil para la determinación de un punto de transición del modelo de regresión lineal al cuadrático. Finalmente, el enfoque propuesto tiene un gran potencial para proveer una estimación valiosa de la vida útil restante del motor y ayudar a los usuarios de turbinas de gases en sus actividades de mantenimiento basadas en la condición del equipo.

En 2007, Wu et al. desarrollaron un sistema integrado basado en redes neuronales para el soporte en la toma de decisiones del mantenimiento predictivo de equipos rotatorios, así como para estimar la vida útil restante del equipo apoyados en la información sensorial recopilada del mechanical condition monitoring sobre el estado de la máquina.

La metodología propuesta en esta investigación se basó en el uso de información de condiciones de monitoreo (condition monitoring) en tiempo real para predecir la vida útil de la máquina y determinar la política óptima de reemplazo de componentes empleando datos históricos basados en vibraciones. Además, suponen que la condición de una máquina rotativa depende principalmente de la condición de los rodamientos. En resumen:

- (1) Preparación del sistema offline: adquisición de datos sensoriales (condition monitoring).
- (2) Predicción de vida útil: desarrollo y entrenamiento del modelo de red neuronal. El modelo predice la condición del rodamiento al proporcionar un porcentaje de vida estimado del rodamiento.
- (3) Modelado de costos: Finalmente, dado el porcentaje de vida residual predicho, el sistema inteligente proporciona al responsable de la toma de decisiones un tiempo de reemplazo óptimo con la tasa de costo de mantenimiento mínimo.

Mediante la integración del monitoreo de condiciones, la red neuronal, el análisis

y la optimización de costos, el sistema de apoyo de decisión inteligente propuesto transfiere las condiciones de la máquina a los programas de mantenimiento. Con la capacidad de monitorear, predecir y programar simultáneamente, concluyeron que el sistema puede optimizar la programación de mantenimiento al minimizar el costo de mantenimiento esperado por unidad de tiempo.

Zheng et al. (2017) emplearon redes neuronales recurrentes con neuronas del tipo LSTM (Long Short-Term Memory) para determinar el tiempo de vida útil remanente de un turbofan, haciendo uso de cuatro bases de datos del equipo provistos por la NASA. Adicionalmente, compararon sus resultados con otras investigaciones donde, empleando las mismas base de datos, usaron otro tipo de redes neuronales, como por ejemplo: redes neuronales convolucionales (CNN), redes neuronales recurrentes convencionales (RNN) y métodos de modelado estadístico como el modelo de Markov.

Propusieron distintas arquitecturas para evaluar las redes neuronales en todas las bases de datos de la NASA. Sin embargo, la que mejor arrojó resultados fue la arquitectura LSTM(32, 64); NN(8, 8), es decir, 2 capas LSTM con 32 y 64 neuronas respectivamente; Seguidas de 2 capas del tipo perceptrón con 8 neuronas cada una.

Sus resultados los compararon con otras investigaciones y llegaron a las siguientes conclusiones:

- 1 Obtuvieron mejoras significativas prediciendo el tiempo de vida útil remanente al usar las neuronas LSTM en comparación al resto de redes neuronales de las investigaciones. Concretamente obtuvieron 12.52%, 19.15%, 18.37% y 3.40% de mejora para la base de datos 1, 2, 3 y 4 respectivamente.
- 2 Determinaron que al aumentar el tamaño de la ventana de operación para la secuencia usada por la red neuronal, se mejoran significativamente las métricas debido a que una mayor ventana incluye mayor cantidad de patrones que la red es capaz de detectar.

Finalmente, Qi et al. (2018) propusieron un sistema de diagnóstico de fallas para compresores reciprocatantes basado en técnicas de aprendizaje automático para el análisis de datos y el diagnóstico de fallas. El sistema de diagnóstico de fallas en tiempo real se usó para reducir los costos de mantenimiento y mejorar la eficiencia de trabajo de los equipos.

El sistema propuesto se evaluó utilizando datos operativos de 300 Terabytes (TB) de los campos petrolíferos. El 50% de los datos originales se procesaron mediante eliminación de ruido, codificación y clasificación empleando las maquinas de vectores de soporte (SVM) para el entrenamiento.

Como conclusiones, se puede destacar que:

- 1 Aunque la solución propuesta tuvo un buen rendimiento en el diagnóstico de fallas de compresores reciproantes, la tasa de identificación de fallas del 80% aún deja mucho margen de mejora. Para aumentar la precisión de la clasificación de datos, se debe mejorar el proceso de eliminación de ruidos y el aprendizaje del algoritmo.
- 2 El modelo propuesto es robusto al impacto de los cambios en el entorno externo y puede determinar el estado normal de funcionamiento del compresor reciproante para garantizar la alta precisión de la identificación de fallas.

II.3 OBJETIVOS

En base a la información presentada, en este trabajo de investigación se plantea los siguientes objetivos:

II.3.1 Objetivo General

Predecir el índice de degradación del desempeño de un turbofan y su tiempo de vida útil remanente a partir de datos de operación, empleando técnicas de aprendizaje automático.

II.3.2 Objetivos Específicos

Para alcanzar el objetivo general antes mencionado, se establecieron los siguientes objetivos específicos:

- (a) Investigar sobre el mecanismo y el modelado del proceso de degradación del desempeño de un turbofan enfocado en aplicaciones de pronóstico.
- (b) Desarrollar un modelo para detección de falla para una turbomáquina a partir de datos típicos de operación.
- (c) Desarrollar un modelo para estimar la degradación del desempeño de una

turbomáquina a partir de datos típicos de operación.

- (d) Caracterizar el proceso de degradación empleando los parámetros: índice de degradación y tiempo de vida útil remanente.

CAPÍTULO III

MARCO TEÓRICO

En este capítulo se describen las bases teóricas fundamentales para este Trabajo Especial de Grado.

III.1 Motores aeronáuticos

Para mover un avión por el aire, el empuje es generado por algún tipo de sistema de propulsión, conocidos como motores aeronáuticos. Desde los motores de combustión interna utilizados por los hermanos Wright en su primer vuelo, hasta los cohetes que impulsan las naves espaciales, pasando por los motores a reacción utilizados por la aviación comercial. Hoy en día, se ha investigado y desarrollado numerosas tecnologías con el fin de mejorar todos estos sistemas de propulsión.

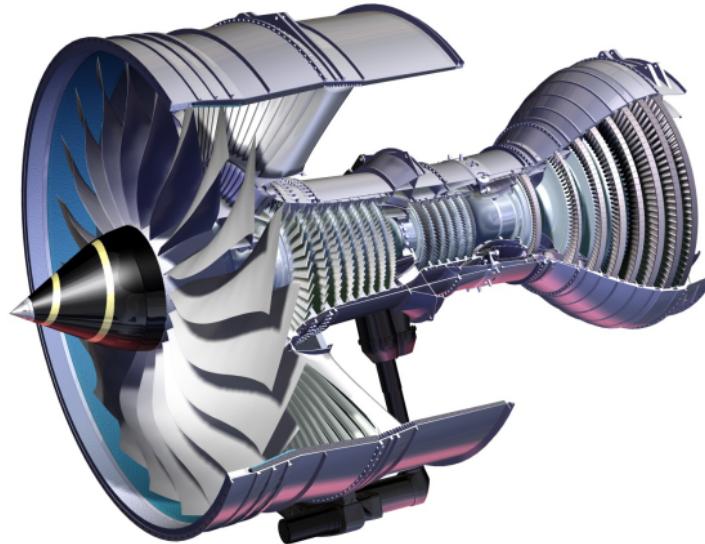


Figura N° 1. Motor aeronáutico. Tomado de www.wikipedia.org.

Los tipos de motores que se han utilizado a lo largo de la historia se pueden clasificar de la siguiente:

III.1.1 Clasificación de los motores aeronáuticos

- (1) Compresión dinámica: debido a la velocidad del aire.
 - (a) Ramjet: operan en la gamma de velocidades de entre 3.0 y 6.0 Mach y La compresión de estos motores es subsónica.
 - (b) Scramjet: la compresión es supersónica. Son motores en que la combustión es instantánea (utiliza hidrógeno como combustible, ya que no crea fricción). La velocidad límite es de 20.0 Mach.
- (2) Compresión estática: contienen compresores.
 - (a) Turbofan: son propulsores en los que primera etapa de compresión es sustituida por un gran ventilador que es el que proporciona la mayor parte del empuje. Este tipo de motores es el más utilizado en aviones comerciales por su eficiencia y bajo consumo de combustible relativo frente a otras opciones (Leyva Leyva).

III.2 Turbofan

Un motor turbofan es la variación más moderna del motor de turbina de gas básico. Al igual que con otras turbinas de gas, hay un motor central. En el motor turbofan básico, el motor central está rodeado por un ventilador en la parte delantera, un compresor, cámara de combustión y una turbina adicional en la parte trasera. En los turbofan modernos sólo una pequeña parte del aire que sale del fan entra en el núcleo central para promover la combustión. El resto pasa por el conducto secundario (by-pass duct) donde se volverá a encontrar con los gases del flujo primario después de la turbina.

Los alabes del fan son muy eficientes y generan una cantidad de empuje destacable sin necesidad de combustión.

Principales ventajas de un turbofan moderno:

- Reduce el consumo de combustible.
- Mayor eficiencia impulsiva. Se empuja una mayor masa de aire hacia el compresor a menor velocidad (mayor presión).
- Menor ruido. El aire sale de la tobera a menor velocidad y genera menos

fricción con la atmósfera.

- El flujo de aire secundario refrigerá el motor.

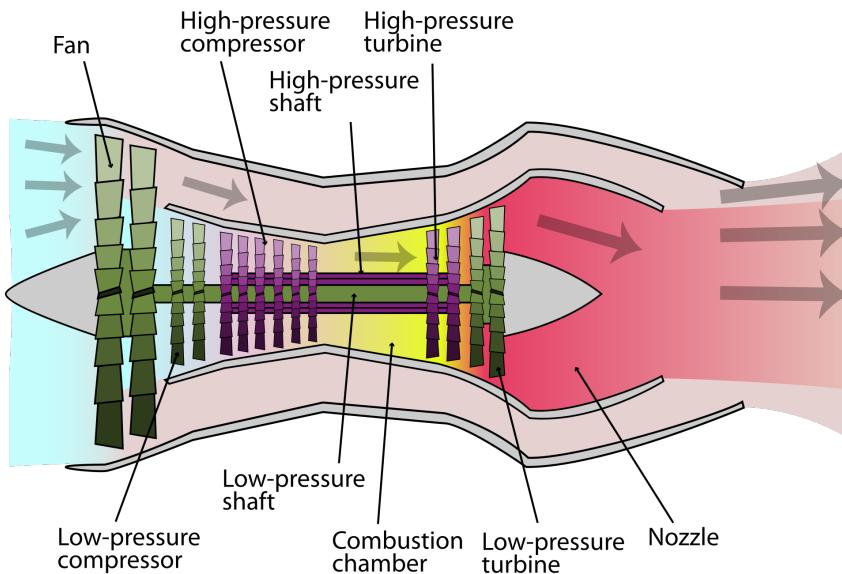


Figura N° 2. Partes de un Turbofan. Tomado de www.wikipedia.org.

En un turbofan, solamente a una parte del aire que penetra en el ventilador se le somete a combustión (a este aire se le denomina flujo primario). La otra parte del aire que entra en el ventilador (flujo secundario) no se le somete a combustión, con la finalidad de acelerarlo únicamente por la acción de los álabes de un compresor. Con este sistema se obtiene una variación de la cantidad de movimiento significativa (Granero, 2012).

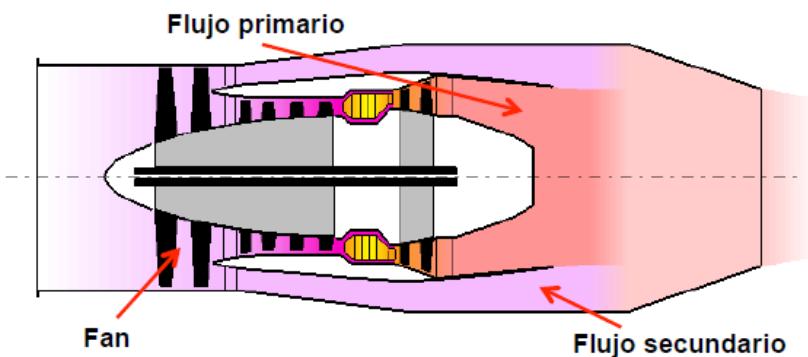


Figura N° 3. Flujos de aire en un turbofan. Tomado de www.wikipedia.org.

III.2.1 Principio de funcionamiento

A continuación se realiza una descripción de las partes y el funcionamiento de un turbofan genérico como se muestra en la Figura N° 4.

1. Toma dinámica: es la entrada de aire del motor. Se sitúa en la parte delantera y conduce el aire hasta el fan. Su objetivo es que entre la mayor cantidad de aire posible en el motor minimizando las pérdidas debidas a gradientes de presión.
2. Fan: es la primera etapa del compresor, la cual posee unas características especiales. Consta de un disco con álabes unido al eje del motor. Recibe el flujo de la toma dinámica y lo comprime en una única etapa, en la cual envía una parte del fluido hacia el interior del motor y el resto lo conduce hacia la tobera por fuera del motor, produciendo de esta forma el empuje. Dependiendo del tamaño, se pueden distinguir dos tipos: (Richter, 2011).
 - De baja relación de derivación (Low by-pass ratio): el fan es de tamaño pequeño respecto al diámetro del equipo. La parte de fluido que deriva fuera del motor es pequeña (una fracción respecto al que envía al interior del motor, varia entre 0,3 y 2). Los turbofans de bajo bypass funcionan bien a velocidades aéreas muy altas, lo que explica su elección para aviones militares, donde la economía de combustible no es la mayor prioridad.
 - De alta relación de derivación (High by-pass ratio): el fan es de un gran tamaño si lo consideramos respecto al diámetro del motor. La relación de derivación en este tipo de motores oscila entre 4 y 11. Los motores de alto bypass tienen valores de empuje de despegue muy grandes, lo que les permite levantar aeronaves pesadas con distancias de despegue relativamente cortas (aviones comerciales modernos).

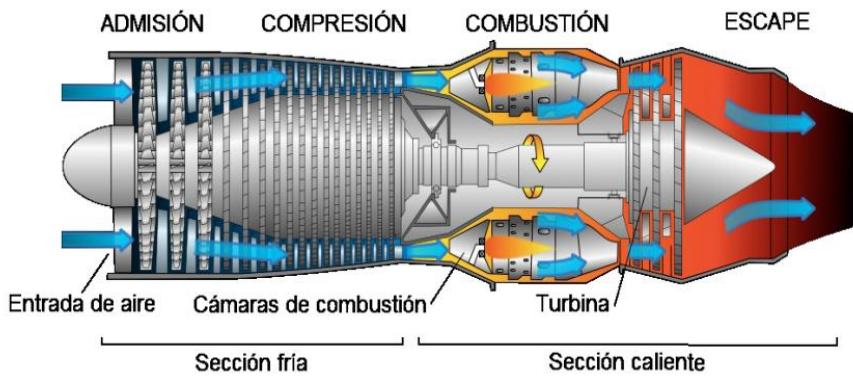


Figura N° 4. Turbofan. (Ernest Vilajosana, 2011)

3. Compresor: es la parte donde se aumenta la presión del aire antes de su combustión. Está formada por varias etapas, cada una de las cuales comprime el aire una fracción del total. Cada etapa la compone una pareja de discos, una de las cuales está fija a las paredes del motor (estátor) y otra está unida al eje (rotor). Los discos llevan acoplados una serie de álabes. El compresor puede ser de tipo axial (el aire sigue la dirección del eje del motor) o centrífugo (el aire se desvía en cada etapa en dirección perpendicular al eje y posteriormente se reconduce). La relación de compresión determina el aumento relativo de la presión respecto a la que había a la entrada del compresor.
4. Cámara de combustión: Es el lugar donde se mezcla el aire previamente comprimido con el combustible y se produce la ignición de la mezcla. De esta forma se aumenta la energía que posee el fluido, que se transmite a las siguientes etapas donde se aprovecha para producir empuje. El combustible se introduce por una serie de inyectores distribuidos a lo largo de la cámara.
5. Turbina: es la zona donde se obtiene energía del fluido con el fin de mover el compresor. Consta de varias etapas, en cada una de las cuales se extrae una fracción de la energía que sale desde la cámara de combustión. Estas etapas están formadas por un disco acoplado al eje del motor con una serie de álabes. En este tipo de motores, la turbina se suele dividir en dos partes: turbina de alta presión, que proporciona energía a las últimas etapas del

compresor, y turbina de baja presión, que proporciona energía al fan y a las primeras etapas del compresor. Cada una de las partes va asociada a un eje independiente, por lo que podemos ver que este tipo de motores consta de doble eje.

6. Tobera: es la salida del aire del motor. Posee una forma que logra maximizar la velocidad de salida del fluido a costa de disminuir tanto la presión como la temperatura del mismo. Al maximizar la velocidad de salida, conseguimos que el empuje sea el máximo posible.

III.3 Límites de operación

En un turbofan, los componentes individuales tienen sus propias eficiencias, que determinan colectivamente la eficiencia general del equipo. Por lo tanto, es importante describir las eficiencias de los componentes individuales así como los rangos seguros de operación. Tenga en cuenta que, en algunos casos, las condiciones de operación altamente eficientes pueden no corresponder necesariamente a condiciones seguras.

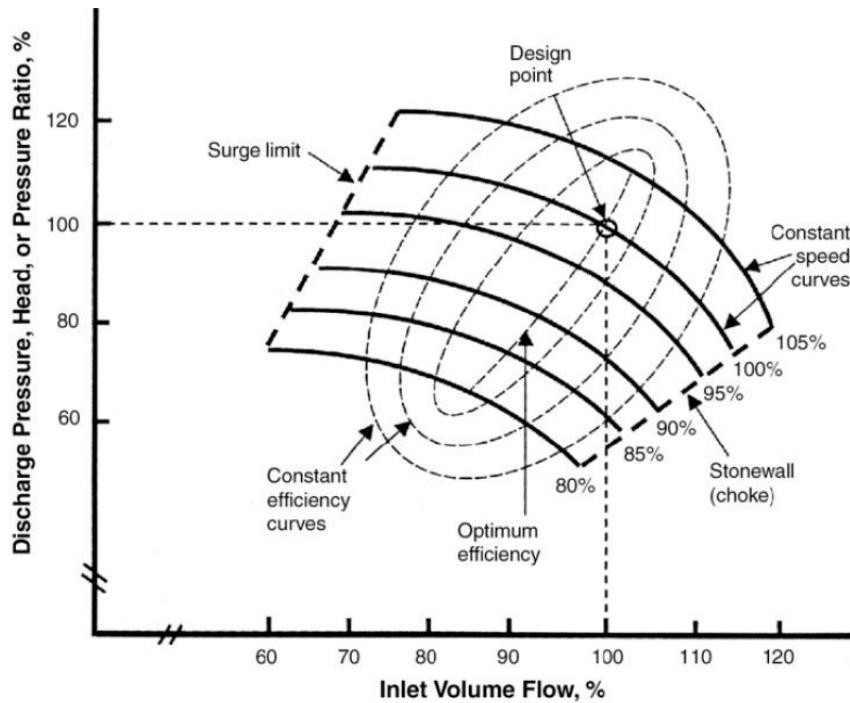


Figura N° 5. Mapa de un compresor. (Gil Tolmo, 2013)

Los límites de eficiencia y operatividad se describen para ventiladores, compresores y turbinas por medio de un mapa de componentes como el que se muestra en la Figura N° 5. Un mapa de algún componente del turbofan se construye operando el componente en varias combinaciones de relación de presión de salida/entrada y una tasa de flujo mÁsico a condiciones de estado estable. Aunque es habitual utilizar formas no dimensionales y corregidas de presión y caudal, la gráfica describe esencialmente la relaciÓn entre la tasa de flujo mÁsico (eje horizontal) y la relaciÓn de presión correspondiente (eje vertical) (Richter, 2011).

III.4 DegradaciÓn del desempeño de un compresor

Según González (2015), el deterioro en compresores es inevitable. Estos equipos operan en un amplio rango de temperaturas, velocidades, potencias y condiciones de operaciÓn, lo que provoca que sus componentes se deterioren con el tiempo y el rendimiento del equipo disminuya.

En un compresor centrífugo pueden darse distintos tipos de degradación, los cuales corresponden a fenómenos físicos que ocurren en distintos componentes de la máquina y que provocan que su rendimiento disminuya (Figura N° 6).

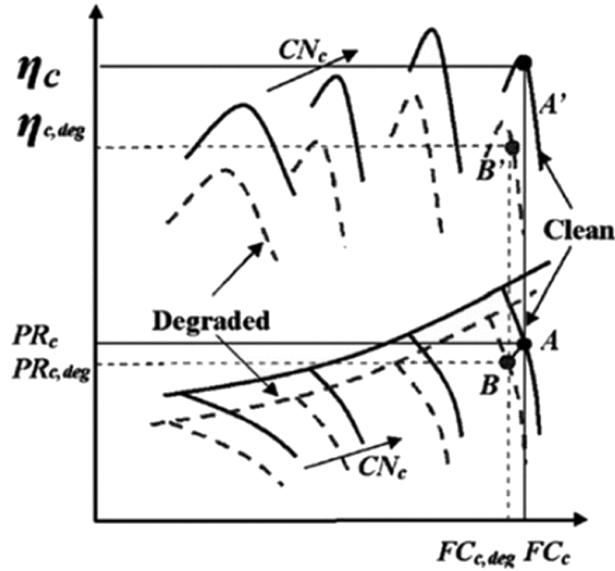


Figura N° 6. Degradación del desempeño de un compresor (Li, 2017).

La degradación produce tres efectos en el compresor principalmente: cambios en los perfiles aerodinámicos de los álabes, aumento de las holguras en la punta de los álabes, y rugosidad superficial.

La rugosidad superficial producida por efecto del ensuciamiento, erosión o corrosión provoca un aumento de las pérdidas por fricción y adelantan la formación de régimen turbulento en la capa límite. La pérdida de material en las puntas de los álabes o en el borde de ataque debido a erosión o corrosión provocan una alteración en los ángulos y en las velocidades de salida del fluido en cada etapa, cambiando a un estado que no es el de diseño.

La degradación de una etapa tiene efectos negativos en la siguiente, ya que el fluido pasará de una a otra con un ángulo de ataque diferente, a menor presión y a mayor temperatura. Por tanto, el motor trabajará en su conjunto en un punto de diseño diferente al óptimo, reduciendo su eficiencia.

Además, la degradación del compresor disminuye el margen surge, por lo que la línea de funcionamiento del compresor podría ubicarse demasiada cerca de la línea

surge (Li, 2017).

III.4.1 Tipos de degradación

Existen varios estudios sobre la degradación, aunque hay una limitación en la información debido al alto coste que implican los test experimentales sobre este tema (Syverud y Bakken, 2005).

En la literatura se puede encontrar la degradación clasificada de diferentes formas. La clasificación más general es la que realiza Martín-Aragón Gutiérrez (2016):

- Degradación del rendimiento.
- Degradación mecánica.

Se puede profundizar en esta clasificación dividiendo las degradaciones anteriores en nuevas categorías en función del tipo de daño causado (Meher-Homji et al., 2004):

- Deterioro recuperable: se puede eliminar mediante ciertas acciones sobre el compresor en operación.
- Deterioro no recuperable: solo puede ser eliminado mediante una parada y revisión general.
- Deterioro permanente: no puede eliminarse. Este tipo de deterioro se presenta cuando la maquinaria se pone en operación y ya hay alguna parte del motor dañada.

Meher-Homji et al. (2004) proporcionan en Figura N° 7 un esquema en el que se presentan algunos de los posibles deterioros que pueden aparecer.

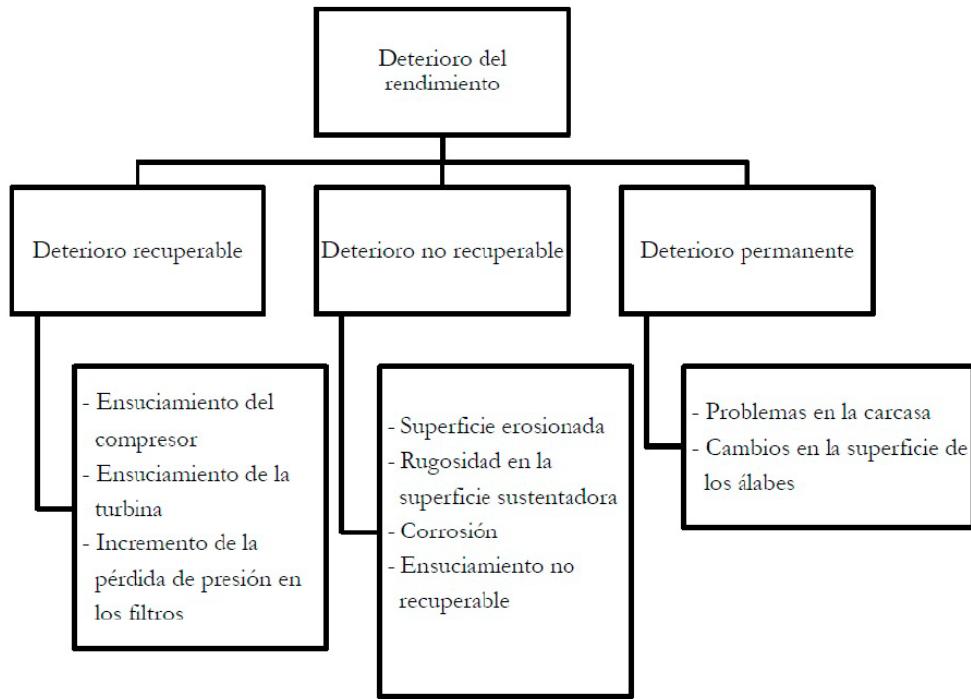


Figura N° 7. Posibles deterioros en turbo-máquinas (Meher-Homji et al., 2004).

III.4.2 Mecanismos de deterioro

Existen diferentes tipos de deterioro que han sido estudiados durante las últimas décadas y que se recogen en diferentes trabajos de investigación. A continuación se va a hacer un resumen de los mismos.

Uno de los mecanismos de deterioro que se da en los compresores es la **corrosión** (ver Figura N° 8). Según Kurz y Brun (2001), la corrosión es causada por los contaminantes que entran con el fluido. La corrosión es a menudo causada por sales como sodio y potasio en combinación con sulfuros.

La **erosión** es otro mecanismo de degradación. Tabakoff et al. (1989) lo definen como el resultado de la fricción de un elemento móvil con una superficie estática. Muchas máquinas usan superficies que son susceptibles de abrasión, donde se produce cierta cantidad de rozamiento durante el funcionamiento de la máquina. El material eliminado incrementaría los espacios entre ambas piezas. Parte de esta degradación depende del tiempo de vida de la máquina.



Figura N° 8. Compresor multi-etapa afectado por corrosión. Tomado de <https://bit.ly/2KjZCJa>

Finalmente se puede citar el **ensuciamiento** del compresor como mecanismo de deterioro. Durante el funcionamiento del compresor, las palas del impulsor pueden ensuciarse. Este ensuciamiento hace que la velocidad relativa aumente, y que la velocidad tangencial disminuya (se produce una disminución del flujo). Por lo que la altura (o energía) en condiciones de ensuciamiento disminuirá (Gil Tolmo, 2013).



Figura N° 9. Ensuciamiento que puede tener lugar en las palas del impulsor (Gil Tolmo, 2013).

Además, se producirá una disminución de la eficiencia y el punto de operación estará más cercano a la línea de *surge*.

Las causas que pueden originar un ensuciamiento son: (a) Partículas suspendidas en el fluido. (b) Partículas arrastradas durante la aspiración. El ensuciamiento no

tiene porque ocurrir solamente en la primera etapa, puede tener lugar en cualquiera de ellas.



Figura N° 10. Impulsor afectado por ensuciamiento. Tomado de "Turbomachinery Magazine" (2015).

III.5 Aprendizaje automático (machine learning)

Es una disciplina científica del ámbito de la Inteligencia Artificial que crea sistemas que aprenden automáticamente. *Aprender* en este contexto se refiere a identificar patrones complejos en millones de datos (*Big Data*). La máquina que realmente aprende es un algoritmo que revisa los datos, determina las relaciones entre las entradas-salidas y es capaz de predecir comportamientos futuros. *Automáticamente*, también en este contexto, implica que estos sistemas se mejoran de forma autónoma con el tiempo, sin intervención humana (Raschka, 2015).

Básicamente, el *aprendizaje automático* evolucionó como un subcampo de la Inteligencia Artificial (IA) que involucra algoritmos de auto-aprendizaje que detectan patrones de los datos para hacer predicciones.

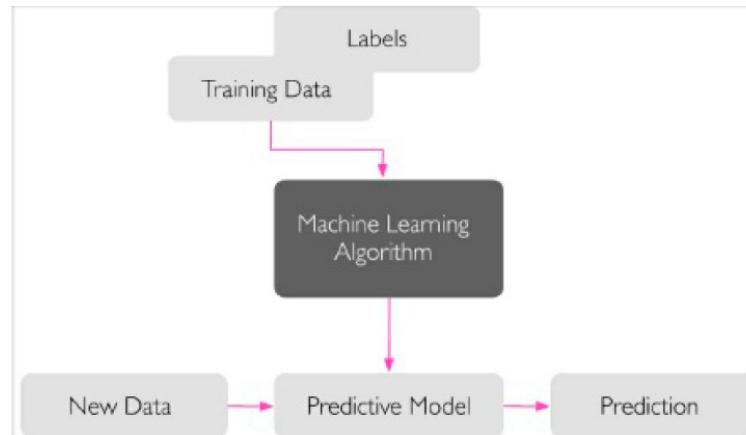


Figura N° 11. Representación básica del aprendizaje automático (Raschka, 2015).

En lugar de requerir que los humanos deriven manualmente reglas y construyan modelos a partir del análisis de grandes cantidades de datos, el aprendizaje automático ofrece una alternativa más eficiente para capturar el conocimiento en datos para mejorar gradualmente el rendimiento de los modelos predictivos y tomar decisiones basadas en datos.

El *aprendizaje automático* no solo es cada vez más importante en la investigación en informática, sino que también desempeña un papel importante en nuestra vida cotidiana. Gracias al aprendizaje automático, disfrutamos de sólidos filtros de correo electrónico no deseado, texto conveniente, software de reconocimiento de voz, motores de búsqueda de webs confiables, desafiantes programas de ajedrez y, con suerte, pronto, autos seguros y eficientes.

El *aprendizaje automático* se puede aplicar empleando múltiples lenguajes de programación. Hoy en día, la batuta en esta rama la tiene Python, debido a su facilidad de sintaxis y versatilidad en aportar librerías muy poderosas enfocadas especialmente al machine learning.

III.6 Entrenamiento de modelos de machine learning

El proceso de entrenamiento consiste en proporcionar datos de los cuales aprende un algoritmo de ML. Básicamente, cuando se entrena el algoritmo, se crea el modelo que es capaz de determinar los patrones y relaciones entre las variables de

entrada y salida (Raschka, 2015).

Los datos de entrenamiento deben contener la respuesta correcta, que se conoce como destino. El algoritmo de aprendizaje, encuentra patrones en los datos de entrenamiento que asignan los atributos de los datos de entrada al destino (la respuesta que desea predecir) y genera un modelo de ML que captura dichos patrones.

Puede utilizar el modelo de ML para obtener predicciones sobre datos nuevos para los que no se conoce la respuesta de destino. Por ejemplo, si desea entrenar un algoritmo de ML para que prediga si un mensaje de correo electrónico es spam o no, le proporcionaría datos de entrenamiento que contienen correos electrónicos para los que conoce el destino (es decir, una etiqueta que indica si un mensaje es spam o no). El algoritmo entrenaría un modelo de ML mediante la utilización de estos datos, lo que se traduce en un modelo que intenta predecir si los correos electrónicos nuevos son spam o no.

III.7 Tipos de aprendizajes en machine learning

El *aprendizaje automático* se divide en dos áreas principales: *aprendizaje supervisado* y *aprendizaje no supervisado*. Aunque pueda parecer que el primero se refiere a la predicción con intervención humana y la segunda no, estos dos conceptos tienen más que ver con qué queremos hacer con los datos.

III.7.1 Aprendizaje automático supervisado

Consiste en una cantidad de algoritmos que aprenden a partir de un conjunto de datos de entrenamientos etiquetados (ejemplares) para luego generalizar al conjunto de todas las entradas posibles. Ejemplo de las técnicas de aprendizaje supervisado son: regresión logística, maquinas de vectores soportados, árboles de decisión, etc.

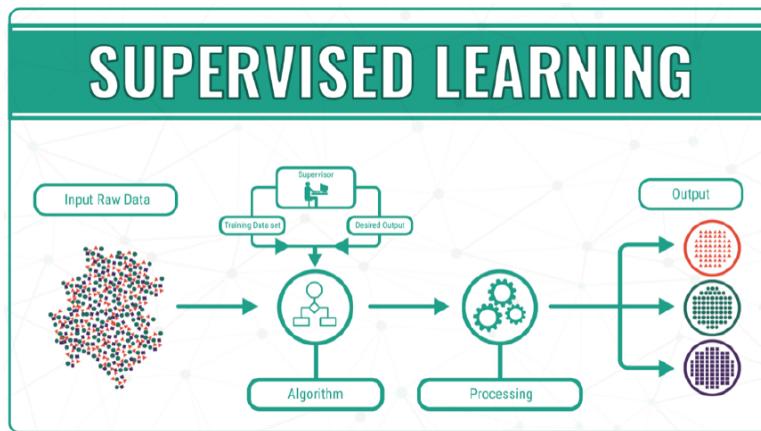


Figura N° 12. Representación básica del aprendizaje supervisado. Tomado de "Data Science Central" (2018).

Se utiliza siempre que se quiere predecir un resultado determinado a partir de una entrada determinada. El objetivo es hacer predicciones precisas para datos nuevos nunca antes vistos (Raschka, 2015).

Existen dos tipos de aprendizaje supervisado: Clasificación y Regresión.

Un sistema de *Clasificación* predice una categoría, mientras que una *Regresión* predice un número. Un ejemplo de *Clasificación* es el correo electrónico. Los correos se “categorizan” como “spam” o como “legítimos”.

La *Regresión*, en cambio, predice un número, como por ejemplo, cuál va a ser el precio de un artículo, o el número de reservas que se harán en un hotel en un mes determinado.

III.7.2 Aprendizaje automático no supervisado

A diferencia del aprendizaje supervisado, el *aprendizaje no supervisado* engloba algoritmos que aprenden de un conjunto de datos no etiquetados. Suelen explorar los datos siguiendo métodos estadísticos, geométricos o criterios de similitud. Entre estas técnicas se tienen: agrupamiento k-means y estimación de densidad kernel (Raschka, 2015).

El *aprendizaje no supervisado* incluye todo tipo de aprendizaje automático en el que no se conoce ningún resultado, nada que instruya el algoritmo de aprendizaje.

En el aprendizaje no supervisado, el algoritmo de aprendizaje solo recibe los datos de entrada y debe encontrar la estructura o patrón desconocido que subyace en dichos datos.

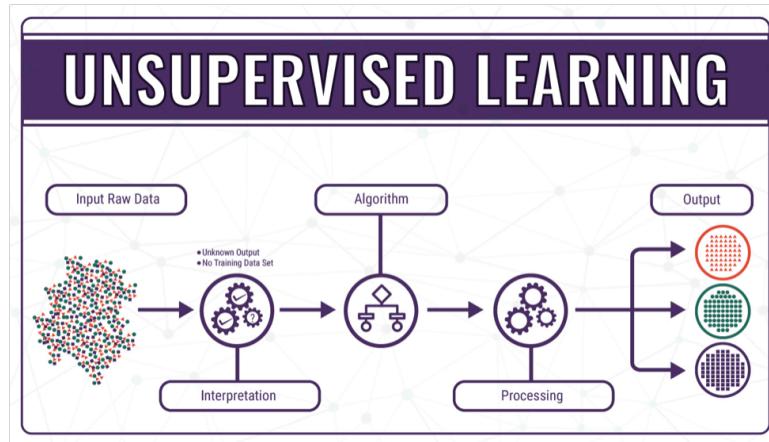


Figura N° 13. Representación básica del aprendizaje no supervisado. Tomado de "Data Science Central" (2018).

Entre los diversos tipos de aprendizaje no supervisado se encuentran:

- Reducción de dimensión del conjunto de datos.
- Agrupamiento (Clustering).
- Detección de valores atípicos (Outliers).
- Detección de novedades.

III.7.3 Aprendizaje profundo (deep learning)

El *aprendizaje profundo* es una forma de aprendizaje automático que permite a las computadoras aprender de la experiencia y comprender el mundo en términos de una jerarquía de conceptos. La jerarquía de conceptos permite que la computadora aprenda conceptos complicados al construirlos a partir de conceptos más simples (Müller y Guido, 2016).

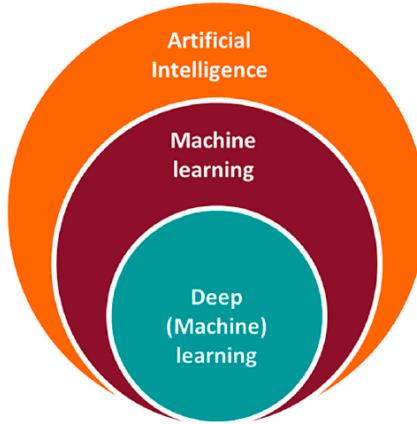


Figura N° 14. Representación de los niveles de la IA. Tomado de "Siemens Health-care" (2016).

El *Deep Learning* lleva a cabo el proceso de Machine Learning usando una red neuronal artificial que se compone de un número de niveles jerárquicos. En el nivel inicial de la jerarquía la red aprende algo simple y luego envía esta información al siguiente nivel. El siguiente nivel toma esta información sencilla, la combina, compone una información algo un poco más compleja, y se lo pasa al tercer nivel, y así sucesivamente.

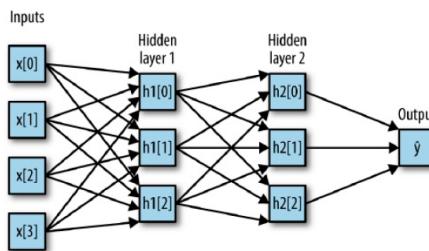


Figura N° 15. Representación gráfica de las RNA (Raschka, 2015).

III.7.4 Clasificación binaria

Tipo de tarea de predicción que da como resultado una de dos clases mutuamente exclusivas. Por ejemplo, un modelo de aprendizaje automático que evalúa mensajes de correo electrónico y da como resultado "es spam" o "no es spam" es un clasificador binario. Generalmente arroja como resultado valores discretos.

En los problemas de clasificación binaria, los datos cuentan con una *etiqueta*, la *clase negativa* y la *clase positiva*.

III.7.4.1 Etiqueta

En el aprendizaje supervisado cada ejemplo de un conjunto de datos etiquetados consiste en uno o más atributos y una etiqueta. Por ejemplo, en un conjunto de datos de casas, los atributos pueden incluir el número de habitaciones, el número de baños y la antigüedad de la propiedad, mientras que la etiqueta puede ser el precio del inmueble. En un conjunto de datos de detección de spam, los atributos pueden incluir el asunto, el remitente y el mensaje de correo electrónico en sí, mientras que la etiqueta probablemente sería "es spam" o "no es spam".

III.7.4.2 Clase negativa

En la clasificación binaria, una clase se expresa como positiva y la otra como negativa. La clase positiva es lo que estamos buscando y la clase negativa es la otra posibilidad. Por ejemplo, la clase negativa en un examen médico puede ser "no es tumor". La clase negativa en un clasificador de correo electrónico puede ser "no es spam".

Cuando el algoritmo de clasificación predice bien esta clase, a dicho resultado se le denomina *Verdadero negativo (VN)*. En caso contrario se le denominará *Falso negativo (FN)*

III.7.4.3 Clase positiva

El resultado positivo es aquello que queremos determinar. Por ejemplo, la clase positiva en un examen médico puede ser "es tumor". La clase positiva en un clasificador de correo electrónico puede ser "es spam".

Cuando el algoritmo de clasificación predice bien esta clase, a dicho resultado se le denomina *Verdadero positivo (VP)*. En caso contrario se le denominará *Falso positivo (FP)*

Además, cuando queremos evaluar un algoritmo de clasificación binaria, los resultados se mostrarán en una matriz que dirá la cantidad de verdaderos negativos,

verdaderos positivos, falsos negativos y falsos positivos que arrojó el modelo. A dicha matriz se le denomina *Matriz de confusión*.

III.7.5 Matriz de confusión

Matriz de $N \times N$ que resume el nivel de éxito de las predicciones de un modelo de clasificación, es decir, la correlación entre la etiqueta y la clasificación del modelo. Un eje de una matriz de confusión es la etiqueta que el modelo predijo; el otro es la etiqueta real. N representa el número de clases. En un problema de clasificación binaria, $N=2$.

Las matrices de confusión contienen información suficiente para calcular una variedad de métricas de rendimiento, incluidas la precisión y la recuperación.

III.7.6 Modelos de regresión

Tipo de modelo que da como resultado valores continuos (generalmente de punto flotante).

III.7.7 Métricas de rendimiento de modelos

Dentro del mundo del modelado de datos, las métricas son una gran herramienta para determinar el rendimiento de un modelo de clasificación y regresión.

- Dentro de las métricas de clasificación tenemos:

III.7.7.1 Exactitud

Fracción de predicciones que se realizaron correctamente en un modelo de clasificación. En la clasificación binaria, la exactitud tiene la siguiente definición:

$$\text{Exactitud} = \frac{VP + VN}{V+VN + FP + FN} \quad (1)$$

Calcularemos la exactitud del siguiente modelo, que clasificó 100 tumores como malignos (la clase positiva) o benignos (la clase negativa):

Verdadero positivo (VP):	Falso positivo (FP):
<ul style="list-style-type: none"> Realidad: Maligno Predicción del modelo de AA: Maligno Número de resultados de VP: 1 	<ul style="list-style-type: none"> Realidad: Benigno Predicción del modelo de AA: Maligno Número de resultados de FP: 1
Falso negativo (FN):	Verdadero negativo (VN):
<ul style="list-style-type: none"> Realidad: Maligno Predicción del modelo de AA: Benigno Número de resultados de FN: 8 	<ul style="list-style-type: none"> Realidad: Benigno Predicción del modelo de AA: Benigno Número de resultados de VN: 90

Figura N° 16. Matriz de confusión para el ejemplo de exactitud. Tomado de: <https://goo.gl/CS5wVg>.

$$\text{Exactitud} = \frac{VP+VN}{V+VN+FP+FN} = \frac{1+90}{1+90+1+8} = 0,91$$

La exactitud resulta ser de 0.91 o 91% (91 predicciones correctas de 100 ejemplos totales).

De los 100 ejemplos de tumores, 91 son benignos (90 VN y 1 FP) y 9 son malignos (1 VP y 8 FN).

De los 91 tumores benignos, el modelo identifica correctamente 90 como benignos. Eso es bueno. Sin embargo, de los 9 tumores malignos, el modelo solo identifica correctamente 1 como maligno; este es un resultado terrible, ya que 8 de 9 malignidades no se diagnostican.

Mientras que un 91% de exactitud podría parecer bueno a primera vista, otro modelo de clasificación de tumores que siempre predijera benigno lograría la misma exactitud (91/100 predicciones correctas) en nuestros ejemplos. En otras palabras, nuestro modelo no es mejor que otro que tenga una capacidad predictiva de cero para distinguir tumores malignos de tumores benignos.

La exactitud sola no muestra el panorama completo cuando se trabaja con un conjunto de datos desequilibrados, como este, donde hay una disparidad significativa entre el número de etiquetas positivas y negativas.

III.7.7.2 Precisión

La precisión responde a la siguiente pregunta: *¿Qué proporción de identificaciones positivas fue correcta?*

La precisión identifica la frecuencia con la que un modelo predijo correctamente la clase positiva. Esto significa lo siguiente:

$$Precision = \frac{VP}{VP + FP} \quad (2)$$

Veamos el siguiente ejemplo:

Calcularemos la precisión ejemplo que se uso en la métrica anterior:

Verdaderos positivos (VP): 1	Falsos positivos (FP): 1
Falsos negativos (FN): 8	Verdaderos negativos (VN): 90

Figura N° 17. Matriz de confusión para el ejemplo de precisión. Tomado de: <https://goo.gl/CS5wVg>.

Donde la precisión se calcularía de la siguiente manera:

$$Precision = \frac{VP}{VP+FP} = \frac{1}{1+1} = 0,5$$

Nuestro modelo tiene una precisión de 0.5, es decir, cuando predice que un tumor es maligno, acierta el 50% de las veces.

III.7.7.3 Recuperación

Métrica que responde a la siguiente pregunta: *de todas las etiquetas positivas posibles, ¿Cuántas identificó correctamente el modelo?*. Esto significa lo siguiente:

$$Recuperacion = \frac{VP}{VP + FN} \quad (3)$$

Si tomamos como ejemplo el que se realizó con la métrica anterior, la recuperación para ese modelo sería la siguiente:

$$Recuperacion = \frac{VP}{VP+FN} = \frac{1}{1+8} = 0,11$$

Nuestro modelo tiene una recuperación de 0.11. Es decir, identifica correctamente el 11% de los tumores malignos.

III.7.7.4 AUC-ROC

La curva ROC representa de la tasa de verdaderos positivos frente a la tasa de falsos positivos en diferentes umbrales de clasificación.

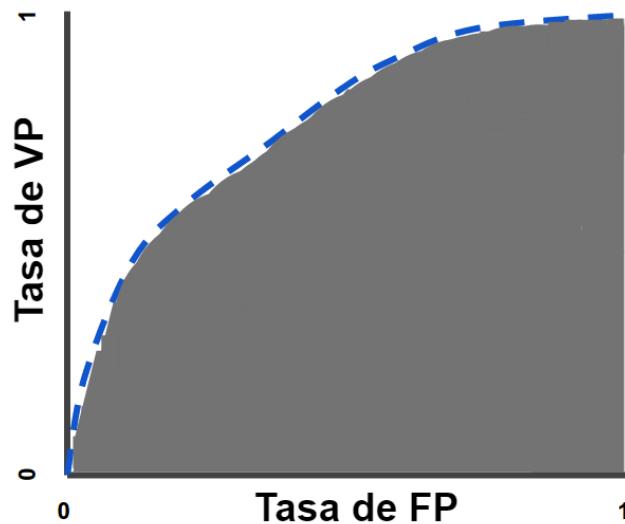


Figura N° 18. Área bajo la curva ROC. Tomado de: <https://goo.gl/NPMeJ6>.

El área bajo la curva (AUC) de la curva ROC es la probabilidad de que un clasificador tenga más seguridad de que un ejemplo positivo elegido al azar sea

realmente positivo con respecto a que un ejemplo negativo elegido al azar sea positivo.

- Dentro de las métricas de regresión tenemos:

III.7.7.5 Error cuadrático medio (MSE)

Promedio de la pérdida al cuadrado de cada ejemplo. El error cuadrático medio se calcula dividiendo la pérdida al cuadrado por el número de ejemplos.

$$MSE = \frac{\sum(y_i - Y_i)^2}{n} \quad (4)$$

Donde: y_i : valor predicho, Y_i : valor real

Al sacarle la raíz al MSE se obtiene la Raíz del Error Cuadrático Medio (RMSE), que es la métrica más empleada en problemas de regresión junto con el coeficiente de determinación R^2 .

el MSE es una métrica negativa, es decir, que el modelo es mucho mejor a medida que la métrica disminuye.

III.7.7.6 Coeficiente de determinación (R^2)

Define la medida de cuántas muestras futuras es probable que prediga el modelo de regresión.

III.7.8 Factor del valor atípico local

El algoritmo de aprendizaje no supervisado, Factor del Valor Atípico Local (LOF, por sus siglas en inglés), se basa en un concepto de densidad local, donde la localidad está dada por los K-Vecinos más cercanos, cuya distancia se utiliza para estimar la densidad. Al comparar la densidad local de un objeto con las densidades locales de sus vecinos, se pueden identificar regiones de densidad similar y puntos que tienen una densidad sustancialmente mayor que sus vecinos. Estos son considerados como valores atípicos.

III.7.9 DBSCAN

Algoritmo de aprendizaje no supervisado *Agrupamiento Espacial Basado en la Densidad de Aplicaciones con Ruido*, DBSCAN (por sus siglas en inglés), basado también en la densidad de puntos, en donde los puntos se clasifican como puntos centrales, puntos alcanzables y valores atípicos

III.7.10 Eliminación recursiva de características (RFE)

Algoritmo de selección de características que se emplea para realizar la reducción de dimensiones. Para utilizar este algoritmo es necesario ajustar un algoritmo de regresión o clasificación externo que le permita, mediante unas serie de métricas, establecer la cantidad de características más representativas que permitan alcanzar el valor óptimo en las métricas.

Funciona de la siguiente manera:

Dado un algoritmo externo que asigna pesos a las características (por ejemplo, los coeficientes de un modelo lineal), el objetivo de la eliminación de características recursivas (RFE) es seleccionar las características considerando recursivamente conjuntos de características cada vez más pequeños. Primero, el estimador se entrena en el conjunto inicial de características. Luego, las características menos importantes se eliminan del conjunto actual de características. Ese procedimiento se repite recursivamente en el conjunto reducido hasta que finalmente se alcanza el número deseado de funciones para seleccionar.

Así que al principio el modelo se ajusta a los datos. Luego tenemos coeficientes de cada característica. Retiramos la(s) característica(s) con menor coeficiente o importancia. Entonces el modelo se ajusta a las características restantes. El proceso se repite hasta que tengamos un número necesario de funciones (o se cumpla algún criterio en función de una métrica establecida).

III.7.11 Red neuronal recurrente con neurona LSTM

La red neuronal recurrente con neuronas LSTM son un tipo especial de RNN, capaz de aprender dependencias a largo plazo. Las LSTM están diseñadas explícitamente para evitar el problema de dependencia a largo plazo. Recordar infor-

mación durante largos períodos de tiempo es prácticamente su comportamiento predeterminado.

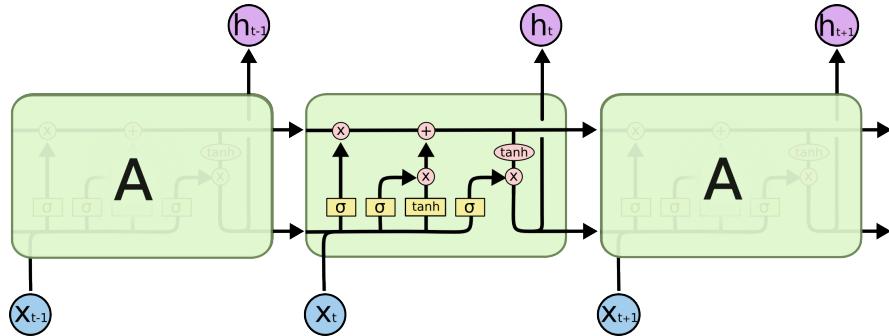


Figura N° 19. El módulo de repetición en un LSTM contiene cuatro capas interactivas. Tomado de: <https://goo.gl/mogahU>.

La clave de la LSTM consiste en sus puertas. Estas también son funciones ponderadas que gobiernan aún más el flujo de información en la capa. Hay tres puertas:

- Puerta del olvido: decide qué información descartar de la celda.
- Puerta de entrada: decide qué valores de la entrada actualizarán el estado de la memoria.
- Puerta de salida: decide la salida en función de la entrada y la memoria de la celda.

Todas las redes neuronales recurrentes (RNN) tienen bucles de retroalimentación en la capa recurrente. Esto les permite mantener la información en "memoria" con el tiempo. Pero, puede ser difícil capacitar a las RNN estándar para resolver problemas que requieren el aprendizaje de dependencias temporales a largo plazo. Esto se debe a que el gradiente de la función de pérdida decae exponencialmente con el tiempo (llamado el problema del gradiente de fuga). Las redes LSTM son un tipo de RNN que utilizan unidades especiales además de las unidades estándar. Las unidades LSTM incluyen una "celda de memoria" que puede mantener la información en la memoria por largos períodos de tiempo. Se utiliza un conjunto de compuertas para controlar cuándo la información ingresa a la memoria, cuándo se envía y cuándo se olvida. Esta arquitectura les permite aprender dependencias a largo plazo (Chung et al., 2014).

III.7.12 Vida útil remanente

Es un período de tiempo durante el cual el sistema, luego de detectada la falla, es utilizable para el propósito que fue diseñado.

III.7.13 Índice de degradación

Es un indicador del progreso de la salud de un activo físico. En pocas palabras, es un indicador capaz de mostrar como, a medida del uso del equipo, su rendimiento se degrada con el tiempo. La disminución gradual de la capacidad de una batería eléctrica para almacenar energía es un ejemplo de un sistema en el modo de degradación. El modo de degradación conduce eventualmente a la falla completa del sistema.

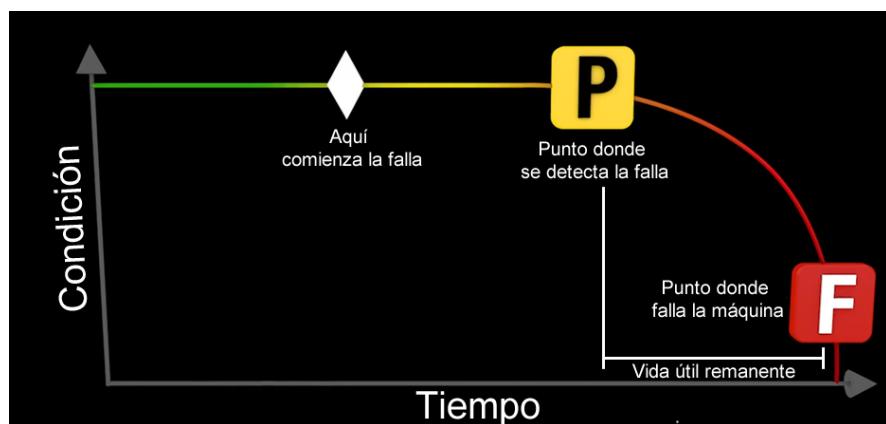


Figura N° 20. el rango P-F será la vida útil remanente.

Es un indicador que se suele usar en el campo de Pronósticos y Diagnósticos de la Salud de activos, porque al representarlo gráficamente, se puede observar cual será el progreso de la salud de la máquina y poder tomar medidas de mantenimiento preventivas.

CAPÍTULO IV

MARCO METODOLÓGICO

En esta sección, se presenta la metodología para alcanzar los objetivos planteados.

IV.1 Investigar sobre el mecanismo y el modelado del proceso de degradación del desempeño de una turbomáquina enfocado en aplicaciones de pronóstico.

Se propone realizar una búsqueda de investigaciones donde se haya estudiado y modelado las degradaciones de motores aeronáuticos. Así mismo, emplear la base de datos que se haya utilizado en alguna de las investigaciones para poder cumplir con el resto de objetivos específicos.

IV.2 Desarrollar un modelo para detección de falla para una turbomáquina a partir de datos típicos de operación.

Para el desarrollo de un modelo de diagnóstico para la detección de fallas, se plantea la siguiente estrategia:

1. Identificar la estructura más adecuada para trabajar con la base de datos de la turbomáquina.
2. Elaborar un análisis estadístico descriptivo de los datos que incluya distribuciones de los datos, métricas estadísticas típicas (desviación estándar, máximo, mínimo, media, etc) y análisis de correlación entre variables.
3. Realizar el procesamiento de los datos. Esto incluye: detectar y retirar los valores atípicos, detectar algún patrón de degradación en los datos.
4. Normalizar los datos luego de realizar el item 3.
5. Identificar los parámetros de procesos que permitirán estimar el estado de salud de la turbomáquina. Estos se denominarán *indicadores clave de rendimiento (KPI)*.
6. Construir un modelo de clasificación que permita llevar a cabo la detección de las fallas utilizando los *indicadores clave de rendimiento (KPI)*.

IV.3 Desarrollar un modelo para estimar la degradación del desempeño de una turbomáquina a partir de datos típicos de operación.

Para construir el modelo de pronóstico de la vida útil remanente, se propone el siguiente esquema:

1. Elaborar un modelo de aprendizaje automático para el pronóstico de la vida útil remanente, empleando los datos salientes del clasificador del objetivo anterior.
2. Empleando un proceso de deep learning, construir una red neuronal artificial para predecir el tiempo de vida útil remanente del equipo.
3. Comparar los resultados del ítem 1 con los obtenidos en el ítem 2 y determinar cuál proceso de pronóstico es más adecuado.
4. Obtener el índice de degradación de la máquina mediante el mejor método del ítem 3.

IV.4 Caracterizar el proceso de degradación empleando los parámetros: índice de degradación y tiempo de vida útil remanente.

Determinar mediante una gráfica típica de degradación de componentes, como varia la salud de la turbomáquina empleando el *índice de degradación* y *tiempo de vida útil remanente*.

CAPÍTULO V

RESULTADOS Y DISCUSIÓN

A continuación se presentan los resultados obtenidos de la metodología planteada.

V.1 Investigar sobre el mecanismo y el modelado del proceso de degradación del desempeño de un turbofan enfocado en aplicaciones de pronóstico.

En la última década, la ciencia de los pronósticos ha madurado bastante y el entendimiento general del problema en cuanto a la predicción de la salud de un activo ha mejorado enormemente. Tanto los métodos basados en datos como los basados en la física del equipo han demostrado tener ventajas únicas que son específicas de los contextos de aplicación (Ramasso y Saxena, 2014). Sin embargo, hasta hace muy poco, un cuello de botella común en el desarrollo de métodos basados en datos era la falta de disponibilidad de datos que abarcaran desde la ejecución del equipo hasta la falla del mismo. En la mayoría de los casos del mundo real, los datos contienen condiciones de falla crecientes en varios niveles de severidad, pero poca es la información en los datos que captura la evolución de las fallas hasta que esta ocurre. La obtención de datos reales que permita monitorear la progresión de fallas del equipo suele requerir mucho tiempo y es costoso (Ramasso y Saxena, 2014). La mayoría de las veces, los sistemas de campo no están adecuadamente instrumentados para la recopilación de datos relevantes o no pueden ser distribuidos debido a restricciones de propiedad.

Para la mayoría de los sistemas complejos como los motores de aviación, encontrar un modelo adecuado que permita monitorear cambios relacionados con la salud del equipo es ciertamente un desafío en sí mismo. Además, el efecto de cómo se debe modelar la propagación del daño dentro de un modelo también es un situación que debe abordarse. Los problemas giran en torno a cómo esta propagación se manifiesta en los datos de los sensores (ver Figura N° 21), de modo que los usuarios puedan construir soluciones de pronóstico significativas.

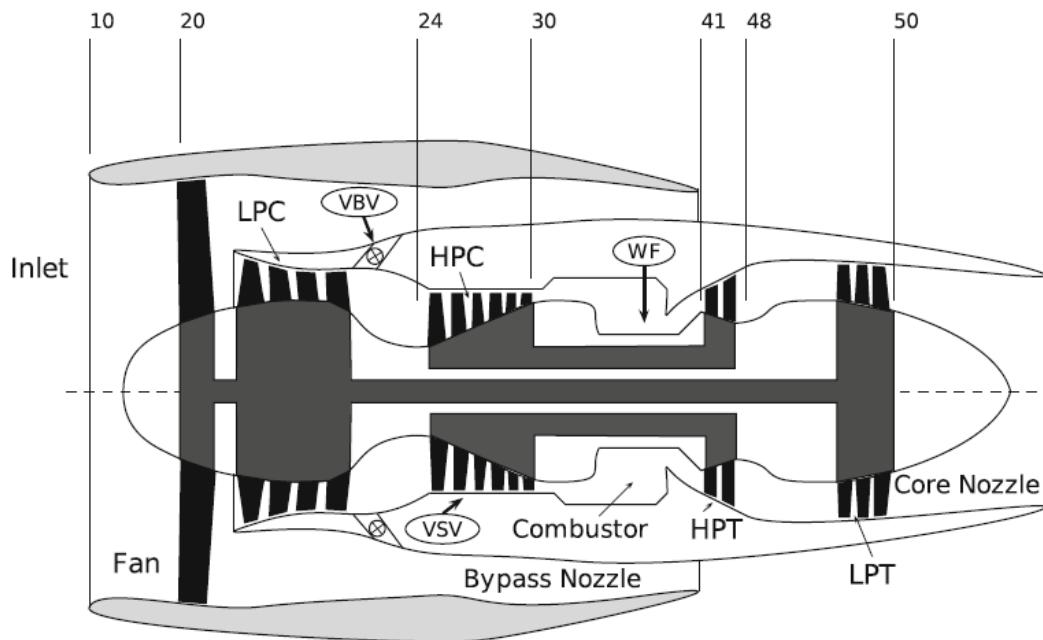


Figura N° 21. Localización de los sensores típicos en un turbofan (Richter, 2011).

La ausencia de datos comunes, que los investigadores pueden usar para comparar sus enfoques, ha sido un impedimento para el progreso en el campo de los pronósticos. En un intento de aliviar este problema, el *Centro de Investigación Ames de la NASA* (ARC, por sus siglas en inglés) ha puesto a disposición de la comunidad interesada en el Pronóstico y Diagnóstico de la Salud de activos, un conjunto de datos, constituidos por 4 archivos que van desde la ejecución hasta la falla de un motor aeronáutico turbofan, para permitir la evaluación y comparación de algoritmos de diagnósticos y pronósticos.

Los conjuntos de datos que provee la NASA fueron desarrollados empleando el *Simulador Comercial de Sistemas de Aeropropulsión Modular* (C-MAPSS, por sus siglas en inglés), el cual está hecho bajo Matlab y Simulink (Saxena y Goebel, 2008) y plantean varios desafíos que han sido abordados por diferentes métodos en la literatura. En particular, los datos tienen alta variabilidad debido al ruido de los sensores, efectos de las condiciones de operación y la presencia de múltiples modos de falla (*para tener acceso a los datos, siga el siguiente enlace: <https://goo.gl/W9EPmo>*).

Este conjunto de datos se han usado en distintas investigaciones para realizar el estudio de diagnóstico y pronóstico como se muestra en la Figura N° 22.

Además, estás bases de datos han recibido más de 7000 descargas desde su publicación para llevar a cabo la tarea PHM. Una de las investigaciones que tiene mayor interés para este Trabajo Especial de Grado, es el realizado por Zheng et al. en el año 2017, en donde utilizando estás bases de datos, crearon un modelo de redes neuronales recurrentes con neuronas LSTM, para determinar el estado de salud de los turbofan.

Su modelo lo compararon con otras investigaciones y determinaron mejoras significativas. Algunas de estas mejoras superan el 10%, que para un algoritmo de deep learning representa un porcentaje de mejora importante.

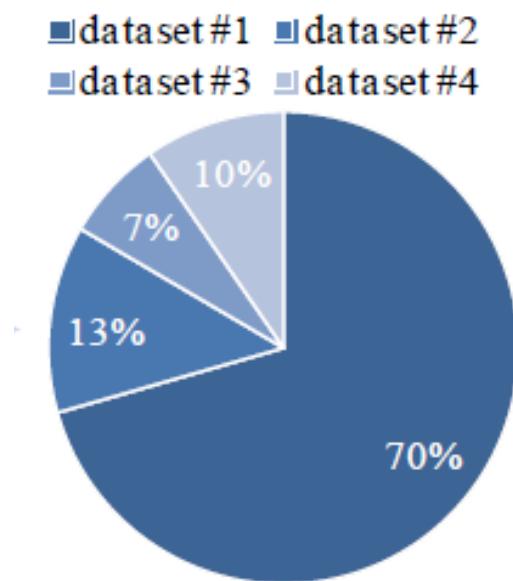


Figura N° 22. Representación del porcentaje de uso de la base de datos de la NASA en el estudio PHM.

Usando el entorno de simulación C-MAPSS, se generaron cuatro conjuntos de datos. Como lo describen Saxena y Goebel en el año 2008, los parámetros de inyección de fallas seleccionados se variaron para simular tendencias de degradación continua. Los datos de varias partes del turbofan se recopilaron para registrar los

efectos de las degradaciones en las mediciones de los sensores y proporcionar series de tiempo que muestren comportamientos de degradación en unidades múltiples. Estos conjuntos de datos poseen características únicas que los hacen muy útiles y adecuados para desarrollar algoritmos de pronóstico.

1. Los datos representan una respuesta multidimensional de un sistema complejo no lineal de una simulación de alta fidelidad que modela muy de cerca un sistema real.
2. Estas simulaciones incorporaron altos niveles de ruido introducidos en varias etapas para adaptarse a la naturaleza de la variabilidad que generalmente se encuentra.
3. Los efectos de las fallas se ocultan debido a las condiciones operativas, que es otro rasgo común de la mayoría de los sistemas operativos.
4. Se proporcionan datos de muchas unidades para permitir que los algoritmos extraigan tendencias y generen asociaciones para aprender el comportamiento del sistema útil para predecir el RUL.

Tabla N° 1. Conjunto de datos del simulador C-MAPSS para un turbofan.

Dataset	Máquinas train	Máquinas test	Cond. de op.	Modo falla
FD001	100	100	UNO	HPC
FD002	260	259	SEIS	HPC
FD003	100	100	UNO	FAN, HPC
FD004	249	248	SEIS	FAN, HPC

Son cuatro conjuntos de datos que consisten en múltiples series de tiempo de sensores. Cada conjunto de datos se divide en subgrupos de entrenamiento, prueba y el tiempo de vida útil remanente (RUL). Cada serie de tiempo corresponde a un motor diferente. Cada motor arranca con distintos grados de desgaste inicial y variación de fabricación que es desconocida para el usuario. Este desgaste y

variación inicial se consideran normales, es decir, no se considera una condición de falla. Hay tres configuraciones operativas que tienen un efecto sustancial en el rendimiento del turbofan. Estas configuraciones también se incluyen en los datos. Finalmente, los datos también están contaminados con el ruido del sensor.

- Datos de entrenamiento: son los datos de ejecución del turbofan desde el estado inicial hasta que ocurre la falla.
- Datos de prueba: Son los datos de operación del turbofan sin eventos de falla registrados.
- Datos RUL: contiene la información de los ciclos verdaderos restantes para cada motor en los datos de prueba.

Los datos de entrenamiento consisten en múltiples series de tiempo con "ciclos" como unidad de tiempo, junto con 21 lecturas de sensores para cada ciclo.

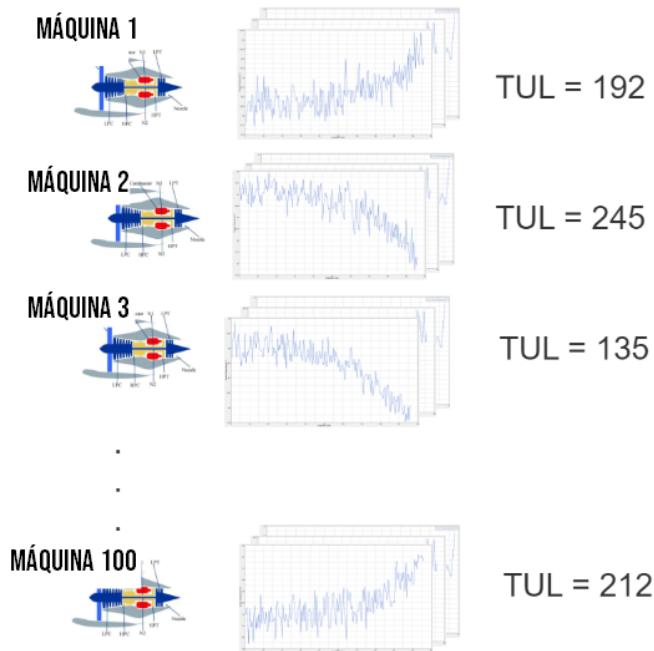


Figura N° 23. Estructura datos de entrenamiento.

En estos datos de entrenamiento, se conoce la vida total de las maquinas (TUL, por sus siglas en inglés). Además, se supone que el motor está funcionando nor-

malmente al comienzo de cada serie de tiempo. Luego, comienza a degradarse en algún momento durante la serie de los ciclos operativos. La degradación progresiva y crece en magnitud. Cuando se alcanza un umbral desconocido, entonces el motor se considera inseguro para operaciones posteriores. En otras palabras, se puede inferir que el último ciclo en cada serie de tiempo puede considerarse como el punto de falla del motor correspondiente.

Tomando como ejemplo los datos de entrenamiento del primer conjunto de datos, el turbofan con id = 1 falla en el ciclo 192, y el motor con id = 2 falla en el ciclo 245.

Los datos de prueba tienen el mismo esquema que los datos de entrenamiento, la única diferencia es que estos datos no indican cuándo ocurre la falla (en otras palabras, el último ciclo de tiempo no representa el punto de falla). Por lo tanto, los datos de prueba representan una serie de tiempo parcial con la que tenemos que predecir en función del último ciclo dado.

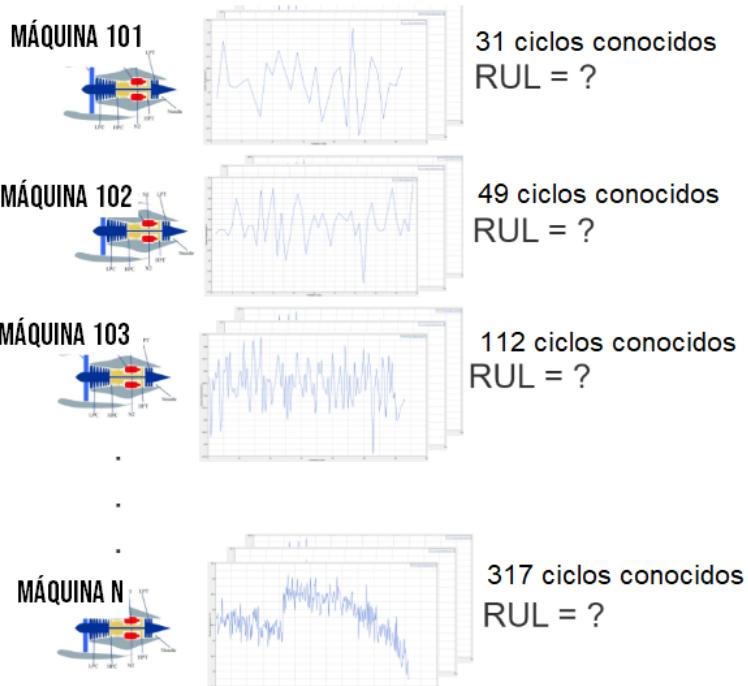


Figura N° 24. Estructura datos de prueba.

Los datos de vida útil remanente proporcionan el número de ciclos de trabajo

restantes para los motores en los datos de prueba.

Tabla N° 2. Estructura datos RUL (simplificado)

No. Máquina	Ciclo actual	RUL verdadero (ciclos)
101	31	112
102	49	98
103	112	69
104	106	82

La estructura general de los datos de entrenamiento y prueba es la que se muestra en la siguiente tabla:

Tabla N° 3. Estructura de todos los dataset

Atributos	Descripción	Unidades
unit Number	Número correspondiente a la máquina	–
cycles	Ciclo de operación de la máquina	–
cond. op. 1	Altitud	ft
cond. op 2	Número de Mach	–
cond. op 3	Ángulo de la palanca de aceleración (TRA)	%
sensor 1	Temperatura total a la entrada del fan (T20)	°R
sensor 2	Temperatura total a la salida del LPC (T24)	°R
sensor 3	Temperatura total a la salida del HPC (T30)	°R
sensor 4	Temperatura total a la salida de la LPT (T50)	°R
sensor 5	Presión a la entrada del ventilador (P20)	psia
sensor 6	Presión total en el ducto de Bypass (P15)	psia
sensor 7	Presión total a la salida del HPC (P30)	psia
sensor 8	Velocidad física del ventilador (Nf)	rpm
sensor 9	Velocidad física del núcleo (Nc)	rpm
sensor 10	Relación de presión del motor (P50/P20) (epr)	–

Atributos	Descripción	Unidades
sensor 11	Presión estática en el HPC (Ps30)	psia
sensor 12	Relación de flujo de combustible a Ps30 (phi)	pps/psi
sensor 13	Velocidad del ventilador corregida (NRf)	rpm
sensor 14	Velocidad del núcleo corrediga (NRc)	rpm
sensor 15	Relación de Bypass (BPR)	–
sensor 16	Relación combustible-aire del quemador (farB)	–
sensor 17	Entalpia de purga (htBleed)	–
sensor 18	Velocidad del ventilador (Nf_dmd)	rpm
sensor 19	Velocidad corregida del ventilador (PCNfR)	rpm
sensor 20	Purga de refrigerante de HPT (W31)	lbm/s
sensor 21	Purga de refrigerante de LPT (W32)	lbm/s

Los 21 sensores definen la salud de nuestra unidad en un tiempo de ciclo dado. Algunos sensores tienen signos más evidentes de degradación.

Un ciclo se refiere a un recorrido completo de la aeronave.

Finalmente, las condiciones de operación se utilizan para describir la condición operativa de un punto de datos determinado para una unidad bajo las cuales operan los distintos turbofan.

Tabla N° 4. Estructura datos RUL

Cond. Op.	Altitud (ft)	Mach	TRA (%)
UNO (nivel del mar)	aprox. 0	aprox. 0	100
SEIS	(0-42K)	(0-0.84)	100

Por último, en este Trabajo Especial de Grado se trabajó solamente con la base de datos 1 del repositorio de la NASA.

V.2 Desarrollar un modelo para detección de falla para un turbofan a partir de datos típicos de operación.

El enfoque para el desarrollo de este objetivo consistió en 7 etapas:

- (1) Adaptación de los datos para mayor facilidad en su manejo a lo largo del análisis.
- (2) Análisis exploratorio de los datos empleando técnicas estadísticas.
- (3) Eliminación de valores atípicos.
- (4) Agrupamiento de los datos.
- (5) Normalización de los datos.
- (6) Selección de las características más importantes (KPI).
- (7) Implementación del modelo de clasificación para la detección de fallas.

Con la detección de fallas se buscó obtener el punto en donde los datos comienzan a mostrar un patrón de degradación significativo (ver Figura N° 25)

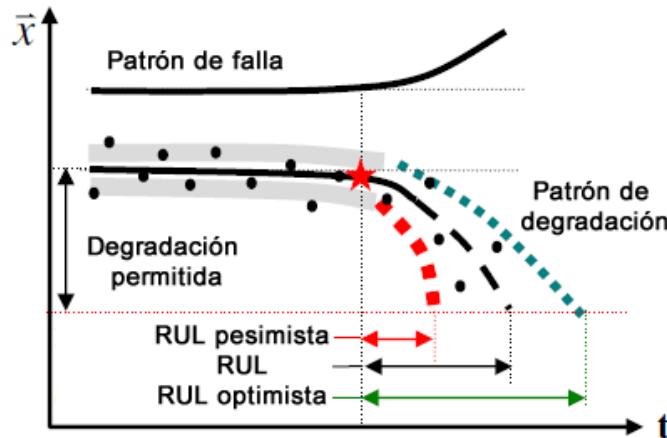


Figura N° 25. Detección de fallas. Se indica mediante la estrella roja (Li y Nilkitsaranont, 2009).

V.2.1 Adaptación de los datos

Aunque es posible comprender nuestro conjunto de datos, sería inapropiado simplemente insertar nuestros datos en cualquier modelo sin procesarlos. Como es típico en el análisis de datos, fue necesario preprocessar los datos antes de llevarlos hacia un modelo de aprendizaje automático para obtener columnas específicas, como sensores determinados, configuraciones de operación o número de ciclo, que permitieran aumentar la probabilidad de obtener buenos resultados con la menor cantidad de variables.

V.2.1.1 Datos faltantes

En esta etapa, empleando Python, se cargaron los cuatro dataset de entrenamiento para observar los tipos de datos almacenados en cada columna (característica) y corroborar la ausencia de datos faltantes (NaN). Una vez hecho la búsqueda se determinó que todos los datos en cada columna eran valores numéricos. De igual forma, no se encontraron valores faltantes (NaN) en ninguna columna, lo que permitió continuar con el tratamiento de los mismos.

Tabla N° 5. Datos de entrenamiento.

Id	Cycle	Opset1	Opset2	Opset3	Sensor1	...	Sensor21
1	1	-0.0007	-0.0004	100.0	518.67	...	23.4190
1	2	0.0019	-0.0003	100.0	518.67	...	23.4236
1	3	-0.0043	0.0003	100.0	518.67	...	23.3442
...

V.2.1.2 Condiciones de operación

Como se puede observar en la Tabla N° 5 de una manera simplificada, la misma máquina operó bajo diferentes configuraciones de operación, lo que sería un problema a la hora de poder comparar las máquinas entre si y/o los resultados. Sin embargo, existían configuraciones que se repetían a lo largo de los datos. Por lo tanto, se decidió determinar las configuraciones de operación únicas, con la

finalidad de poder observar, en la data original, a que configuración de operación corresponden los datos.

Se determinó que solo 12289 configuraciones eran únicas y el resto eran réplicas de estas condiciones.

Tabla N° 6. Condiciones de operación únicas.

#	Opset1	Opset2	Opset3
1	-0.0007	-0.0004	100.0
2	0.0019	-0.0003	100.0
3	-0.0043	0.0003	100.0
...

Una vez se contaba con las condiciones de operación únicas se procedió a buscar estas condiciones a lo largo de la base de datos para asegurar bajo que configuración de operación estaban operando las diferentes máquinas. El resultado se puede observar en la tabla Tabla N° 7.

Tabla N° 7. Datos con la nueva columna de las configuraciones de operación.

Id	Cycle	Opset_id	Sensor1	Sensor2	...	Sensor21
1	1	0	518.67	641.82	...	23.4190
1	2	1	518.67	642.15	...	23.4236
1	3	2	518.67	642.35	...	23.3442
...

V.2.2 Análisis exploratorio de los datos

A partir de este punto se realizó todo el análisis de los datos mediante técnicas estadísticas y de visualización.

V.2.2.1 Visualización de los datos

En esta etapa se graficaron todos los sensores para diferentes máquinas para tener una representación del comportamiento de los mismos.

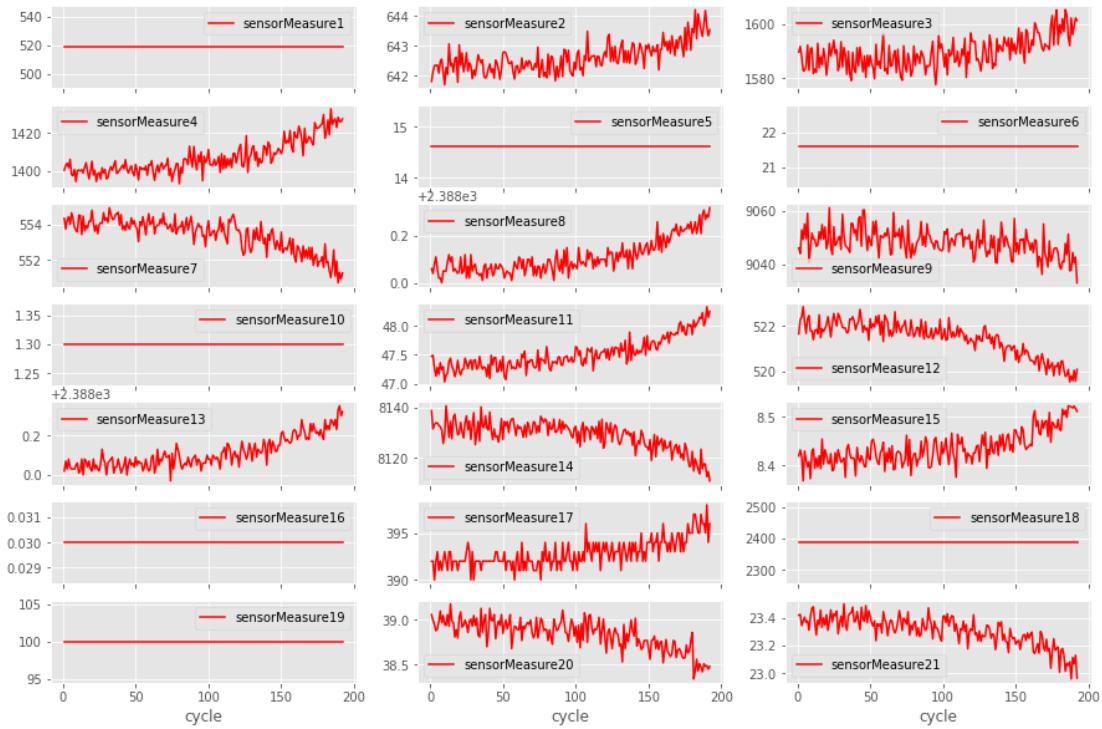


Figura N° 26. Sensores para la máquina 1.

En la Figura N° 26 se muestra el comportamiento de los sensores. Un punto que es importante mencionar es la presencia de sensores con valores constantes durante toda la vida de la máquina, esto es importante debido a que los sensores constantes deben retirarse de los datos para agilizar el aprendizaje de los algoritmos.

Para corroborar lo anterior, se determinó la varianza de todos los sensores a lo largo de la base de datos. Sensores cuya varianza sea cero, serán los sensores constantes que se retiraran en la selección de características.

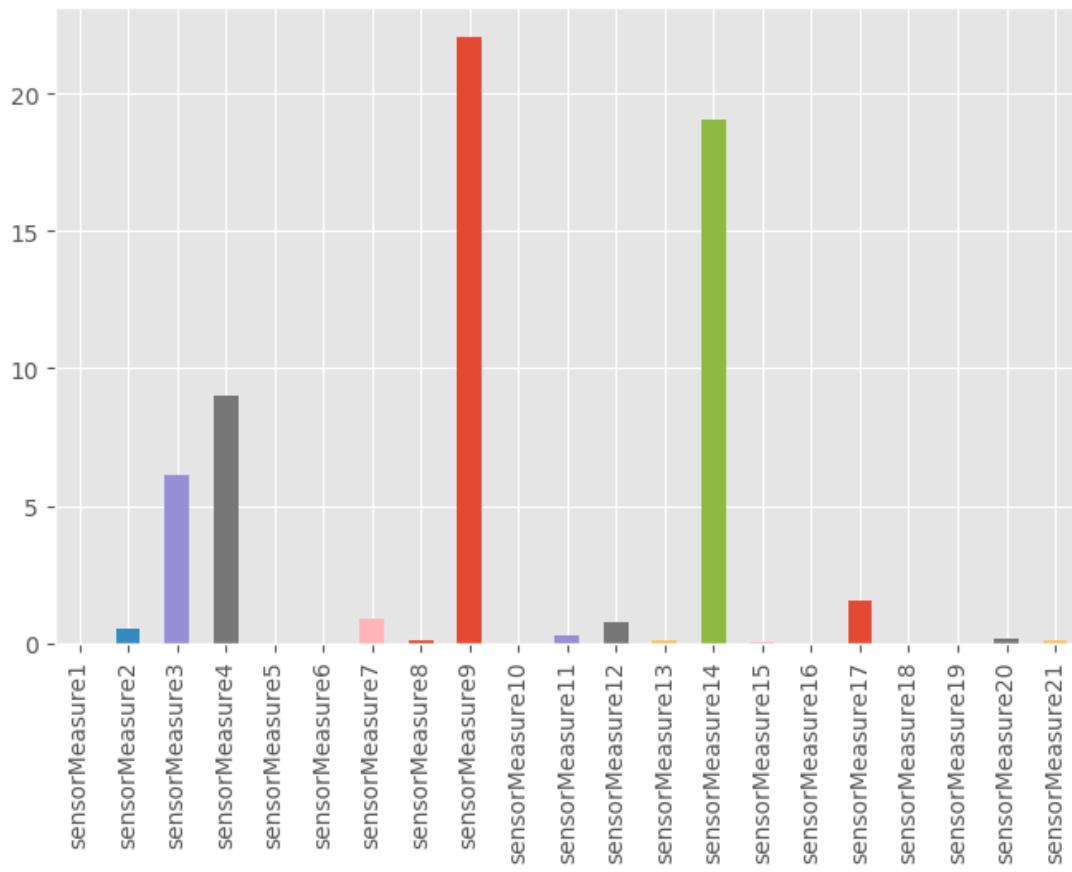


Figura N° 27. Varianza de todos los sensores.

Se observa que efectivamente hay sensores que se mantienen constantes en toda la base de datos. Sean estos: Sensor 1, Sensor 5, Sensor 6, Sensor 10, Sensor 16, Sensor 18 y Sensor 19.

V.2.2.2 Distribuciones

Es necesario observar qué tipo de distribución siguen los datos para determinar que tipo de técnicas son las más apropiadas aplicar, siendo estas técnicas paramétricas o no paramétricas. Si los datos siguen una distribución normal, se aplicaran técnicas paramétricas, en el caso contrario se aplicaran técnicas no paramétricas. Por esta razón, se graficó la función de densidad de los sensores.

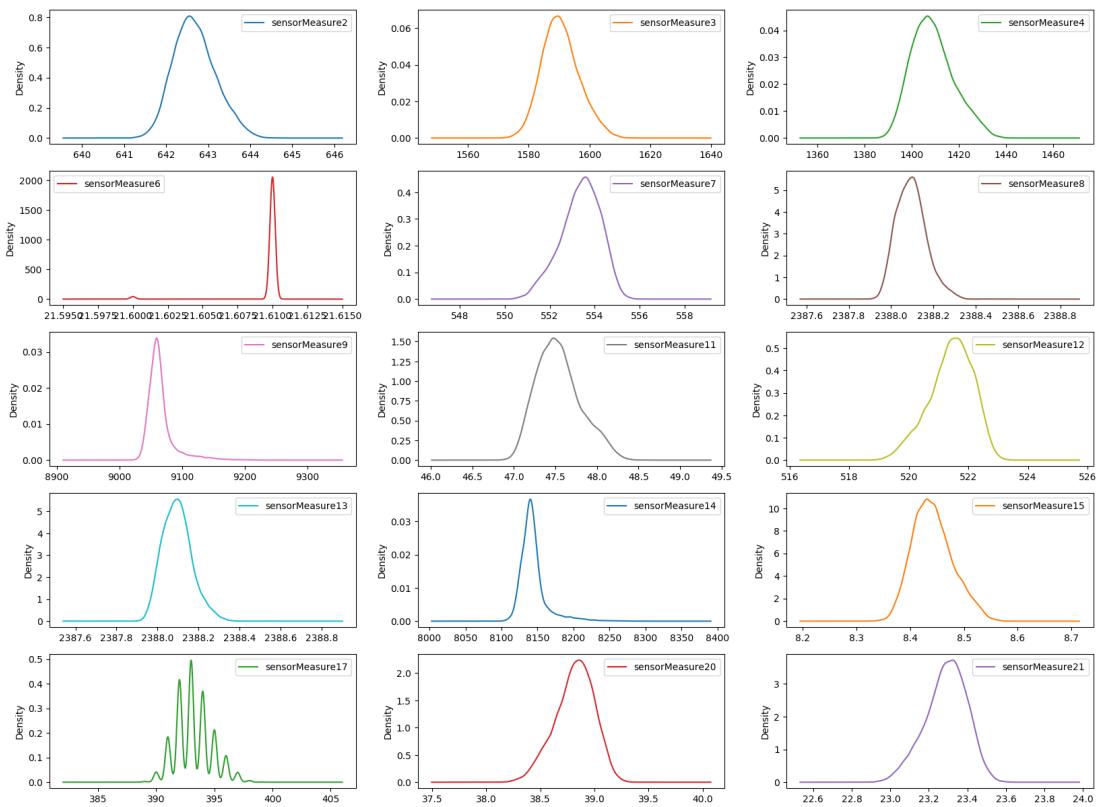


Figura N° 28. Distribución de los sensores.

En la figura anterior se observa como los datos de la mayoría de los sensores parecieran ajustarse a una distribución normal típica, sin embargo, no queda claro si se puede asumir que todos sigan distribución normal. A pesar de que se aplicaron transformaciones a los datos para mejorar su distribución, como por ejemplo, transformación logarítmica o raíz cuadrada, esto no tuvo un cambio significativo en la distribución de los mismos.

Para acabar con dicha incertidumbre se realizó una prueba de normalidad, que en términos generales consiste en establecer una hipótesis nula, una hipótesis alternativa y el nivel de significancia (α), para luego determinar una probabilidad, y en función de dicha probabilidad verificar si para el nivel de significancia establecido se acepta o no alguna de las hipótesis.

La prueba de normalidad que se realizó fue la prueba D'Agostino (Echeverri et al., 2015). Para ello se establecieron las siguientes hipótesis y nivel de signifi-

cancia:

- *Hipótesis nula (H_0)*: La muestra proviene de una población con distribución normal.
- *Hipótesis alternativa (H_1)*: La muestra no proviene de una población con distribución normal.
- $\alpha = 0.05$



Figura N° 29. Prueba de normalidad estadística (La imagen es similar para las pruebas de dos colas, sólo que aparecerían áreas amarillas y verdes en ambas colas).

Al aplicar dicha prueba en los datos se corroboró que ningún sensor proviene de una distribución normal. Por lo tanto, se determinó que se deben aplicar técnicas no paramétricas.

Para finalizar, de manera adicional se realizó una matriz de distribuciones de los datos, en ella se incluye la función objetivo RUL para observar cómo se comportan los datos respecto a este parámetro.

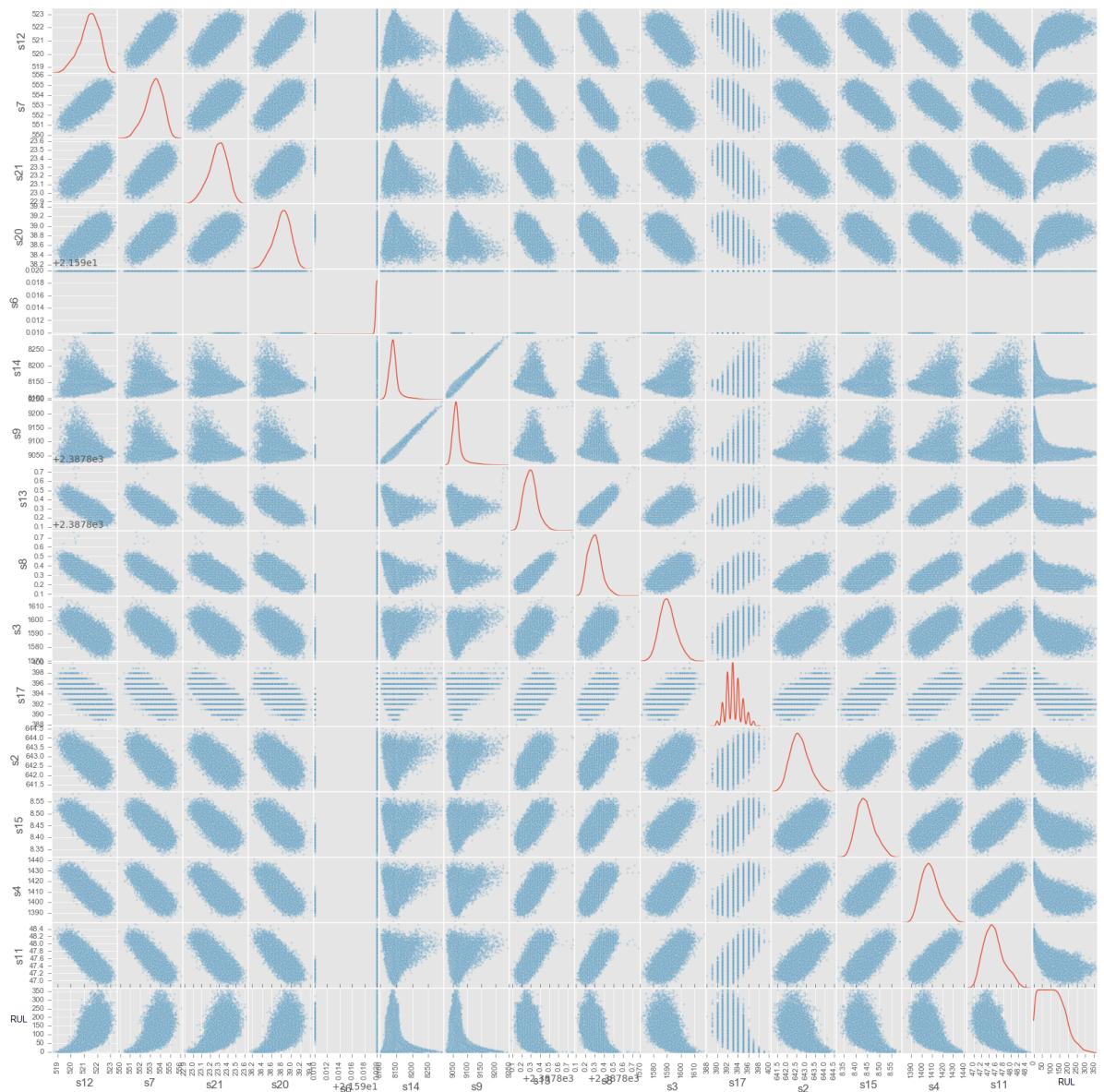


Figura N° 30. Matriz de distribuciones (la mayoría de las características tienen una relación no lineal con respecto al RUL).

V.2.2.3 Matriz de correlación

A partir de este punto, se buscó tener una mayor información en cuanto a la correlación entre los sensores. Para ello se empleó el coeficiente de correlación de Spearman, con la finalidad de poder determinar los sensores a los que habría que

prestarle mayor atención en la sección de Selección de Características.

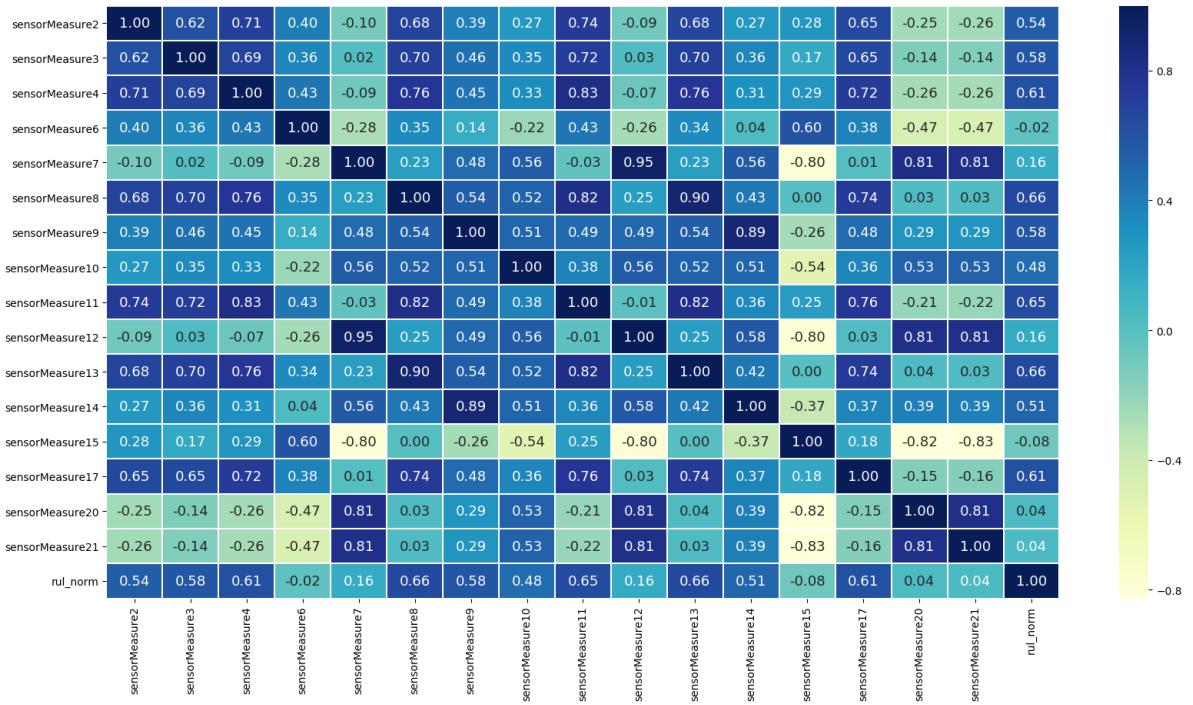


Figura N° 31. Matriz de correlación del archivo 1

La información útil que se puede extraer de esta matriz es la presencia de sensores que están altamente correlacionados entre sí, es decir, tienen una correlación mayor igual a 0.8 (Bruce y Bruce, 2017).

- (sensor 14, sensor 9) - Correlación: 0.90
- (sensor 11, sensor 4) - Correlación: 0.82
- (sensor 11, sensor 7) - Correlación: -0.82
- (sensor 11, sensor 12) - Correlación: -0.84
- (sensor 4, sensor 12) - Correlación: -0.81
- (sensor 7, sensor 12) - Correlación: 0.81
- (sensor 8, sensor 13) - Correlación: 0.83

Observando estos sensores se determinó que su correlación puede afectar el rendimiento de algunos algoritmos de aprendizaje automático. Por lo tanto, parte de estas características se deberán eliminar en la selección de características durante la fase de modelado.

V.2.3 Eliminación de valores atípicos

Hasta ahora solo se han implementado técnicas estadísticas para analizar los datos. Es a partir de este punto donde se comienzan a emplear algoritmos de aprendizaje automático para la limpieza de los datos. Para observar la presencia de valores atípicos en nuestros datos se aplicó el algoritmo de aprendizaje no supervisado Local Outlier Factor (LOF) que se propone en la investigación realizada por Zhao et al. (2015)

Primero se graficaron los datos de un sensor aleatorio para una máquina cualquiera como se muestra en la siguiente figura.

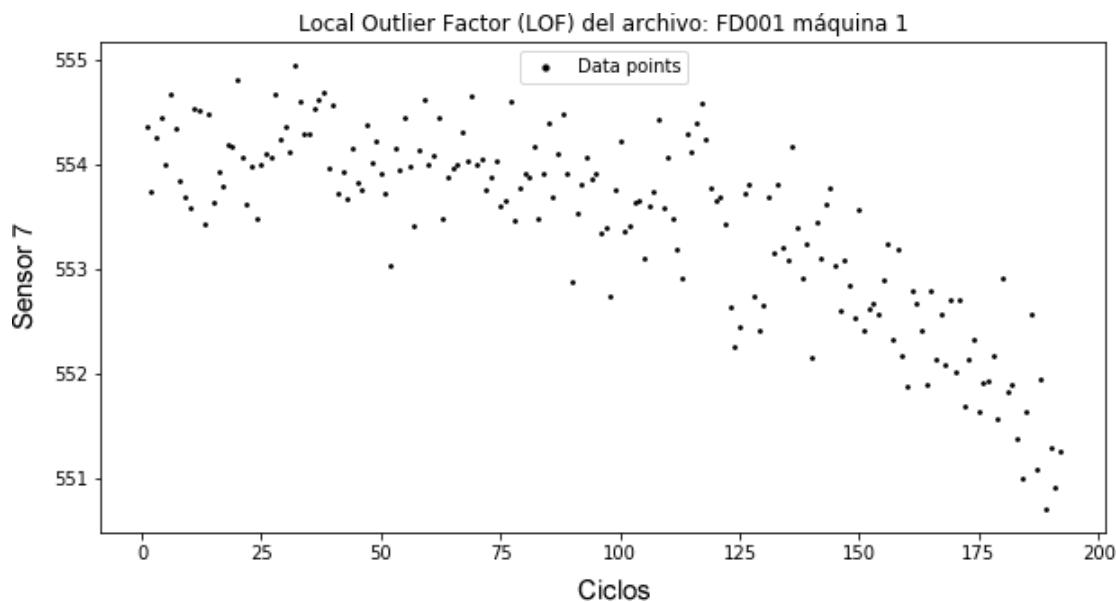


Figura N° 32. Serie de tiempo del sensor 7 de la máquina 1.

Luego, se aplicó el algoritmo LOF al mismo sensor para determinar cuáles posibles puntos podían ser valores atípicos.

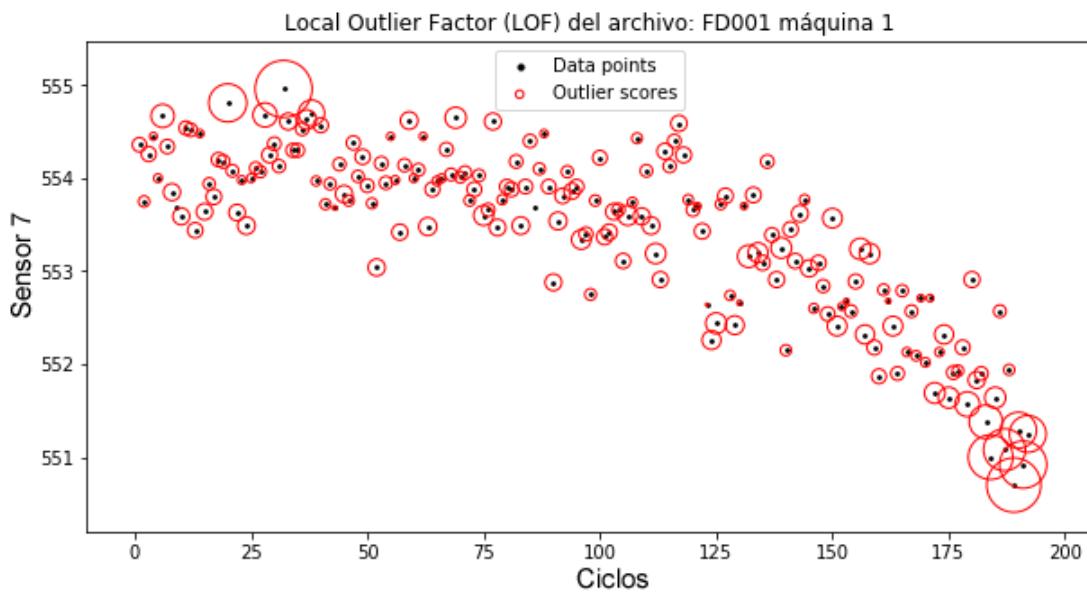


Figura N° 33. Serie de tiempo del sensor 7 con el score del LOF

Como se explicó en el marco teórico, el algoritmo LOF evalúa la densidad de los puntos para un número de vecinos establecidos y luego establece el puntaje correspondiente. Los puntajes cercanos a -1 serán considerados datos "normales" (círculos rojos cercanos a los valores). Los datos con puntajes lejos de -1 serán considerados atípicos (círculos rojos lejos de los valores). En la investigación de Zhao et al. (2015) establecen como puntaje máximo -1.3 para ser considerado valor normal. Este valor es relativo a los datos, sin embargo cuando se probó el algoritmo con otros puntajes, tanto por encima como por debajo de -1.3 se evidenció que el algoritmo dejaba por fuera valores que si tenían el comportamiento de valor atípico, o agregaba valores que no debían considerarse.

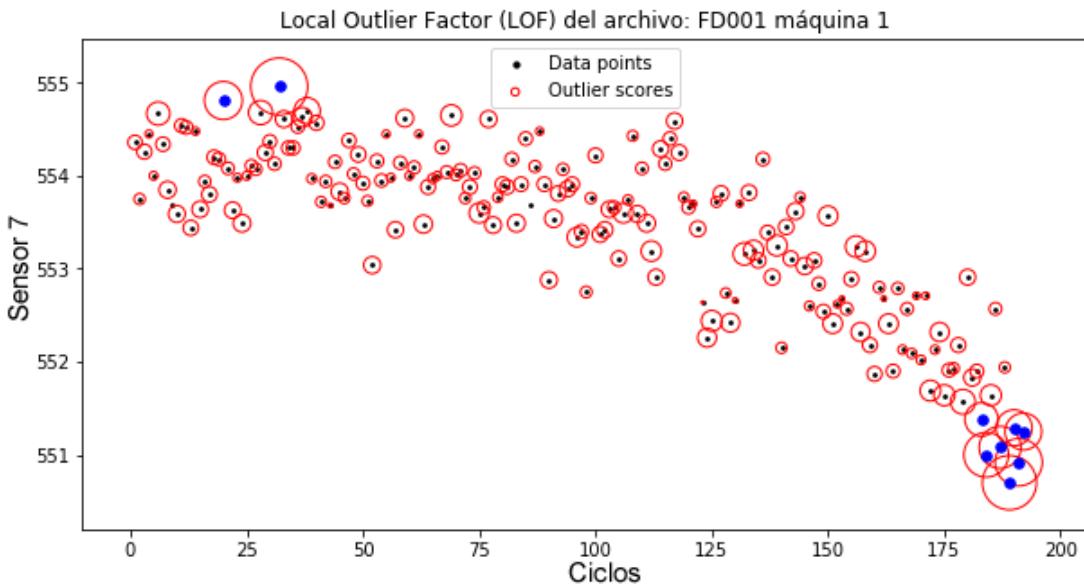


Figura N° 34. Serie de tiempo del sensor 7 señalando los valores atípico.

Al aplicar el mismo procedimiento con otro sensor.

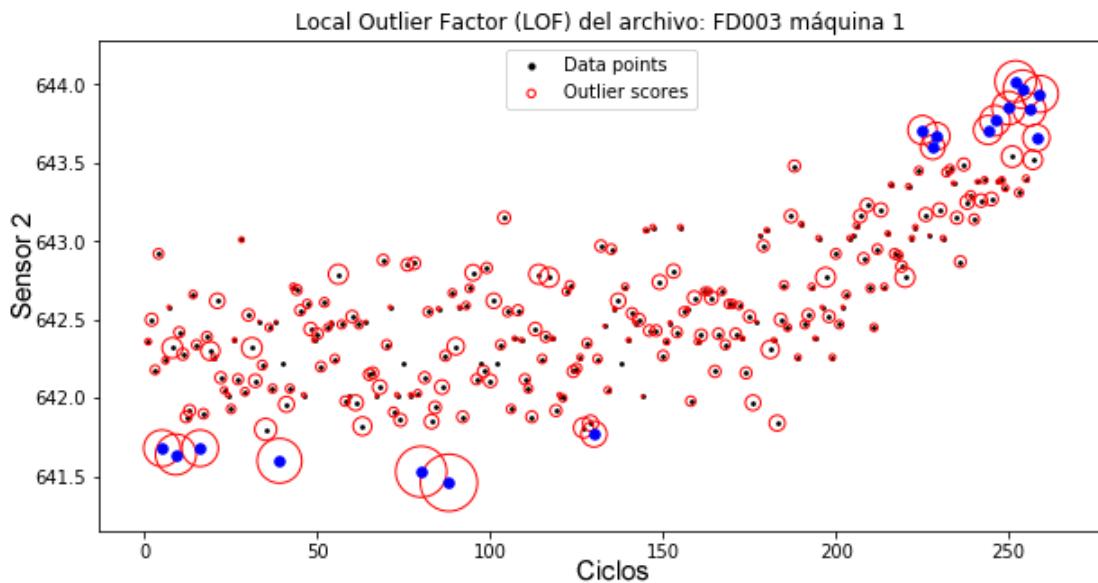


Figura N° 35. Serie de tiempo del sensor 2 señalando los valores atípico

Llegados a este punto, se observó un patrón interesante en los datos, y es que

el algoritmo es capaz de detectar los valores atípicos y estos en su mayoría se encuentran al final de la serie de tiempo. Al confirmar este patrón en el resto de los sensores fue necesario analizar la física de los datos y lo que estaba ocurriendo. El conglomerado de puntos atípicos al final de los datos se interpretó como la representación de la degradación del equipo desde el inicio de la falla hasta el final de la misma. Por lo tanto, estos valores "atípicos" que se encuentran al final de la serie de tiempo, no se retiraron de los datos.

Por lo justificado anteriormente, se procedió a retirar el resto de los valores atípicos empleando la técnica del promedio entre un valor antes y uno después del valor atípico correspondiente, dando como resultado lo que se muestra en la siguiente figura.

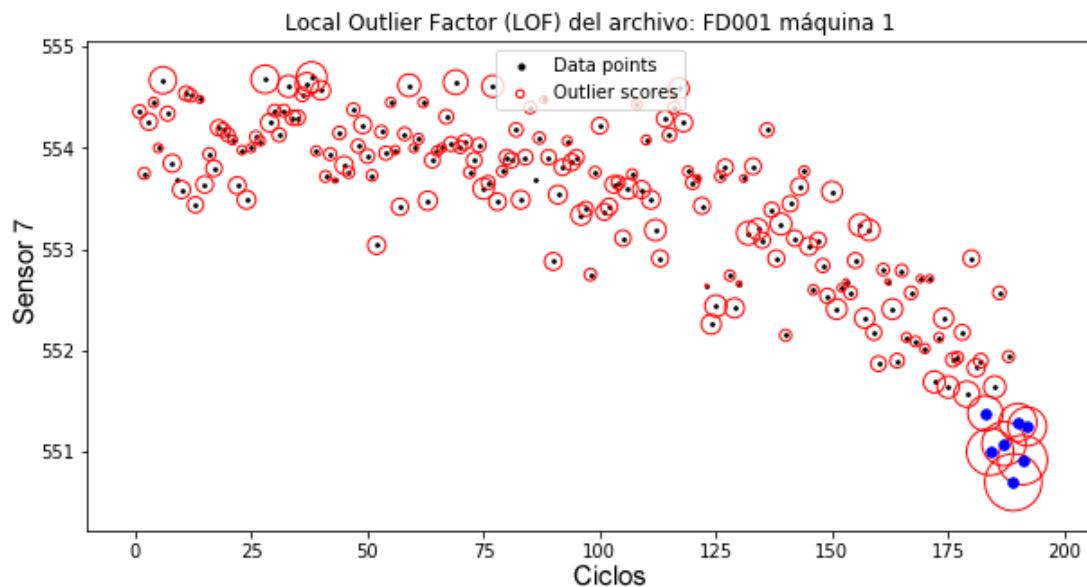


Figura N° 36. Serie de tiempo del sensor 7 sin valores atípicos

Como curiosidad, fue que la gran mayoría de los datos que representan la degradación se encontraban a 30 ciclos antes del último ciclo reportado en los datos de entrenamiento.

V.2.4 Agrupamiento de los datos

Si bien con el algoritmo LOF se logró identificar una manera de agrupar los datos entre valores normales y valores degradados, la finalidad de dicho algoritmo no va enfocada directamente al agrupamiento de datos. Por lo tanto, en esta parte se implementó el algoritmo no supervisado también basado en la densidad de puntos DBSCAN.

En este caso, y como se mencionó en la parte de teoría, los parámetros necesarios para hacer uso del DBSCAN son la distancia entre puntos (ϵ) y la cantidad mínima de puntos por cluster (minsamples). Haciendo uso de la referencia (Thang y Kim, 2011) se utilizaron los parámetros que ellos proponen de manera optimizada, es decir, $\epsilon = 2.0$ y minsamples = 1000.

Tomando como ejemplo la máquina 1 que cuenta con 192 registros para cada sensor, el algoritmo fue capaz de agrupar estos datos en 2 clusters. El primero comenzaba en el inicio de los datos y terminaba en el registro 158, mientras que el siguiente cluster comenzaba en el registro 159 y terminaba en el último dato registrado, es decir, en el 192 (ver figura N° 37).

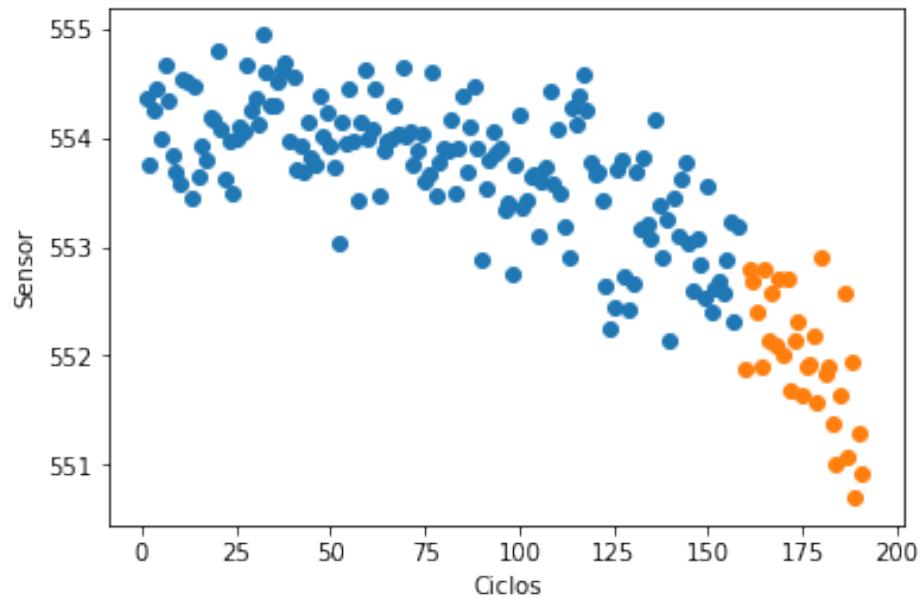


Figura N° 37. Agrupamiento del DBSCAN.

Además, recordando lo discutido en la sección anterior del algoritmo LOF, que se observó un patrón de datos degradados con un rango de 30 ciclos antes del último ciclo. Con el algoritmo DBSCAN fue posible corroborar lo observado anteriormente ya que si vemos el rango del segundo cluster ($192 - 159 = 33$ ciclos) coincide casi perfectamente con lo que nos arrojó el algoritmo LOF. Además, cuando se aplicó el DBSCAN al resto de las máquinas, la agrupación de datos fue similar con pequeñas variaciones de rango.

Fue entonces cuando se creó una columna nueva en la que se almacenaban las etiquetas que caracterizaban a cada dato. Es decir, los valores considerados como normales por el algoritmo DBSCAN tendrán una etiqueta de cero (clase negativa) y los considerados como anómalos tendrán la etiqueta de uno (clase positiva). La nueva columna será la variable categórica (función objetivo) en el modelo de clasificación supervisada.

V.2.5 Normalización de los datos

Existen muchas maneras diferentes de normalizar los datos. Sin embargo, en el trabajo de investigación de Zheng et al. en el año 2017, él emplea la misma base de datos y utiliza la normalización estándar, mejor conocida como normalización z-score, para poder comparar sus resultados con la de 5 investigaciones más. Con base en lo anterior, en este trabajo especial de grado también se aplicó la misma normalización con fines comparativos.

La normalización se le aplicó tanto a los datos de entrenamiento como a los datos de prueba.

V.2.6 Selección de características

Un gran conjunto de características a menudo contiene información redundante e irrelevante, y como consecuencia de esto se puede degradar el rendimiento del clasificador. Por esta razón, la reducción de dimensiones mediante la selección de características, es uno de los pasos fundamentales en el procesamiento de datos. La selección de características consiste en la elección de un subconjunto de atributos que logre el error más bajo de acuerdo con ciertos límites permitidos (Zheng et al., 2017).

En primer lugar, se eliminaron los sensores cuya serie de tiempo era constante a lo largo de los datos como se mostró en la figura N° 26.

En segundo lugar, para lograr la selección de características, se empleó el algoritmo de Eliminación de Características Recursivas (RFE) sobre el método de regresión de Árboles de Decisión. En este caso, el algoritmo RFE se ajusta al modelo de Árboles de Decisión y elimina la característica (o características) más débiles hasta que se alcanza el número mínimo de atributos que arrojan las mejores métricas de ajuste. En este caso se usó el Error Cuadrático Medio como métrica de selección.

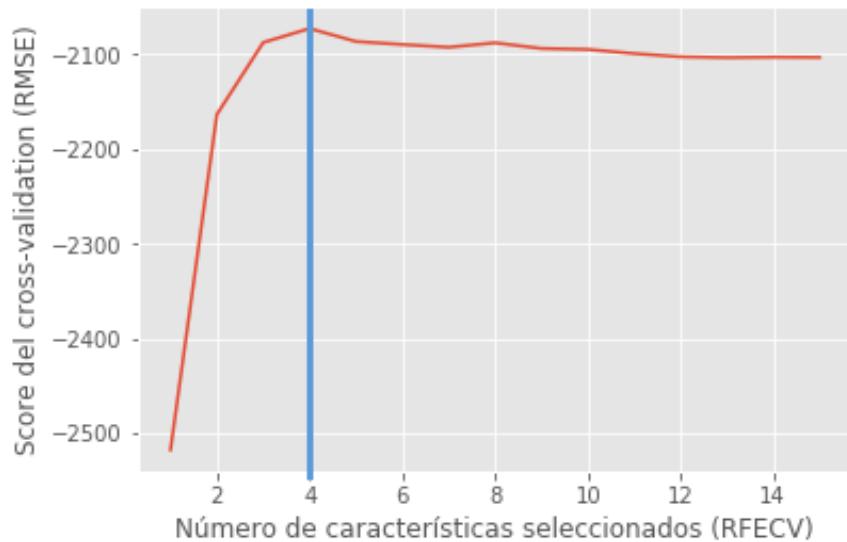


Figura N° 38. Gráfica de la selección de características para los cuatro archivos.

En la Figura N° 38 se muestra la cantidad de atributos seleccionados por el algoritmo RFE para nuestra base de datos.

- Sensor 4: Temperatura total a la salida de la LPT (T50).
- Sensor 9: Velocidad física del núcleo (Nc)
- Sensor 11: Presión estática en el HPC (Ps30)
- Sensor 12: Relación de flujo de combustible a Ps30 (phi)

De 21 sensores en la base de datos, se logró reducir la dimensión en 4 atributos. Estos sensores serán los denominados *indicadores clave de rendimiento (KPI)*.

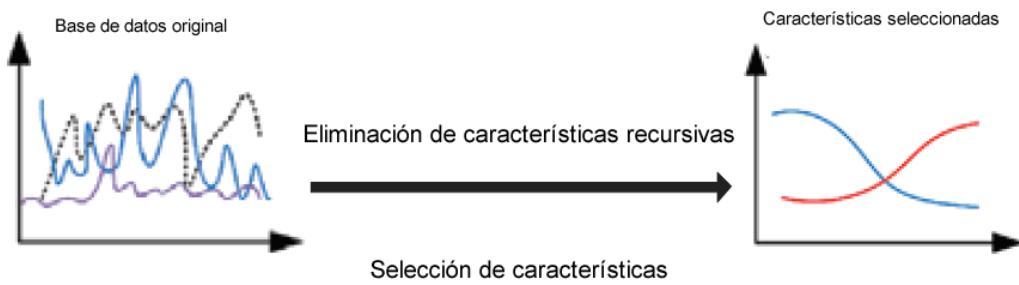


Figura N° 39. Con la selección de características se obtienen atributos que representan de manera más clara y con una alta precisión a los datos originales.

Adicionalmente, se muestran las importancias que tienen estos atributos.

Tabla N° 8. Importancia de los sensores seleccionados

Base de datos	Importancia
Sensor 11	0.69
Sensor 9	0.17
Sensor 4	0.1
Sensor 12	0.04
Opset_id	0.0006

Esta tabla se refiere a la importancia que le asigna el algoritmo RFE a cada característica luego de haber realizado la reducción de dimensiones. Se puede observar como el sensor 11 (*Presión estática en el HPC*) es el más importante en los datos finales. Esto tiene sentido ya que la degradación que ocurre en cada máquina de esta base de datos ocurre en el compresor de alta presión.

V.2.7 Modelo de clasificación para la detección de fallas

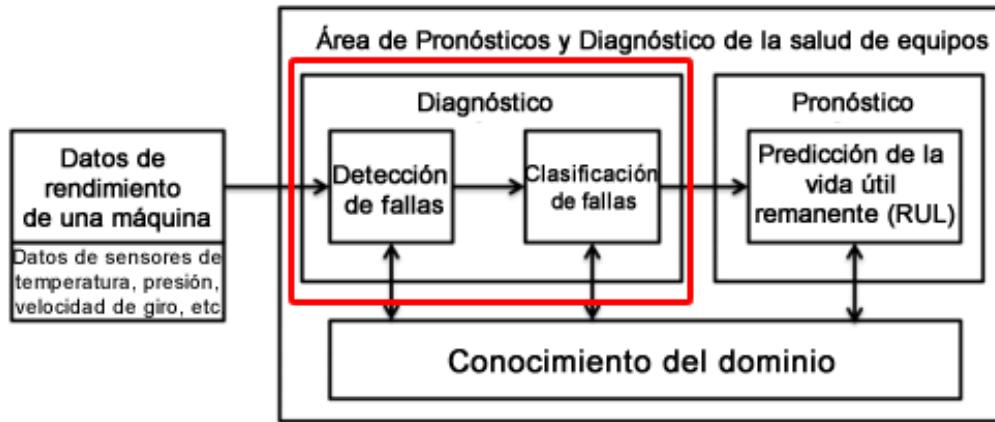


Figura N° 40. Ilustración del recorrido en el área de Pronósticos y Diagnósticos de la salud - Sección Diagnóstico

En esta última etapa, se construyó el modelo de clasificación capaz de poder determinar si un punto de la base de datos de prueba es un valor normal o un valor degradado. Antes de entrar en el detalle de cómo se construyeron y se evaluaron los algoritmos de clasificación, es pertinente recordar la estructura de los datos de prueba.

Los datos de prueba contienen la misma estructura de los datos de entrenamiento, sin embargo, no indican el ciclo en el que falla la máquina, es decir, las máquinas en los datos de prueba están registradas hasta un ciclo que es aún operativo. Por ejemplo, la máquina 1 sólo se tiene registrada hasta el ciclo 31. La cantidad de ciclos restantes de las máquinas en la base de datos de prueba se encuentran en los archivos RUL. Estos ciclos restantes se usarán para determinar cuál será la etiqueta real en los datos de prueba.

Sabiendo lo anterior, y recordando que con el algoritmo LOF se observó que los puntos degradados mantenían un rango de 30 ciclos antes del último ciclo, se procedió a etiquetar los datos de prueba para tener su etiqueta real.

Tabla N° 9. Estructura datos de prueba asociados al archivo 1

Id	cycle	Opset	Sensor4	Sensor9	Sensor11	Sensor12	RUL	Label
1	31	372	-0.871	-0.175	-0.950	-0.0752	112	0
2	49	50	0.912	-1.193	1.296	-0.014	98	0
3	126	831	2.120	-0.800	2.370	-1.640	69	0
—	—	—	—	—	—	—	—	—
18	133	104	2.161	-0.204	1.449	-1.230	28	1
—	—	—	—	—	—	—	—	—

Estás etiquetas se obtuvieron de la siguiente manera:

$$RUL > 30 \rightarrow Label = 0$$

$$RUL \leq 30 \rightarrow Label = 1$$

Con lo anterior mencionado, se emplearon distintos algoritmos de clasificación supervisada y se compararon entre ellos para determinar cuál arrojaba los mejores resultados.

Los algoritmos utilizados fueron los siguientes:

- Regresión logística. (Rosario, 2004)
- Árboles de decisión. (Reif et al., 2008)
- Vectores soportados lineales. (Chen et al., 2011)
- Bosques aleatorios. (Wang et al., 2018)
- K-vecinos.

Como se puede ver, los primeros 4 algoritmos se decidieron utilizar por tener referencias que soportaban su uso en problemas de detección de anomalías. Sin embargo, el último se propuso en esta investigación para compararlo con el resto.

De igual manera, se emplearon las métricas típicas de clasificación:

- Exactitud

- Precisión
- Recuperación
- AUC-ROC

No obstante, a las métricas que se les prestaron mayor atención para escoger el mejor algoritmo de clasificación fueron la Precisión, Recuperación y AUC-ROC.

Antes de aplicar los algoritmos fue necesario optimizar los parámetros requeridos, por lo tanto, se aplicó un método de optimización de hiperparámetros conocido como GridSearch.

A cada algoritmo se le asignó como función objetivo la columna *Label* y se entrenó con la configuración de operación y los sensores correspondientes para la base de datos.

Una vez que ya estuviera entrenado, el algoritmo se probó con la base de datos de prueba para determinar que etiqueta predecía los algoritmos de clasificación binaria.

V.2.7.1 Regresión logística

Tabla N° 10. Resultados de la Regresión Logística

Máquinas	1	2	3	4	–	91	92	93	94	95	96	97	98	99	100
RUL	112	98	69	82	–	38	20	85	55	128	137	82	59	117	20
Reales	0	0	0	0	–	0	1	0	0	0	0	0	0	0	1
Predichos	0	0	1	0	–	1	1	1	1	0	0	0	0	0	1

Para darle un apoyo a la visualización de los resultados, se utilizó una *matriz de confusión* para determinar la cantidad de VP, VN, FP y FN obtenidos.

	Verdadero positivo (VP):	Falso positivo (FP):
60	15	
	Falso negativo (FN):	Verdadero negativo (VN):
0	25	

Figura N° 41. Matriz de confusión para Regresión Logística

Tener una cantidad nula de falsos negativos parecería algo bastante bueno debido a que implicaría que la Recuperación es igual a 1. Sin embargo, que el algoritmo clasifique de manera perfecta todas las clases positivas reales se le puede atribuir al hecho de que el algoritmo se está esforzando más de la cuenta en clasificar bien la clase positiva. Esto explicaría porque se tienen tantos falsos positivos. Además, tener una alta recuperación implicaría, en la mayoría de los casos, tener una baja precisión.

Veamos ahora las métricas:

Tabla N° 11. Métricas para la Regresión Logística

Métricas	Puntaje
Exactitud	0.850
Precisión	0.625
Recuperación	1.000
AUC-ROC	0.970

Efectivamente se puede comprobar que el algoritmo al tener una buena Recuperación está arriesgando Precisión para lograr tal fin. En los algoritmos de clasificación siempre se debe buscar el equilibrio entre estas dos métricas las cuales se consideran las más importantes.

Ahora bien, vemos que el algoritmo tiene una AUC de la curva ROC bastante alto, 0.970, lo que quiere decir que el algoritmo de Regresión Logística se siente

bastante seguro al clasificar la clase positiva como ya se mencionó antes. Sin embargo, para lograr tal seguridad se arriesga precisión.

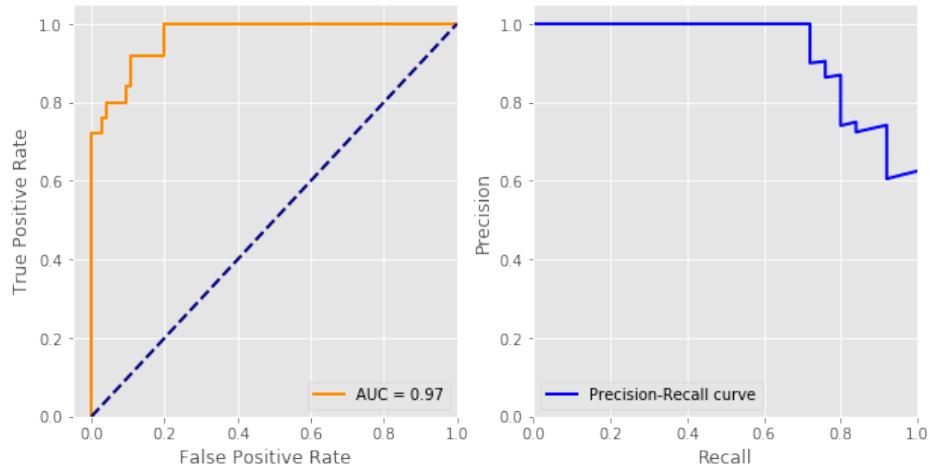


Figura N° 42. Curva ROC y Curva Precisión-Recuperación de la Regresión Logística

V.2.7.2 Árboles de decisión

Tabla N° 12. Resultados de los Árboles de Decisión

Máquinas	1	2	3	4	-	91	92	93	94	95	96	97	98	99	100
RUL	112	98	69	82	-	38	20	85	55	128	137	82	59	117	20
Reales	0	0	0	0	-	0	1	0	0	0	0	0	0	0	1
Predichos	0	0	1	0	-	1	1	1	1	0	0	0	1	0	1

Ahora veamos la matriz de confusión:

	Verdadero positivo (VP):	Falso positivo (FP):
60	15	
Falso negativo (FN):		Verdadero negativo (VN):
1	24	

Figura N° 43. Matriz de confusión para Árboles de decisión

En este caso, en los Árboles de Decisión vemos que el número de verdaderos positivos y falsos positivos es igual al que se obtuvo con la Regresión Logística, lo que quiere decir que el algoritmo sigue esforzándose por querer clasificar correctamente la clase positiva. No obstante, vemos que con este algoritmo se encontró un falso negativo, es decir, clasificó mal la clase negativa, lo que se traduce en que el algoritmo a pesar de que se está esforzando en clasificar bien la clase positiva, hubo un ejemplo el cual él lo clasificó como positivo a pesar de que la probabilidad de seguridad era baja. Esto se podrá observar en la métrica AUC-ROC siendo esta más pequeña en comparación a la de la Regresión Logística.

Tabla N° 13. Métricas para los Árboles de Decisión

Métricas	Puntaje
Exactitud	0.840
Precisión	0.615
Recuperación	0.960
AUC-ROC	0.954

Efectivamente, como se observa en las métricas, el valor del AUC de la curva ROC disminuyó con respecto al valor de la Regresión Logística, lo que quiere decir que los Árboles de Decisión tienen menos seguridad a la hora de clasificar la clase positiva.

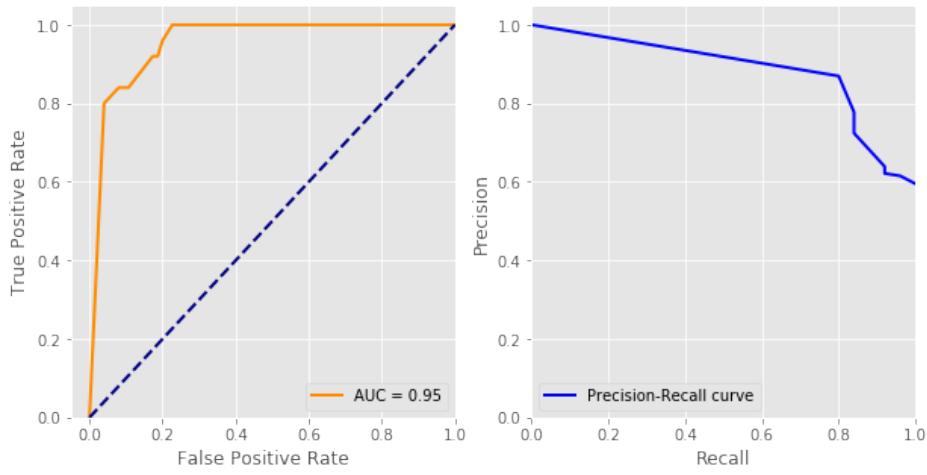


Figura N° 44. Curva ROC y Curva Precisión-Recuperación para Árboles de Decisión

V.2.7.3 Bosques aleatorios

Tabla N° 14. Resultados de los Bosques aleatorios

Máquinas	1	2	3	4	-	91	92	93	94	95	96	97	98	99	100
RUL	112	98	69	82	-	38	20	85	55	128	137	82	59	117	20
Reales	0	0	0	0	-	0	1	0	0	0	0	0	0	0	1
Predichos	0	0	1	0	-	1	1	1	1	0	0	0	0	0	1

Ahora veamos la matriz de confusión:

	Verdadero positivo (VP):	Falso positivo (FP):
	60	15
	Falso negativo (FN):	Verdadero negativo (VN):
	0	25

Figura N° 45. Matriz de confusión para los Bosques aleatorios

Con los Bosques Aleatorios tanto la matriz de confusión como las métricas fueron idénticas a la Regresión Logística. Por lo tanto fue necesario seguir probando el resto de algoritmos.

Tabla N° 15. Métricas para los Bosques Aleatorios

Métricas	Puntaje
Exactitud	0.850
Precisión	0.625
Recuperación	1.000
AUC-ROC	0.970

V.2.7.4 Vectores soportados lineales

Tabla N° 16. Resultados de los Vectores Soportados

Máquinas	1	2	3	4	-	91	92	93	94	95	96	97	98	99	100
RUL	112	98	69	82	-	38	20	85	55	128	137	82	59	117	20
Reales	0	0	0	0	-	0	1	0	0	0	0	0	0	0	1
Predichos	0	0	1	0	-	1	1	1	1	0	0	0	0	0	1

Ahora veamos la matriz de confusión:



Figura N° 46. Matriz de confusión para los vectores soportados

Con los Vectores soportados encontramos mejor rendimiento en comparación

con los algoritmos anteriores. Esto se puede ver porque tiene una menor cantidad de falsos positivos, lo cual es bueno porque el clasificador tiene mayor probabilidad de acierto en la clase positiva.

Tabla N° 17. Métricas para los Vectores soportados

Métricas	Puntaje
Exactitud	0.890
Precisión	0.705
Recuperación	0.960
AUC-ROC	0.975

Como esperábamos, las métricas son mucho mejores que las anteriores y eso se puede observar al comparar las matrices de confusión de los algoritmos. Un punto importante de resaltar es que vemos como aumentó la precisión aun cuando la recuperación es bastante alta, esto es lo que se busca en los algoritmos de clasificación. Además, los vectores soportados lineales son los que han tenido mayor seguridad a la hora de clasificar la clase positiva (ver métrica AUC-ROC).

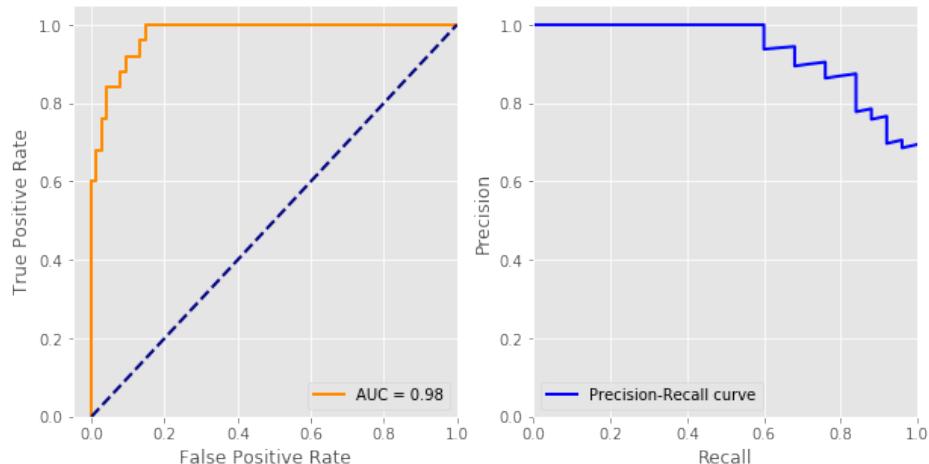


Figura N° 47. Curva ROC y Curva Precisión-Recuperación para los vectores soportados lineales

V.2.7.5 K-Vecinos (KNN)

Por último, veamos el algoritmo propuesto:

Tabla N° 18. Resultados de K-Vecinos

Máquinas	1	2	3	4	-	91	92	93	94	95	96	97	98	99	100
RUL	112	98	69	82	-	38	20	85	55	128	137	82	59	117	20
Reales	0	0	0	0	-	0	1	0	0	0	0	0	0	0	1
Predichos	0	0	0	0	-	1	1	0	1	0	0	0	0	0	1

Ahora la matriz de confusión:



Figura N° 48. Matriz de confusión para los K-Vecinos

Podemos ver como los K-Vecinos ha sido el algoritmo que mayor cantidad de verdaderos positivos ha alcanzado. No obstante, también es el que mayor cantidad de falsos negativos tiene, lo que significa que aunque el algoritmo en gran medida clasifique bien la clase positiva, probablemente lo haga sin tener una probabilidad alta de acierto, en palabras claras, es muy probable que no tenga mucha seguridad en clasificar la clase positiva y de igual manera lo hace.

Tabla N° 19. Métricas para K-Vecinos

Métricas	Puntaje
Exactitud	0.900
Precisión	0.800
Recuperación	0.800
AUC-ROC	0.957

Con las métricas podemos corroborar lo que se comentó anteriormente. Al ver el AUC-ROC nos damos cuenta que, junto con los Árboles de Decisión, los K-vecinos es el algoritmo que menos seguridad tiene a la hora de clasificar la clase positiva. Por otro lado, la recuperación también es la menor de todos los algoritmos y eso influye en la seguridad que tiene el algoritmo. Por lo tanto, no será el algoritmo seleccionado.

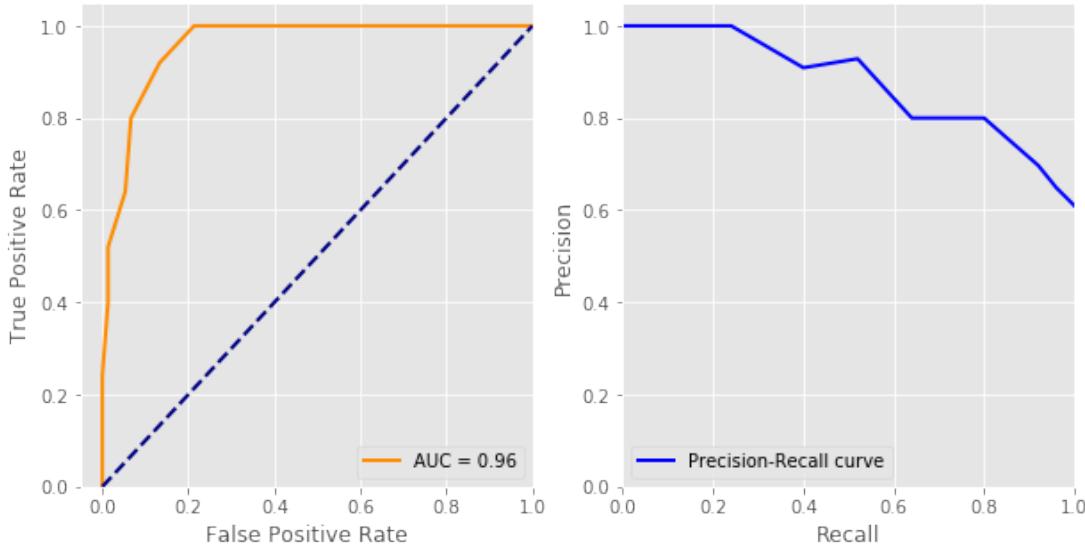


Figura N° 49. Curva ROC y Curva Precisión-Recuperación para los K-Vecinos

Ahora, se mostrarán las métricas finales obtenidas por los 5 algoritmos.

Tabla N° 20. Métricas de todos los algoritmos

Métricas	Regresión L.	Árboles	Bosques	Vectores	K-Vecinos
Exactitud	0.850	0.840	0.850	0.890	0.900
Precisión	0.625	0.615	0.625	0.706	0.800
Recuperación	1.000	0.960	1.000	0.960	0.800
Valor F	0.770	0.750	0.770	0.813	0.900
AUC-ROC	0.970	0.954	0.970	0.975	0.957

Por último, las gráficas de la curva ROC con su respectivo puntaje de AUC y la curva Precisión-Recuperación.

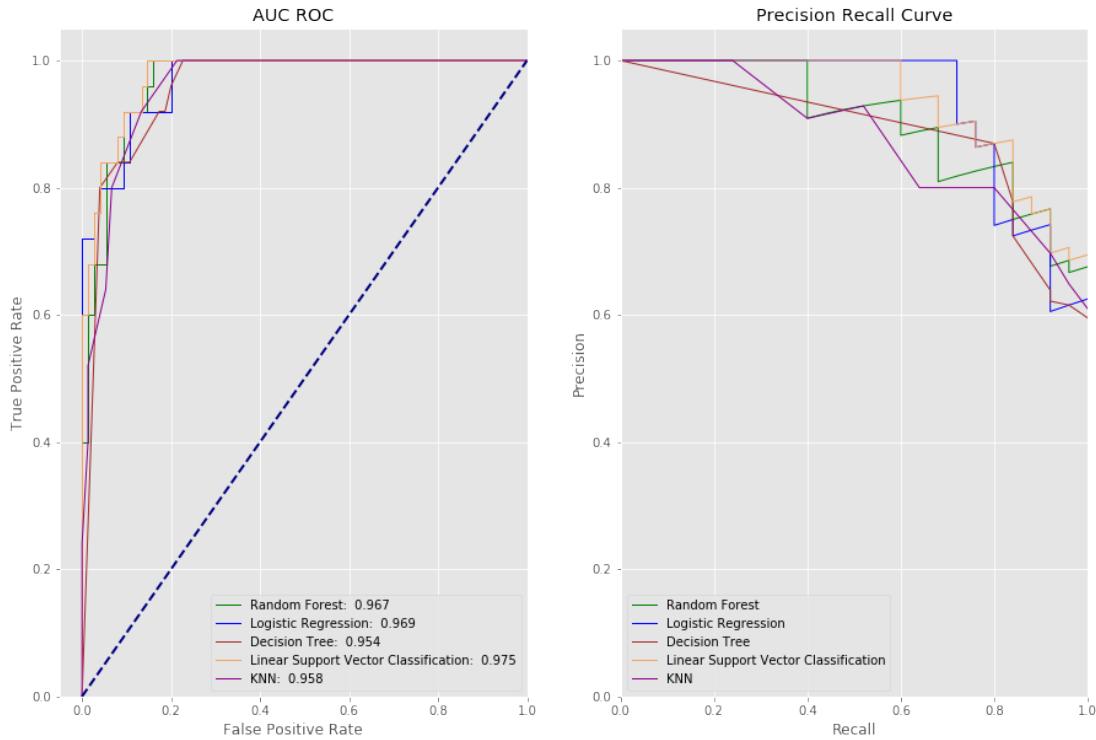


Figura N° 50. Curva ROC y Curva Precisión-Recuperación para todos los algoritmos

Como se puede ver en la comparación global, el mejor algoritmo, tanto a nivel de métricas como en la seguridad de clasificación fue el algoritmo de Vectores

Soportados Lineales. Este algoritmo fue capaz de lograr la detección de fallas con una probabilidad de éxito del 97.5%.

Antes de finalizar este objetivo, se hizo la comparación de los resultados obtenidos con los resultados que se obtuvieron al aplicar los mismo algoritmos de clasificación pero esta vez a la data completa, es decir, sin haber realizado la reducción de dimensiones.

Tabla N° 21. Métricas obtenidas al usar los datos con todos los sensores (Sin reducción de dimensiones)

Métricas	Regresión L.	Árboles	Bosques	Vectores	K-Vecinos
Exactitud	0.860	0.810	0.860	0.850	0.910
Precisión	0.641	0.568	0.641	0.631	0.735
Recuperación	1.000	1.000	1.000	0.960	1.000
Valor F	0.781	0.725	0.781	0.762	0.847
AUC-ROC	0.980	0.930	0.980	0.976	0.972

Se puede observar como las métricas son muy parecidas, y en el caso de la Recuperación y la Precisión son más deficientes en comparación con la Tabla N° 20. De esta manera se demuestra que la reducción de dimensiones y más general, el pretratamiento correcto de los datos, es un paso fundamental y critico para obtener mejores resultados ya que así se retiran datos irrelevantes y redundantes que pueden perjudicar el rendimiento de los algoritmos de aprendizaje automático.

V.2.8 Desarrollar un modelo para estimar la degradación del desempeño de un turbofan a partir de datos típicos de operación.

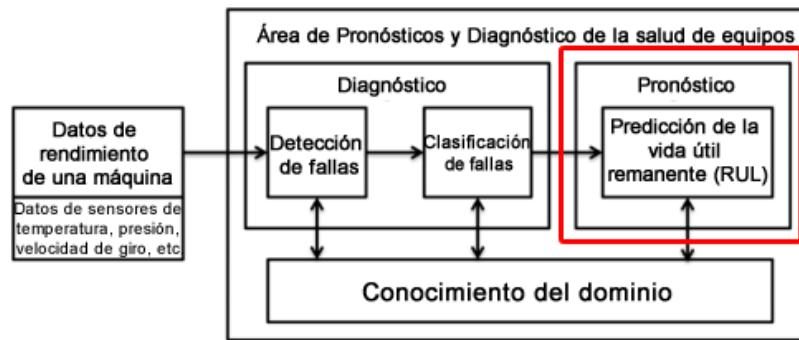


Figura N° 51. Ilustración del recorrido en el área de Pronósticos y Diagnósticos de la salud - Sección Pronóstico

Ahora necesitaremos predecir cuánto tiempo, en ciclos, les restan a las máquinas en los distintos archivos de prueba. Para ello aplicamos distintos algoritmos de regresión que nos permitieran obtener una predicción satisfactoria.

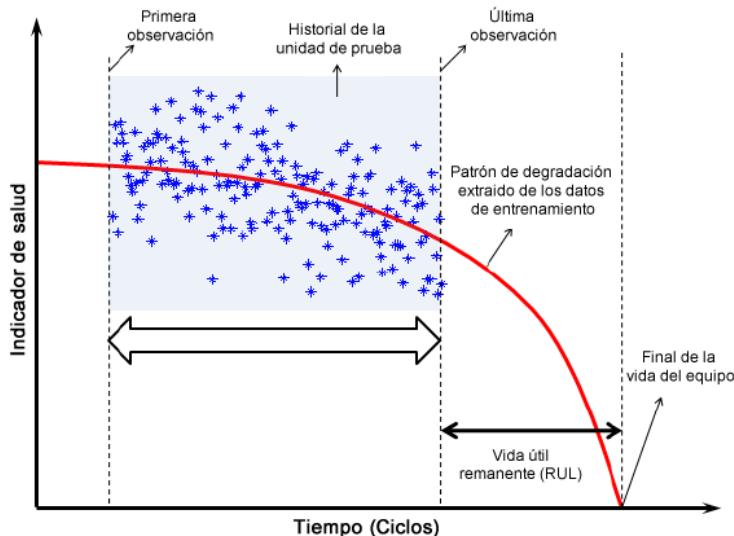


Figura N° 52. Ilustración de las etapas de la degradación de un activo representando el rango del RUL (Wang et al., 2008).

Basados en el hecho de que los ciclos aumenta linealmente con el orden de los puntos de datos para cada máquina (como se muestra en la Tabla N° 22), debemos esperar que nuestro RUL disminuya linealmente en relación con el ciclo para cada punto de datos en cada máquina.

Tabla N° 22. Estructura de los datos de entrenamiento hasta este punto.

Id	cycle	Opset_Id	Sensor4	Sensor9	Sensor11	Sensor12
1	1	0	-0.980	-0.870	-0.230	0.365
1	2	1	-0.688	-0.965	-0.216	1.225
1	3	3	-0.567	-0.562	-1.057	1.419
-	-	-	-	-	-	-

En primer lugar, se creó una columna adicional en el set de entrenamiento que representó el tiempo de vida útil remanente de las máquinas. Recordando que en la base de datos de entrenamiento si se conoce la cantidad de ciclos totales que dura en operación los turbofan.

Para ello aplicaremos la siguiente formula:

$$RUL(x) = \begin{cases} c - x, & x \leq c \\ 0, & x = 0 \end{cases} \quad (5)$$

Dónde: x es el número del ciclo para el registro evaluado y c es el ciclo máximo para la máquina evaluada.

Tabla N° 23. Estructura de los datos de entrenamiento con la columna RUL.

Id	cycle	Opset_Id	Sensor4	Sensor9	—	RUL
1	1	0	-0.980	-0.870	—	191
1	2	1	-0.688	-0.965	—	190
1	3	3	-0.567	-0.562	—	189
—	—	—	—	—	—	—
1	192	13	2.059	-1.460	—	0
—	—	—	—	—	—	—

De manera adicional, se creó otra columna que donde se tendría el RUL normalizado en el rango de 1-0, donde 1 representa la máquina nueva y 0 representa la máquina fallada.

Tabla N° 24. Estructura de los datos de entrenamiento del archivo 1 con la columna RUL y RUL_norm.

Id	cycle	Opset_Id	Sensor4	Sensor9	—	RUL	RUL_norm
1	1	0	-0.980	-0.870	—	191	0.995
1	2	1	-0.688	-0.965	—	190	0.989
1	3	3	-0.567	-0.562	—	189	0.984
—	—	—	—	—	—	—	—
1	192	13	2.059	-1.460	—	0	0.000
—	—	—	—	—	—	—	—

Los algoritmos que emplearon en esta fase fueron:

- Regresión Lineal (R.L)
- Regresión Polinomial (R.P)
- Árboles de Decisión (A.D)
- Bosques Aleatorios (B.A)

De igual forma como se realizó en la fase de diagnóstico, a cada algoritmo se le asignó como función objetivo la columna *RUL norm* y se entrenó con la configuración de operación y los sensores correspondientes para cada base de datos.

Una vez que ya estuviera entrenado, el algoritmo se probó con la base de datos de prueba para determinar el tiempo de vida útil remanente predicho por los algoritmos correspondientes.

Las métricas empleadas de regresión fueron:

- Raíz del error cuadrático medio (RMSE)
- Coeficiente de determinación (R^2)

El entrenamiento y prueba de los algoritmos de regresión fue análoga a los algoritmos de clasificación. Por lo tanto se mostrarán directamente los resultados y las métricas de cada algoritmo.

Tabla N° 25. Resultados de los algoritmos de Regresión

Máquinas	1	2	3	4	...	97	98	99	100
RUL	112	98	69	82	...	82	59	117	20
R.L	141	87	19	92	...	72	-51	127	-109
R.P	142	87	5	89	...	-48	44	104	46
A.D	150	46	5	60	...	40	46	90	10
B.A	148	54	10	85	...	52	39	78	12

Como se puede observar, los algoritmos Regresión Lineal y Regresión Polinomial son los únicos que devolvieron tiempos de vida útil remanente negativos en algunas máquinas, y dicho resultado no tiene significado físico, por lo tanto, estos algoritmos no se tomarán en consideración.

En cuanto a los Árboles de Decisión y Bosques Aleatorios, devolvieron sólo resultados positivos pero en muchos casos, alejados de la respuesta real.

Tabla N° 26. Métricas de todos los algoritmos de regresión

Métricas	R.L	R.P	A.D	B.A
RMSE	42.184	38.490	37.575	32.140
R^2	-0.030	0.142	0.182	0.402

Vemos como la raíz del error cuadrático medio disminuye a medida que se

ejecutaron los algoritmos. Los Bosques Aleatorios fueron los que presentaron el menor puntaje y al ser una métrica de puntaje negativo, esto es lo ideal. Sin embargo, el poder de predicción (R^2) que tienen los algoritmos es muy pobre para decidirse por alguno de estos algoritmos.

De igual manera como se hizo con la fase de clasificación, se compararan estos resultados con los que se obtuvieron al probar los mismos algoritmos con la base de datos sin haberle realizado la reducción de dimensiones.

Tabla N° 27. Métricas de todos los algoritmos de regresión al usar la base de datos completa

Métricas	R.L	R.P	A.D	B.A
RMSE	40.200	52.467	36.187	26.323
R^2	0.064	-0.594	0.242	0.598

De manera similar que la fase anterior, al comparar estos resultados con los de la Tabla N° 26 los resultados son muy similares cuando se usan 21 sensores a cuando se usan los sensores obtenidos mediante la reducción de dimensiones (4 en promedio), lo que quiere decir que la conclusión es la misma a la que se llegó en el objetivo anterior. La reducción de dimensiones es necesaria para evitar trabajar con exceso de datos que obstaculizan el aprendizaje y el mejor rendimiento de los algoritmos.

A pesar de que se podía intentar con otros algoritmos de regresión, por ejemplo, Vectores soportados, Regresión LASSO o Regresión Ridge para tener más algoritmos de comparación, se decidió intentar con una Red Neuronal Recurrente debido a que se tiene referencia (Zheng et al., 2017) de que su poder de detección de patrones y ajuste es mucho más poderoso que los algoritmos de regresión convencionales de aprendizaje automático.

V.2.8.1 Red Neuronal Recurrente

Ahora se empleará una Red Neuronal Recurrente del tipo LSTM. El tipo LSTM significa que se utilizarán neuronas del tipo Long-Short Term Memory (Memoria corta a largo plazo) debido a que se tiene referencias (Brownlee, 2017) de que este

tipo de neuronas son poderosas al trabajar con series de tiempo secuenciales en la detección de anomalías.

Para trabajar con este tipo de red neuronal, es necesario establecer una ventana de operación, también conocida como *Longitud de la secuencia*, debido a que las neuronas LSTM trabajan por lotes de información. La ventana óptima que se estableció fue de 50 datos (Zheng et al., 2017). Lo que quiere decir, por ejemplo, que para la máquina 1, que tiene 192 filas con 23 columnas, la estructura es la siguiente:

1. 0 - 50 -> Datos de los sensores de la fila 0 a la fila 50 (Muestra 1).
2. 1 - 51 -> Datos de los sensores de la fila 1 a la fila 51 (Muestra 2).
3. ...
4. 141 - 191 -> Datos de los sensores de la fila 141 a la fila 191 (Muestra 50).

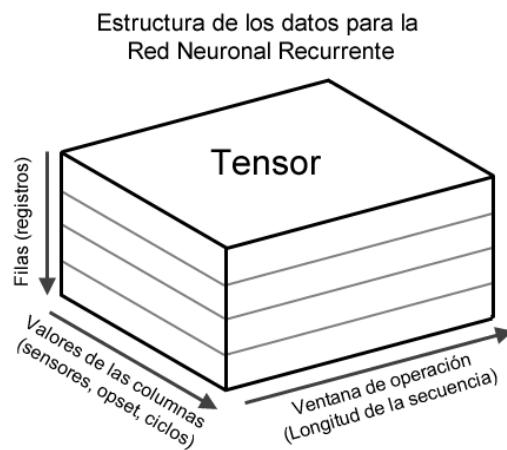


Figura N° 53. Estructura de los datos para una Red Neuronal Recurrente

Quedando la estructura total de los datos para el ejemplo anterior como: (142 x 50 x 23). Esto se realizó con todas las bases de datos.

Una ventaja poderosa de este tipo de red neuronal es que se le incluye el tiempo, por lo tanto, es capaz de aprender de los datos y de las secuencias a los cuales estos pertenecen.

En la fase entrenamiento se probaron con distintas arquitecturas. Sin embargo, como lo reporta Zheng et al. (2017), la mejor arquitectura para ellos fue el tipo LSTM(64,32) NN(8,8), que significa que utilizaron 2 capas LSTM. La prime-

ra con 32 neuronas y la segunda con 64 neuronas. Las últimas dos capas tienen neuronas básicas del tipo perceptrón con 8 neuronas en cada capa. Además, emplearon como función de optimización *Rmsprop* que es una mejora del *Descenso del Gradiente Estocástico*. Además, se utilizaron las mismas métricas de regresión que se emplearon con los algoritmos de aprendizaje automático, es decir, RMSE y R^2 . Por otra parte, el 10% de la base de datos de entrenamiento se utilizó para validar la red neuronal a medida que se iba entrenando y corroborar la función de pérdida, que para dicha función la métrica utilizada fue el error cuadrático medio. Finalmente, la función objetivo que se le dio a la red neuronal para que se entrena y ajustará fue la columna de tiempo de vida útil remanente normalizado de 1 a 0.

Al terminar el entrenamiento de la RNN se observó la gráfica de pérdida para corroborar la convergencia de la red neuronal.

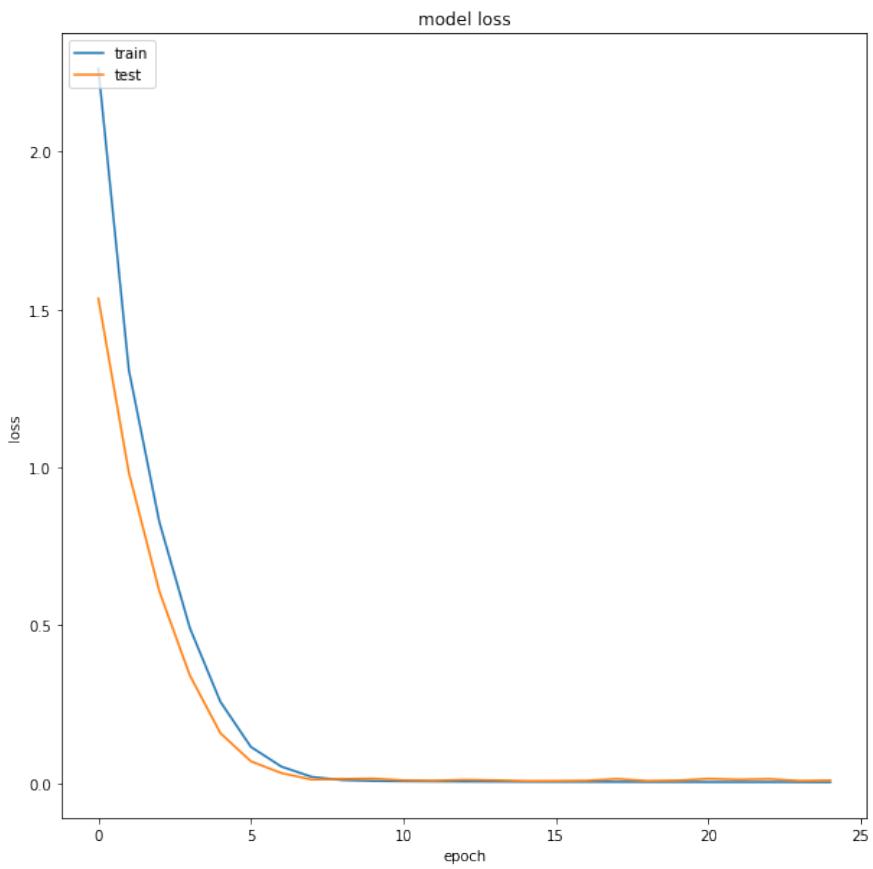


Figura N° 54. Gráfica de la pérdida o error de los datos de entrenamiento y los de validación de la RNN.

Se observa como la pérdida del set de entrenamiento y el de validación al principio de las épocas son diferentes, pero luego convergen hacia el mismo punto, lo que se interpreta como el hecho de que la red neuronal fue capaz de entrenarse correctamente evitando el sobreajuste de los datos.

Las métricas que se obtuvieron con los datos de entrenamiento fueron las siguientes:

Tabla N° 28. Métricas del entrenamiento de la RNN

Métricas	Puntaje
RMSE	17.188
R^2	0.897

Se ven que las métricas de entrenamiento fueron bastante buenas, sin embargo no son suficientes para determinar el poder predictivo de la RNN. Por lo tanto, se procedió a probar la red con los datos de prueba para comprobar los resultados y posteriormente las métricas.

Tabla N° 29. Resultados de la RNN

Máquinas	1	2	3	4	...	97	98	99	100
RUL_norm	0.354	0.436	0.481	0.474	...	0.380	0.327	0.546	0.092
Predichos	0.209	0.265	0.390	0.432	...	0.301	0.165	0.454	0.014

Los resultados que estamos observando es la cantidad (en la escala de 1 a 0) de vida remanente que le queda a las máquinas. Por ejemplo, para la máquina 1, la red neuronal predijo que le restaba el 20% pero en realidad le resta el 35%. A pesar de que pueda parecer un diferencia alarmante, es importante mencionar que para mejorar esa desviación se puede mejorar la arquitectura de la red neuronal, usando más o menos capaz, empleando regularizadores, o simplemente entrenando a la red con más datos.

Tabla N° 30. Métricas del prueba de la RNN

Métricas	Puntaje
RMSE	25.383
R^2	0.755

Se observa, como era de esperarse, una variación en las métricas que se obtuvieron con los datos de prueba. Estás métricas reportan un puntaje coherente debido a que la red realizó la predicción de datos que no había visto antes y es normal esperar una disminución en las métricas de ajuste. Sin embargo, es importante acotar que la red neuronal aprende de todos los datos nuevos que se le asignen y se mejora automáticamente.

De esta manera se observa una mejora significativa al comparar la RNN con los algoritmos de regresión de aprendizaje automático.

V.2.9 Caracterizar el proceso de degradación empleando los parámetros: índice de degradación y tiempo de vida útil remanente.

Para cumplir con el último objetivo específico, se realizó una gráfica que representará el índice de degradación de cada máquina en función de los ciclos operativos restantes. Para realizar dicha gráfica se le suministró a la RNN las siguientes variables de la base de datos de prueba:

- Condición de operación.
- Ciclos de operación.
- Sensores.

Al suministrar dichas variables, la RNN fue capaz de determinar la progresión de la vida útil de la máquina para el resto de ciclos operativos. Para mayor simplicidad, se mostrará la solución obtenida por la máquina 15.

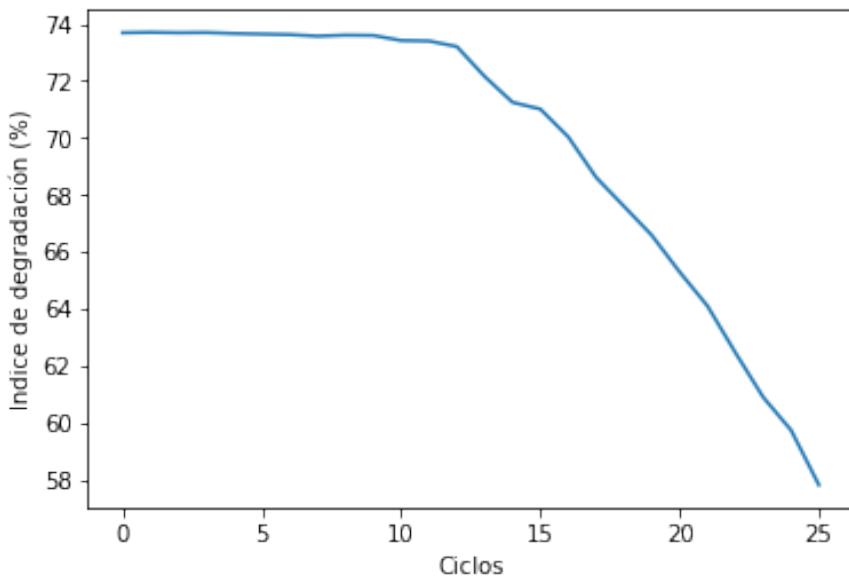


Figura N° 55. RUL vs Ciclos para una máquina escogida al azar de la base de datos de prueba.

En primer lugar, se observa como la gráfica parte desde un porcentaje de vida útil del 74% aprox. Lo que quiere decir, que los datos para la máquina 15 que se tienen en la base de datos de prueba, se reportan hasta el 74% de vida.

En segundo lugar, vemos como la vida útil de la máquina se mantiene prácticamente constante por un periodo de aproximadamente 10 ciclos, luego de ese tiempo, la degradación comienza y el porcentaje de vida disminuye significativamente.

En tercer y último lugar, vemos como la gráfica obtenida se aproxima bastante bien a lo que se muestra en la figura N° 25 y en la figura N° 52, que coincide con lo que se viene reportando en la mayoría de las investigaciones que abordan el tema de detección de anomalías, como por ejemplo, la investigación de Li y Nilkitsaranont, Wang et al., entre otras.

Ahora bien, teniendo la predicción de la RNN, se propuso determinar un modelo que ajustará lo mejor posible estos datos, y de ese modo, tener una expresión matemática del índice de degradación de la máquina en cuestión.

Al observar la figura N° 55, vemos que la tendencia de la curva pareciera seguir un modelo exponencial. Además, tal cual como lo reporta Meher-Homji et al. en su investigación del año 2004 al mostrar el índice de ensuciamiento de un compresor axial, el cual sigue un comportamiento exponencial, la regresión del tipo exponencial parece ser el ajuste que mejor representa las curvas de índice de degradación (ID).

$$y = A \cdot e^{B \cdot x} + C \quad (6)$$

Al realizar el ajuste de los datos predichos, se obtuvo la siguiente ecuación:

$$ID = -0,00561298 \cdot e^{0,13873498 \cdot Ciclos} + 0,749745 \quad (7)$$

La ecuación (7) es la ecuación que modela la salud de la máquina 15 en función de los ciclos. Esta ecuación es la representación del índice de degradación de dicha máquina en función los ciclos operativos.

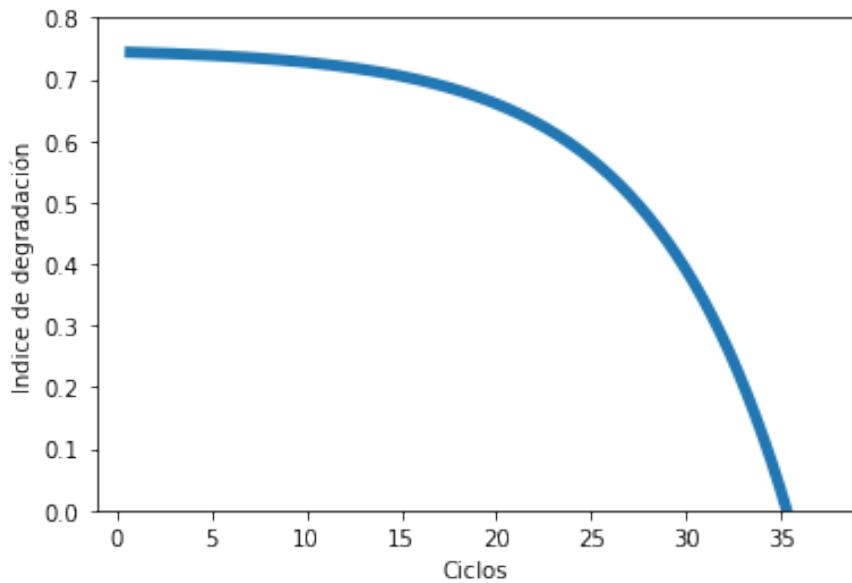


Figura N° 56. Comportamiento del índice de degradación al evaluarla en distintos ciclos.

Viendo la figura anterior, se puede establecer el inicio de la vida útil remanente desde el punto de inflexión hasta que el índice de degradación se hace cero. De esta manera, podemos caracterizar la máquina 15 empleando el índice de degradación y el tiempo de vida útil remanente gracias al modelo matemático obtenido a partir de los datos predichos por la RNN.

Para acceder a los códigos desarrollados a lo largo de este Trabajo Especial de Grado se invita a visitar el siguiente repositorio de GitHub:

<https://github.com/Portilloronny6/prueba>

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

VI.1 CONCLUSIONES

- El área de Pronósticos y Diagnósticos de la salud de equipos está presentando un intenso auge que ha llevado a la NASA a participar en la generación de datos sintéticos de diferentes turbomáquinas.
- El preprocesamiento es la fase que más tiempo consume y es a la que más cuidado hay que prestarle. Específicamente en la limpieza de los datos y remoción de valores atípicos.
- Se demostró que la reducción de dimensiones es muy importante ya que al reducir la dimensión de los datos, se obtienen resultados más precisos en comparación al caso en el que se usan todas las dimensiones.
- El algoritmo de Vectores de Soportes Lineales fue el mejor en la detección de fallas con una seguridad de acierto en la detección del 97.5% en los 4 archivos de la base de datos.
- En la fase de regresión, todos los algoritmos que se emplearon arrojaron métricas deficientes. Por lo tanto, este método de obtención de la vida útil remanente fue el menos adecuado.
- La Red Neuronal Recurrente del tipo LSTM fue capaz de determinar el RUL con unas métricas de ajuste y de poder predictivo mucho mejor que los algoritmos de aprendizaje automático.
- Se logró obtener la curva Indice de degradación vs Ciclos empleando los valores predichos de la RNN. Esta curva presentó el mismo comportamiento de otras investigaciones del área.

VI.2 RECOMENDACIONES

A fin de dar continuidad al trabajo realizado, se presentan las siguientes recomendaciones que permitirán abarcar este tema en mayor profundidad.

- Realizar el análisis del procesamiento de la señal empleando la transformada rápida de fourier, para retirar las frecuencias asociadas al ruido para tener una data más limpia y óptima.
- Efectuar la selección de características con otro algoritmo y comparar los resultados.
- Probar con otros algoritmos de Regresión, como por ejemplo: Regresión LASSO, Regresión Ridge, Vectores de soportes, entre otros.
- Mejorar la arquitectura de la Red Neuronal Recurrente LSTM y los parámetros asociados para obtener mejores resultados.
- Trabajar con el resto de los archivos del repositorio de la NASA para fortalecer el aprendizaje de los algoritmos de clasificación, regresión y red neuronal con la finalidad de que sean capaces de obtener patrones de multiples datos y con diferentes degradaciones.

BIBLIOGRAFÍA

- Brownlee, J. (2017). Long short-term memory networks with python develop sequence prediction models with deep learning. *Machine Learning Mastery, EBook*.
- Bruce, P. y Bruce, A. (2017). *Practical Statistics for Data Scientists: 50 Essential Concepts.* ” O'Reilly Media, Inc.”.
- Chen, K.-Y., Chen, L.-S., Chen, M.-C. y Lee, C.-L. (2011). Using svm based method for equipment fault detection in a thermal power plant. *Computers in industry*, 62(1): 42–50.
- Chung, J., Gulcehre, C., Cho, K. y Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- Echeverri, L. I., Arroyave, E. A. y Barajas, F. H. (2015). Comparación de pruebas de normalidad. *XXI Simp. Int. Estad*, pages 8–11.
- Ernest Vilajosana, V. (2011). *El motor de turbina*.
- Gil Tolmo, C. (2013). Selección y diseño de compresores centrífugos. Trabajo de grado de Maestría no publicado.
- González, L. M. C. (2015). Análisis de metodologías para la identificación de la degradación en turbinas de gas aplicación en el motor rolls-royce rb211.
- Granero, J. M. L. (2012). Estudio de un turbofan.
- Kurz, R. y Brun, K. (2001). Degradation in gas turbine systems. *Journal of Engineering for Gas Turbines and Power*, 123(1): 70–77.
- Lazzaretto, A. y Toffolo, A. (2001). Analytical and neural network models for gas turbine design and off-design simulation. *International Journal of Thermodynamics*, 4(4): 173–182.
- Leyva Leyva, H. E. Análisis microestructural de la superaleación udimet 700 utilizada en un álabe de compresor para uso aeronáutico.

- Li, Y. y Nilkitsaranont, P. (2009). Gas turbine performance prognostic for condition-based maintenance. *Applied energy*, 86(10): 2152–2161.
- Li, Y.-G. (2017). Diagnostics of power setting sensor fault of gas turbine engines using genetic algorithm. *The Aeronautical Journal*, 121(1242): 1109–1130.
- Martín-Aragón Gutiérrez, J. *Método para optimizar la planificación de los lavados “off-line” en compresores de turbinas de gas*. PhD thesis, Industriales, 2016.
- Meher-Homji, C. B., Bromley, A. et al. Gas turbine axial compressor fouling and washing. In *Proceedings of the 33rd Turbomachinery Symposium*. Texas A&M University. Turbomachinery Laboratories, 2004.
- Müller, A. C. y Guido, S. (2016). *Introduction to machine learning with Python: a guide for data scientists.* ” O'Reilly Media, Inc.”.
- Qi, G., Zhu, Z., Erqinhui, K., Chen, Y., Chai, Y. y Sun, J. (2018). Fault-diagnosis for reciprocating compressors using big data and machine learning. *Simulation Modelling Practice and Theory*, 80: 104–127.
- Ramasso, E. y Saxena, A. Review and analysis of algorithmic approaches developed for prognostics on cmapss dataset. In *Annual Conference of the Prognostics and Health Management Society 2014.*, 2014.
- Raschka, S. (2015). *Python machine learning*. Packt Publishing Ltd.
- Reif, M., Goldstein, M., Stahl, A. y Breuel, T. M. Anomaly detection by combining decision trees and parametric densities. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.
- Richter, H. (2011). *Advanced control of turbofan engines*. Springer Science & Business Media.
- Rosario, D. S. Highly effective logistic regression model for signal (anomaly) detection. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages V–817. IEEE, 2004.
- Saxena, A. y Goebel, K. (2008). Turbofan engine degradation simulation data set. *NASA Ames Prognostics Data Repository*.

- Syverud, E. y Bakken, L. E. Online water wash tests of ge j85-13. In *ASME Turbo Expo 2005: Power for Land, Sea, and Air*, pages 339–347. American Society of Mechanical Engineers, 2005.
- Tabakoff, W., Lakshminarasimha, A. y Pasin, M. Simulation of compressor performance deterioration due to erosion. In *ASME 1989 International Gas Turbine and Aeroengine Congress and Exposition*, pages V003T05A005–V003T05A005. American Society of Mechanical Engineers, 1989.
- Thang, T. M. y Kim, J. The anomaly detection by using dbSCAN clustering with multiple parameters. In *Information Science and Applications (ICISA), 2011 International Conference on*, pages 1–5. IEEE, 2011.
- Wang, C., Lu, N., Cheng, Y. y Jiang, B. Deep forest based multivariate classification for diagnostic health monitoring. In *2018 Chinese Control And Decision Conference (CCDC)*, pages 6233–6238. IEEE, 2018.
- Wang, T., Yu, J., Siegel, D. y Lee, J. A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, pages 1–6. IEEE, 2008.
- Wu, S.-j., Gebraeel, N., Lawley, M. A. y Yih, Y. (2007). A neural network integrated decision support system for condition-based optimal predictive maintenance policy. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 37(2): 226–236.
- Zhao, Z., Yang, J., Lu, W. y Wang, X. Application of local outlier factor method and back-propagation neural network for steel plates fault diagnosis. In *Control and Decision Conference (CCDC), 2015 27th Chinese*, pages 2416–2421. IEEE, 2015.
- Zheng, S., Ristovski, K., Farahat, A. y Gupta, C. Long short-term memory network for remaining useful life estimation. In *Prognostics and Health Management (ICPHM), 2017 IEEE International Conference on*, pages 88–95. IEEE, 2017.