



Universidad de
Oviedo



ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN.

GRADO EN INGENIERÍA INFORMÁTICA EN TECNOLOGÍAS DE LA INFORMACIÓN

ÁREA DE CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL

MINERÍA DE ANOMALÍAS APLICADA AL DIAGNÓSTICO DE MOTORES A REACCIÓN

GONZÁLEZ LÓPEZ, Laura
TUTOR: D. SÁNCHEZ RAMOS, Luciano

FECHA: Julio de 2021

ÍNDICE

1. Hipótesis de partida y alcance.....	5
1.1.- INTRODUCCIÓN.....	5
1.2.- MOTIVACIÓN	6
2. Objetivos concretos y relación con el estado actual	7
2.1.- OBJETIVOS DEL TRABAJO	7
2.2.- MONITORIZACIÓN DE MOTORES A REACCIÓN	8
2.3.- DETECCIÓN DE ANOMALÍAS	13
2.3.1.- Conceptos y tipos de anomalías	13
2.3.2.- Técnicas tradicionales	14
2.3.3.- Técnicas de Deep Learning.....	18
2.4.- ANÁLISIS DE COMPONENTES PRINCIPALES (PCA)	33
3. Metodología de trabajo.....	36
3.1.- DATOS UTILIZADOS	37
3.1.1.- Conjunto de datos “CMAPPS” (NASA).....	38
3.1.2.- Conjunto de datos “Rolls” (Datos reales)	39
3.2.- HERRAMIENTAS	40
3.3.- PREPROCESAMIENTO	41
3.4.- DISEÑO	43
3.4.1.- Diseño de los datos	44
3.4.2.- Diseño experimental	45
4. Trabajo realizado y resultados obtenidos	49
4.1.- FORMATO Y REPRESENTACIÓN DE LOS DATOS	49
4.2.- PREPARACIÓN Y REDUCCIÓN DE LOS DATOS	49
4.4.- RESULTADOS	53
4.4.1.- Resultados para el dataset CMAPPS (NASA).....	53
4.4.2.- Resultados para el dataset Rolls.....	55
5. Conclusiones y trabajos futuros.....	60
6. Bibliografía	61

ÍNDICE DE ILUSTRACIONES

Ilustración 2.1.- Turborreactor y sus componentes principales [6]	9
Ilustración 2.2.- Temperatura y presión de los componentes del motor a lo largo de su funcionamiento [6]	10
Ilustración 2.3.- Proceso de mantenimiento de un motor a reacción [9]	12
Ilustración 2.4.- Distribución normal [13]	15
Ilustración 2.5.- Representación de un espacio de tres dimensiones (a) y un espacio bidimensional (b) [15]	16
Ilustración 2.6.- Clasificación de las técnicas de Deep Learning de detección de anomalías y los retos (<i>challenges</i>) que aborda cada uno [14]	19
Ilustración 2.7.- K vecinos más cercanos (K Nearest Neighbors, KNN) [17]	20
Ilustración 2.8.- Importancia del valor K [17]	21
Ilustración 2.9.- Ejemplo de las distancias de la técnica Local Outlier Factor [12]	22
Ilustración 2.10.- Diferencia entre distribuciones con variables con y sin correlación [24]	26
Ilustración 2.11.- Representación gráfica de una distribución de variables aleatorias [25]	28
Ilustración 2.12.- Distancia de Mahalanobis en una distribución [25]	28
Ilustración 2.13.- Representación gráfica de un autoencoder [27]	29
Ilustración 2.14.- Funcionamiento de una red de autoencoders [12]	30
Ilustración 2.15.- Reducción lineal (PCA) y no lineal (Autoencoders) de la dimensionalidad [30]	33
Ilustración 2.16.- PCA [33]	35
Ilustración 3.1.- Técnicas para la preparación de datos [4]	42
Ilustración 3.2.- Ejemplo de una interpolación [35]	44
Ilustración 3.3.- 2 ejemplos para la partición de un dataset [36]	46
Ilustración 3.4.- Representación de los árboles de decisión Random Forest [39]	47
Ilustración 3.5.- Representación de la validación cruzada [41]	48
Ilustración 4.1.- Estructura de la red neuronal con autoencoders	52
Ilustración 4.2.- Distancia de Mahalanobis para un avión con la turbina y el compresor en estado "Bueno"	55
Ilustración 4.3.- Distancia de Mahalanobis para un avión con el estado del motor "Alto" ..	56
Ilustración 4.4.- Error en Autoencoders para un avión con el estado del motor "Bueno" ..	57
Ilustración 4.5.- Error en Autoencoders para un avión con el estado del motor "Malo"	58

ÍNDICE DE TABLAS

Tabla 2.1.-Comparación de las técnicas tradicionales con las de Deep Learning [14]	17
Tabla 3.1.- Organización del dataset CMAPPS	38
Tabla 3.2.- Señales de salud de los motores del conjunto de datos CMAPPS	39
Tabla 3.3.- Organización de los datasets del conjunto Rolls.....	39
Tabla 3.4.- Señales de salud del dataset Rolls	40
Tabla 4.1.- Ejemplo de cómo está representado el conjunto de datos de entrenamiento. ...	50
Tabla 4.2.- Porcentaje de anomalías para cada experimento.....	53
Tabla 4.3.- Porcentaje de aciertos para el modelo de Distancia de Mahalanobis.....	54
Tabla 4.4.- Porcentaje de aciertos para el modelo de Autoencoders (100 EPOCHS).....	54
Tabla 4.5.- Porcentaje de aciertos para el modelo de Autoencoders (200 EPOCHS).....	55
Tabla 4.6.- Porcentaje de aciertos para el modelo de Distancia de Mahalanobis para el dataset de Rolls	57
Tabla 4.7.- Porcentaje de aciertos para el modelo de Autoencoders para el dataset de Rolls	59

1. Hipótesis de partida y alcance

1.1.- INTRODUCCIÓN

La inteligencia artificial, una inteligencia similar a la de los seres humanos mostrada por un ordenador, robot u otro tipo de máquina, hoy en día avanza a pasos agigantados para facilitar la resolución de problemas. Una de sus aplicaciones es la detección de anomalías, una técnica que actualmente se utiliza en múltiples campos (medicina, ciberseguridad, negocios...), cada día con más frecuencia. Se puede definir como una técnica de minería de datos utilizada para identificar patrones inusuales que se alejan del comportamiento esperado de un proceso [1]. Estos objetos anómalos son también llamados outliers o “puntos aislados”, ya que cuando son representados gráficamente están lejos de los demás puntos.

Uno de los campos en el que se aplica estas técnicas de detección de puntos anómalos es el de la aviación, donde la monitorización de los motores tiene una gran importancia a la hora de comprobar su salud y rendimiento para llevar a cabo viajes seguros y eficaces. Durante el funcionamiento de estos motores, a reacción, en concreto, pueden aparecer múltiples fallos, como pérdidas de compresión, fallos mecánicos, o defectos el sistema de inyección, entre otros, por eso es de gran importancia detectarlos a tiempo para mejorar su mantenimiento.

En este proyecto, se diseñará un modelo para intentar detectar este tipo de anomalías en diferentes aviones y llevar a cabo una clasificación para diferenciarlos de atípicos o no, centrándonos en el análisis de los diferentes datos que obtendremos del compresor y la turbina del motor. A modo de validación se usarán diferentes técnicas para analizar la tasa de aciertos del clasificador y así comprobar la eficacia del modelo.

1.2.- MOTIVACIÓN

En la actualidad, el uso de este tipo de modelos se utiliza en múltiples ámbitos donde se pueden encontrar puntos atípicos que pueden ser de gran utilidad para realizar diferentes análisis o identificar otros problemas mayores.

Los datos de accidentes de la Junta Nacional de Seguridad de Transporte de Estados Unidos (National Transportation Safety Board), revelan que un 52% de los accidentes relacionados con el hardware de un avión fueron probablemente causados por un funcionamiento defectuoso del sistema. De estos accidentes, un 36% fueron causados por defectos en los componentes de los sistemas de propulsión [2].

Además, entre los objetivos más importantes de las operaciones de la comunidad de la aviación, está la mejora de la seguridad de los vuelos, la cual se intenta conseguir mediante la monitorización de los datos de vuelo. El uso de las técnicas de minería de datos para analizar estos datos está siendo cada vez más popular ya que muestra una gran eficacia y, sobre todo, un gran potencial [3]. Por tanto, por el gran potencial que muestra este tipo de técnica, sus múltiples usos en la actualidad, y su gran utilidad en el campo de la aviación, se ha decidido realizar este trabajo.

2. Objetivos concretos y relación con el estado actual

2.1.- OBJETIVOS DEL TRABAJO

El objetivo de este proyecto es realizar un modelo de detección de anomalías basado en el análisis de los datos que se han obtenido de la turbina y del compresor a lo largo de varios ciclos de vuelo. Estos datos son representados como series temporales multivaluadas, que se analizarán para encontrar puntos anómalos. Para realizar este trabajo se seguirán los siguientes pasos:

- Se realizará un preprocesamiento del conjunto de datos para que el modelo pueda usar una serie temporal de calidad y útil para la fase de extracción de conocimiento. Primero se hará una limpieza de los datos, después se aplicarán diferentes técnicas como PCA y escalado para reducir la dimensionalidad y preparar los datos correctamente [4].
- Una vez tengamos una serie temporal de calidad, se diseñarán 2 modelos diferentes para la detección de los outliers: Distancia de Mahalanobis y una red neuronal con autoencoders. Se entrenarán los modelos y realizarán su función correspondiente, que es identificar si cada ciclo ha sido atípico o no para cada avión.
- Después se utilizará un clasificador que decidirá, en base a los resultados obtenidos anteriormente, para cada avión si tiene el motor en un estado de salud anómalo o no.
- Finalmente se aplicarán 2 técnicas de validación que comprobarán si el modelo es eficaz y si realmente se clasifican los aviones correctamente.

El objetivo principal es, por tanto, diseñar un modelo que identifique correctamente los puntos atípicos durante un número determinado de periodos de funcionamiento de un motor a reacción y que los clasifique con una tasa de aciertos razonable.

Antes de abordar el problema directamente se hará un análisis previo para conocer los diferentes trabajos y propuestas que ya existen de este tema y su funcionamiento, así como los conceptos que se consideren necesarios para comprender este trabajo. A continuación, se desglosa la información bibliográfica que se ha utilizado para la investigación del tema.

2.2.- MONITORIZACIÓN DE MOTORES A REACCIÓN

Dado que en este proyecto trabajaremos con los datos de motores a reacción, se expondrá brevemente el funcionamiento del motor de un avión y sus partes correspondientes profundizando en las que se analizarán para realizar el diagnóstico.

Una de las piezas fundamentales de los aviones es el motor, que se considera un negocio a parte en el mundo de la aeronáutica ya que el consumo, rendimiento, y otros muchos más factores son claves para el funcionamiento del propio avión.

Los motores a reacción están presentes mayoritariamente en aviones de alta velocidad y altitud, misiles, cohetes y naves especiales. Su funcionamiento se basa en la propulsión a chorro, que consiste en impulsar hacia delante un objeto como reacción a la expulsión hacia atrás de una corriente de líquido o gas a gran velocidad. Existen varios tipos de motores para los aviones: motores de turbina, motores de pistón, de turbohélice, etc... Los que se analizarán son los motores turbofan, los más actuales. Entre las mejoras que supuso respecto a los tipos anteriores, está la utilización de la potencia obtenida para empujar el aire a mucha velocidad. El uso del ventilador ayuda a absorber más aire y por tanto mejorar el empuje del motor y ser más económicos al ahorrar combustible [5].

Los elementos básicos de un turboreactor son: conducto de admisión, ventilador, compresor de baja presión, compresor de alta presión, eje de alta presión, eje de baja presión, cámara de combustión, turbina de alta presión, turbina de baja presión y la tobera de escape (salida).

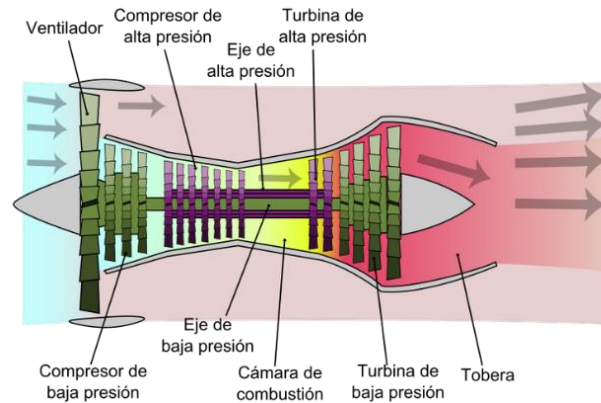


Ilustración 2.1.- Turborreactor y sus componentes principales [6]

Para que el motor acelere el aire, hay que aplicarle al aire varias transformaciones termodinámicas: El aire entra por el conducto de admisión y una parte pasa a través del ventilador, que es el responsable de producir la mayoría del empuje generado por el motor. Sigue hasta el compresor de baja presión, continúa hasta la cámara de combustión, donde se mezcla con carburante que ha sido inyectado previamente. Acto seguido comienza la combustión, y por este fenómeno se expanden los gases y mueven la turbina, donde el aire llega a alcanzar unos 1500°C [7] [8] .

Este complejo funcionamiento a lo largo del tiempo puede generar deterioro en los componentes del motor, y por tanto necesita un complejo y constante mantenimiento a lo largo de su vida útil como cualquier sistema mecánico. El deterioro es un fenómeno inevitable fruto de la exposición del motor a diversas condiciones operacionales y ambientales que resultan en erosión, corrosión, desgaste de las piezas, etc.....

A continuación, se muestra una figura donde se representan aproximadamente los diferentes picos y extremos de temperatura y presión que alcanzan las diferentes partes del motor.

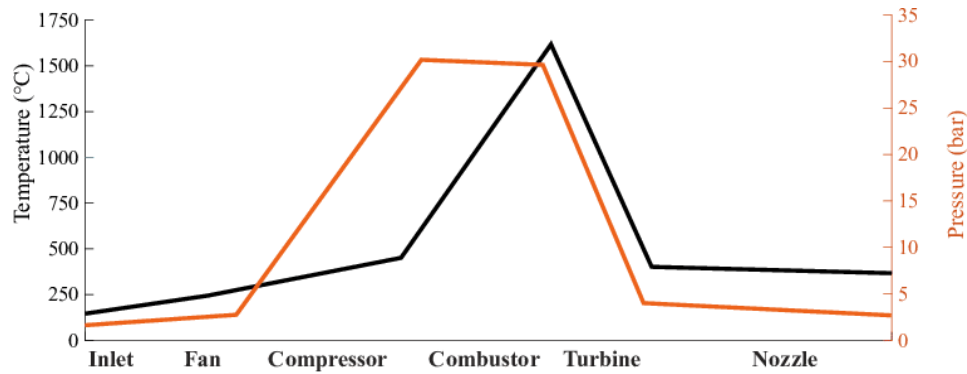


Ilustración 2.2.- Temperatura y presión de los componentes del motor a lo largo de su funcionamiento [6]

Como se ha visto en la ilustración, las diferencias de presión y temperatura entre los componentes del motor son muy notables, e implican unas condiciones de funcionamiento extremas. Estos picos de temperatura y presión son causantes de averías, entre otros, que pueden resultar en diferentes defectos físicos [9]:

- Ensuciamiento: La adherencia de diferentes contaminantes como cenizas, aceite, polvo, o químicos industriales en la superficie de los componentes puede producir irregularidades en su superficie.
- Erosión: Es la pérdida gradual de los materiales que forman la superficie de los componentes causado por la entrada de los contaminantes que causan el ensuciamiento.
- Corrosión: Es un deterioro irreversible causado por el contacto del óxido con contaminantes del aire y los gases de combustión

El ensuciamiento, erosión y corrosión, entre otros, causan una disminución de la eficiencia del compresor, de la turbina y de la capacidad de flujo [9].

Estos diferentes tipos de defectos físicos afectan a los parámetros de rendimiento. Estos parámetros son la capacidad de flujo, la eficiencia de compresor, la capacidad de flujo de la turbina, y la eficiencia de la turbina. Si estos parámetros se ven afectados, entonces los parámetros de medición también, que por lo general suelen ser la temperatura, presión, velocidad del eje y flujo de combustible. Estos parámetros son los que se desean monitorizar, ya que si muestran valores anormales indican, por tanto, una posible avería [9].

En general, los sistemas de monitorización que ya vienen incorporados en los aviones son capaces de detectar daños importantes en los motores, como vibraciones atípicas o picos bruscos de temperatura. Sin embargo, si el daño existente es superficial, los sistemas de monitorización actuales no serán capaces de identificarlo, lo que puede suponer un gran problema en la eficiencia que el motor debe compensar. Esta compensación afecta al compresor y la turbina, 2 componentes principales del motor que están muy relacionados ya que un mal funcionamiento del compresor disminuye su eficiencia termodinámica. Esto hace que se necesite más trabajo para comprimir la misma masa de aire, saliendo más caliente ya que la turbina al recibir menos aire se le tiene que inyectar más combustible y trabajar en otro punto de funcionamiento donde la temperatura es más alta. No obstante, estos elementos también se pueden deteriorar individualmente.

En la monitorización de un motor, existen dos partes importantes del procedimiento: diagnóstico y pronóstico:

El diagnóstico, en el campo del mantenimiento de componentes aeronáuticos, consiste en la detección, identificación y clasificación de una condición de funcionamiento anómala.

Por otra parte, el pronóstico se encarga de predecir posibles fallas del motor en el futuro en base al previo diagnóstico de anomalías, así como establecer la vida útil restante del motor (Remaining Useful Life, RUL) [2].

En el diagnóstico de fallas existen 3 términos que forman parte de este proceso [9]:

- Detección de la anomalía: En esta parte se detecta un comportamiento atípico
- Aislamiento de la anomalía: Se determina la localización y el tipo de anomalía.
- Identificación de la anomalía: Por último, se calcula o estima la magnitud de la falla.

A continuación, se muestra gráficamente cómo es el proceso GPA (Gas Path Analysis) de mantenimiento y diagnóstico de un motor:

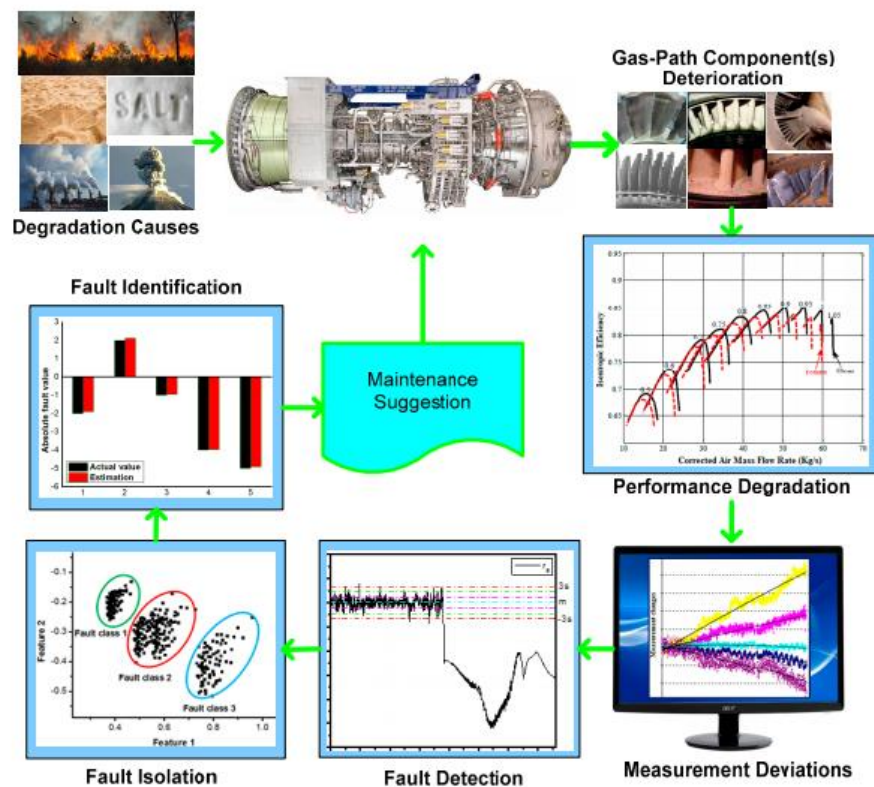


Ilustración 2.3.- Proceso de mantenimiento de un motor a reacción [9]

En la anterior ilustración se muestra el proceso de mantenimiento de un motor mediante Gas Path Analysis, un método que calcula la relación lineal entre los parámetros característicos del motor y las medidas de sus sensores introducido a finales de 1960 y que desde entonces ha ido evolucionando mediante distintas variantes. Permite monitorizar el rendimiento, además de detectar y cuantificar el deterioro del núcleo de motores [10].

Como se ha explicado anteriormente, es importante realizar una monitorización de la salud de los elementos del motor a tiempo real, que es el paso previo a la detección de anomalías que se desea realizar. Actualmente existen muchos estudios donde se emplean diferentes métodos para monitorizar el estado de salud de los motores de un avión. X. Zhang et al. [11] diseñaron un modelo muy robusto de diagnóstico, aislamiento y predicción de averías con un estimador adaptativo no lineal, que cuenta con técnicas de aproximación y umbrales adaptativos que hacen que se mantenga un rendimiento de diagnóstico preciso mientras que el motor se va degradando a lo largo de toda su vida útil.

2.3.- DETECCIÓN DE ANOMALÍAS

Una vez analizado el funcionamiento de los elementos y componentes a monitorizar, se procede a estudiar las técnicas y algoritmos para detectar comportamientos anómalos.

2.3.1.- Conceptos y tipos de anomalías

Antes de analizar las distintas técnicas que existen de detección de anomalías que se usarán como base para la elaboración de este proyecto, se definirán algunos conceptos de este tema, empezando por los distintos tipos de anomalías [12]:

- Anomalías puntuales: Son los valores atípicos o outliers que se han definido previamente, puntos que al representarlos visualmente están lejos del resto.
- Anomalías contextuales: La anomalía es específica dependiendo del contexto, es común en series temporales. Esto quiere decir que aparecen valores atípicos en momentos específicos de la serie, por ejemplo: menos lluvia en los meses de verano.
- Anomalías colectivas: Un conjunto de varios puntos puede ser detectado como anomalía, pero cada punto individualmente no podría identificarse como valor atípico. Solo gracias al conjunto de todos los puntos se puede detectar una anomalía.

A continuación, se definirán los diferentes tipos de detecciones [12]:

- Detección de anomalías supervisada: Se cuenta con un conjunto de entrenamiento con valores normales y atípicos previamente etiquetados con el que se pueden emplear diferentes clasificadores para problemas no balanceados.
- Detección de anomalías no supervisada: En este caso los valores del conjunto de datos no están etiquetados, por tanto, se asocia una puntuación.
- Detección de anomalías semi-supervisada: No hay información de los outliers, solo de los datos normales. En este tipo de detección se asigna un nivel de anomalía a cada punto usando la información de las etiquetas de los puntos normales.

La detección de anomalías en la actualidad es una técnica que se emplea en múltiples ámbitos. Sus usos más interesantes son los siguientes [12]:

- Detecciones de fraude: Los fraudes con tarjeta de crédito en la actualidad suceden todos los días y de diversas maneras. No siempre es posible prevenirlos, pero sí identificar a los culpables y actuar a tiempo para minimizar los daños de manera que el perfil de una persona que hace una compra con una tarjeta de crédito robada es diferente del dueño original, por eso las compañías buscan patrones que caractericen al ladrón por cambios en el comportamiento típico.
- Detección de intrusiones: Se pueden detectar intrusiones no deseadas a servidores o computadores monitorizando comportamientos no frecuentes
 - en las comunicaciones de los sistemas.
- Medicina: Se pueden encontrar síntomas o resultados poco frecuentes en tests y que se pueden corresponder con enfermedades no diagnosticadas.
- Perturbaciones en ecosistemas: Diferentes fenómenos naturales como inundaciones, huracanes o incendios se pueden predecir, a veces, detectando patrones atípicos en el clima o en las temperaturas.

Las anomalías pueden ser causadas por distintos factores, entre los cuales podemos destacar 3 [12]:

- Datos de diferentes clases: Un objeto puede ser diferente de los demás por ser de una clase distinta. Se considera un objeto anómalo ya que es sospechoso de haber sido creado por un mecanismo diferente que a los demás. Las anomalías de las intrusiones y fraudes son causadas por este factor.
- Variación natural: Muchos datos pueden modelarse estadísticamente como en la distribución normal donde la probabilidad de un objeto disminuye cuando sus propiedades son diferentes a la de la media.
- Errores de medición: Puede haber datos anómalos por culpa de una mala medición por un error humano. Estos outliers hay que eliminarlos durante el preprocesamiento.

2.3.2.- Técnicas tradicionales

Las técnicas estadísticas son de las más comunes para la detección de anomalías. Están basadas en modelos donde cada objeto se examina dependiendo de cómo encaja en el modelo.

En estos modelos, un outlier se define como un objeto con una probabilidad muy baja de aparición. En muchas ocasiones, los valores de las propiedades de los datos siguen una distribución normal, por lo que si se calcula la media y la varianza de la distribución se puede obtener la probabilidad de que un objeto aparezca [12]. A continuación, una figura donde se representa la distribución normal de un conjunto de datos y su clasificación por la frecuencia de aparición de los objetos:

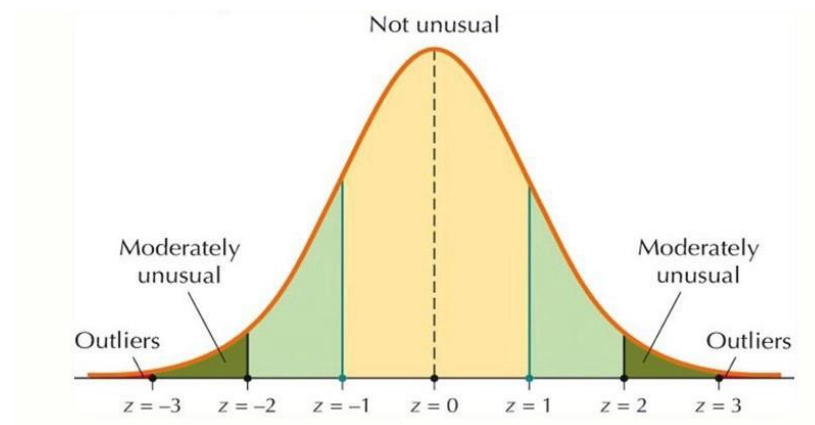


Ilustración 2.4.- Distribución normal [13]

Se observa que los outliers están en los extremos de la gráfica, siendo los que menos probabilidad tienen de aparecer, mientras que los puntos más normales se encuentran en el centro.

Las técnicas tradicionales, como la estadística, presentan diversos problemas o retos para la localización de anomalías en la actualidad que son incapaces de resolver completamente. Los principales son los siguientes:

1. Tasa baja de recuperación de detección de anomalías: Los puntos anómalos son difíciles de identificar, pues su aparición es poco frecuente y suelen ser muy heterogéneos. Los métodos de detección no supervisados son los que más falsos positivos generan en conjuntos de datos reales, por tanto, reducir este tipo de anomalías identificadas erróneamente es muy importante [14].

2. Detección de anomalías en conjuntos de datos con una dimensionalidad alta o datos no independientes: En general, cuando los datos están representados en espacios con una baja dimensión, las anomalías son evidentes y fáciles de identificar, pero cuando se trata de espacios con dimensionalidad más alta, los puntos anómalos pasan más desapercibidos. Por eso, actualmente es un gran reto diseñar modelos que detecten correctamente anomalías en este tipo de espacios, pues, aunque es posible realizar la detección en espacios más reducidos, hay complejas interacciones de características que pueden ser esenciales en datos con altas dimensiones. Además, también es un desafío a día de hoy detectar anomalías en distribuciones con variables que son dependientes [14].

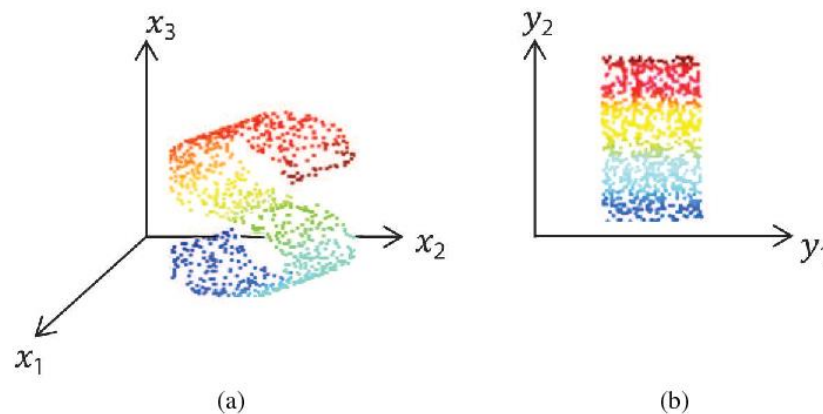


Ilustración 2.5.- Representación de un espacio de tres dimensiones (a) y un espacio bidimensional (b) [15]

3. Eficiencia de aprendizaje: Uno de los grandes retos es diseñar un modelo que sea capaz de aprender a representar expresivamente la anormalidad y la normalidad de un conjunto de datos con pocas anomalías etiquetadas (detección poco supervisada). También es importante dar con una técnica que sea generalizada ante posibles nuevas anomalías que no están dentro de los puntos atípicos previamente etiquetados [14].
4. Detección de anomalías resistentes al ruido: Existen muchos métodos que asumen que el conjunto de datos etiquetado está limpio, sobre todo los semi supervisados, haciendo que esta técnica sea vulnerable ante puntos ruidosos que son erróneamente etiquetados con la clase contraria, además existen conjuntos de datos con muchas anomalías contaminadas con ruido [14].

5. Detección de anomalías complejas: La mayoría de las técnicas existentes para la detección de anomalías están pensadas para simplemente descubrirlas, pero no se pueden usar para anomalías contextuales o colectivas ya que sus comportamientos son mucho más diferentes que el de las anomalías puntuales. Esto se considera un gran reto, pues las técnicas actuales no disponen de esta característica, solo se centran en localizar anomalías de una sola fuente de datos, mientras que muchas aplicaciones necesitan la detección de anomalías de diferentes fuentes, como por ejemplo datos multidimensionales [14].
6. Explicación de las anomalías: Un gran riesgo en la detección de anomalías es la identificación de puntos anómalos que pueden dar lugar a una representación sesgada por el algoritmo de los grupos minoritarios de datos. Por tanto, es de gran importancia conocer por qué los puntos atípicos han sido identificados como anomalías [14].

Las soluciones a estos retos más comunes son basadas en deep learning con redes neuronales, que abordan la mayoría de los retos existentes en la actualidad mencionados previamente, mientras que los métodos tradicionales no. Trabajan mucho mejor con datos heterogéneos y el aprendizaje de relaciones complejas que los modelos clásicos. Además, cuentan con una optimización de extremo a extremo y aprenden una representación a medida específica para los datos, mientras que las técnicas tradicionales no [14].

Método	Optimización de extremo a extremo	Aprendizaje de una representación a medida	Aprendizaje de relaciones complejas	Datos heterogéneos
Tradicional	No	No	Bajo	Bajo
Deep Learning	Sí	Sí	Alto	Alto
Retos	1-6	1-6	1, 2, 3, 5	3,5

Tabla 2.1.- Comparación de las técnicas tradicionales con las de Deep Learning [14]

Se puede apreciar en la tabla anterior que las técnicas de Deep Learning cubren muchos más problemas, por esta razón nos centraremos en las técnicas de este tipo y llevaremos a cabo la detección de anomalías de los motores con ayuda de este tipo de métodos.

2.3.3.- Técnicas de Deep Learning

Se pueden clasificar en tres categorías principales [14]:

- Deep Learning para extracción de características
- Aprendizaje de representaciones de comportamiento normal
 - Genérico
 - Autoencoders
 - Aprendizaje de características dependientes de medidas
 - Basadas en distancia
 - Basadas en proximidad (K-Neighbors)
 - Distancia de Mahalanobis
 - Basadas en clusters
- Aprendizaje de puntuación de anomalías de extremo a extremo

A continuación, una ilustración con la clasificación más detallada de estas técnicas y algunos ejemplos de otros métodos existentes.

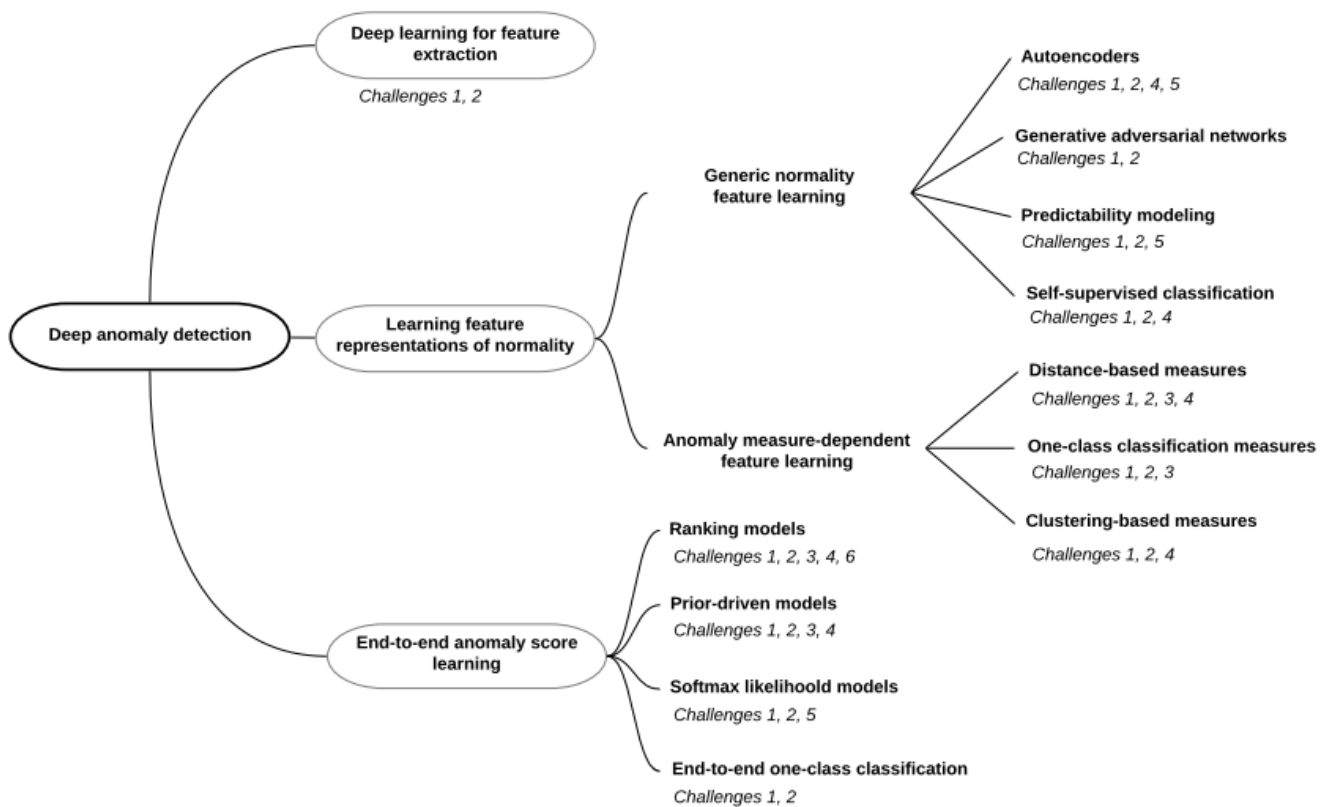


Ilustración 2.6.- Clasificación de las técnicas de Deep Learning de detección de anomalías y los retos (*challenges*) que aborda cada uno [14]

El problema del diagnóstico de motores de este proyecto aborda los retos 1,2,3,4 y 5:

Identificar las anomalías es complejo (*challenge* 1) al tener una tasa de aparición baja y poder aparecer falsos positivos, pues al recurrir a métodos semi-supervisados donde las anomalías no están etiquetadas previamente (solo los datos normales lo están), y al usar un conjunto de datos reales el problema se complica mucho más. La alta dimensionalidad (*challenge* 2) es también una dificultad en este diagnóstico, ya que los conjuntos de datos cuentan con múltiples variables, por lo tanto, facilitaría la detección si se trabajara en dimensiones más bajas o con técnicas que puedan soportar las altas dimensiones del problema. El reto 3, aprendizaje de la anormalidad/normalidad, está presente en este trabajo pues al emplear detección poco supervisada representar las anomalías es difícil.

Se aborda también el *challenge* número 5, detección de anomalías complejas, ya que, al trabajar con series temporales en un motor, muchas de las anomalías no serán puntuales sino contextuales pues el deterioro aumenta con el paso del tiempo. No muchas técnicas

consiguen identificar otro tipo de anomalías que no sean puntuales. Por último, en este proyecto aparece el reto 4, pues al emplear métodos poco supervisados se asume que los datos están limpios, sin ruido, y además se cuenta con una gran cantidad de datos.

Se focalizará el estudio de las diferentes técnicas de deep learning en Aprendizaje de representaciones de comportamiento normal:

- Aprendizaje de características dependientes de medidas
 - Basadas en proximidad: Un punto es atípico si es distante de los demás [8]. Para ello, generalmente se calcula la distancia al k-ésimo vecino más cercano (k-nearest neighbor), pero también se usan métodos como la distancia relativa u otras variaciones de k-neighbor. Este tipo de técnicas son sencillas y directas, pero su mayor limitación está en los datasets de gran tamaño debido a problemas de dimensionalidad. Sin embargo, la mayoría de estas técnicas ya cuentan con realizar una previa reducción de dimensionalidad para proyectar los datos en un espacio de dimensión reducida, por tanto, no sería un gran problema [14]. Uno de los métodos basados en el k-ésimo vecino más cercano es el de Ramaswamy et. al [16], donde calcula la distancia para cada punto del conjunto de datos al kº vecino más cercano y hace una clasificación donde los N puntos con mayor distancia son clasificados como anomalías.

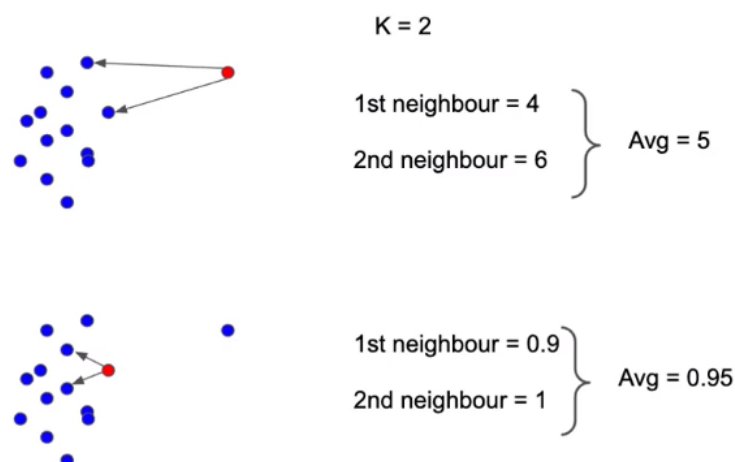


Ilustración 2.7.- K vecinos más cercanos (K Nearest Neighbors, KNN) [17]

La anterior figura muestra un ejemplo en el que se da el valor 2 al parámetro K, es decir se medirán las distancias de los puntos a sus dos vecinos más cercanos y se calculará la distancia media. El punto más anómalo es el primero ya que es el que a más distancia está de sus vecinos. Hay que tener en cuenta que el parámetro K es de gran importancia e interesa escoger un valor óptimo para el mismo ya que existen casos como el siguiente en los que la detección de outliers no es tan efectiva:

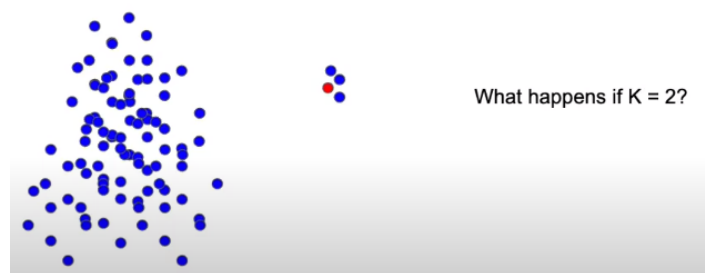


Ilustración 2.8.- Importancia del valor K [17]

Esta técnica, está dentro de los métodos basados en distancias, los cuales abordan varios de los problemas mencionados anteriormente. Es fácil trabajar con una representación con una dimensionalidad baja adaptada a las medidas que se emplearán para medir las distancias entre los puntos. También existen técnicas como las de Hu Wang et al. [18] que son robustas a la contaminación de las anomalías por ruido y trabajan correctamente con métodos no supervisados [14].

- Basadas en densidad: Los objetos anómalos se encuentran en zonas de baja densidad. Uno de los métodos más comunes es calcular el Local Outlier Factor, una puntuación que mide el grado de atipicidad de un punto basado en el cálculo de la densidad local a partir de la distancia a los k puntos más cercanos. Es decir, también se usa la técnica de k-vecinos. Si el LOF de un elemento es mayor que 1, entonces ese punto es anómalo. En cambio, si es próximo a 1, el elemento es considerado como normal. La densidad local (Local Reachability Density, LRD) se calcula con la inversa de la media de las distancias de accesibilidad de un punto desde sus k elementos más cercanos. Cuanto más baja sea esta medida, más distancia hay hasta el siguiente cluster. El cálculo del LOF se lleva a cabo mediante la división de todas las

densidades locales de accesibilidad de los k vecinos del punto con el LRD del punto en cuestión [12].

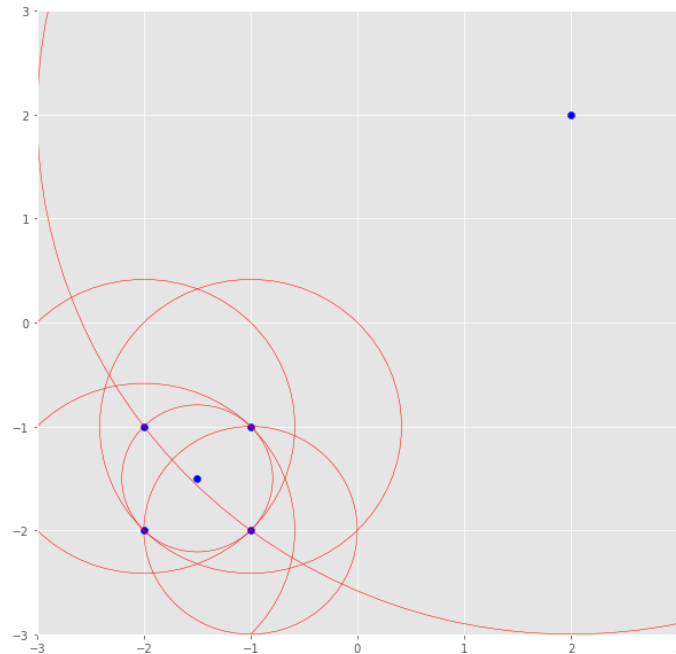


Ilustración 2.9.- Ejemplo de las distancias de la técnica Local Outlier Factor [12]

En la ilustración 9, la LRD del punto de la esquina superior derecha es la media de todas las distancias de sus vecinos más cercanos (puntos en la esquina inferior izquierda), que a su vez tienen otra LRD diferente ya que no cuentan con el punto que estamos evaluando.

- Basadas en clusters: Un cluster es un conjunto de datos que han sido agrupados mediante clustering por tener propiedades y/o características similares. Esta técnica busca grupos de objetos que están altamente relacionados, de manera que los puntos anómalos pertenecen a un cluster con pocos miembros y que esté alejado de otros clusters. Otra manera de usar esta técnica es agrupar todos los objetos en un cluster y después medir el grado de pertenencia al cluster de cada punto. Para calcular este valor, se mide la distancia del punto al centro del cluster sin contar con la densidad del mismo. Sin embargo, si los clusters tuvieran forma de óvalo, se podría emplear otro tipo de distancias, o un método de menos carga computacional, como por ejemplo la relación entre la distancia del punto al centro y la media de distancias de todos los puntos del cluster a su centro [19].

Este método, aborda algunos de los retos que se explicaron anteriormente, en concreto la baja tasa de aparición de anomalías y el problema de los espacios con alta dimensionalidad, pues es capaz de realizar correctamente la detección en un espacio aprendido de baja dimensión de las entradas que preserva la suficiente información para diferenciar los puntos anómalos, incluso mejor que en el espacio original. Si esta técnica se combina con los autoencoders también abordaría el problema de la contaminación por ruido [14].

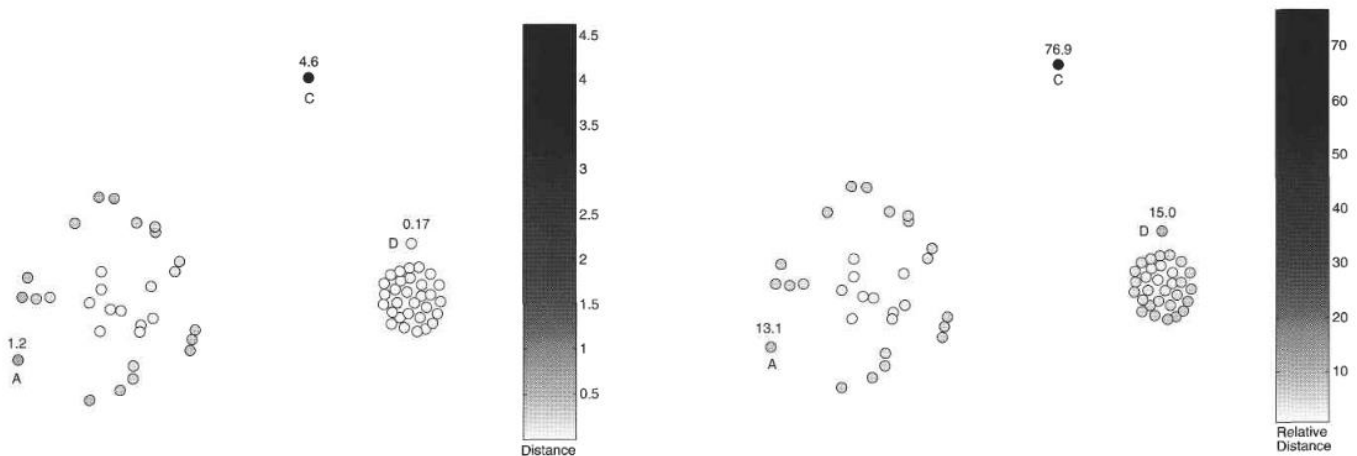


Ilustración 2.10.- Detección de puntos anómalos con clusters

La imagen izquierda muestra la métrica que calcula el grado de pertenencia del punto C al cluster sin contar su densidad. En la imagen derecha se emplea la distancia relativa, que sí tiene en cuenta la densidad del clúster.

- Distancia de Mahalanobis: Es una técnica en la que se emplea la medida de distancia de Mahalanobis, donde dependiendo del valor de esta distancia de cada punto será identificado como anómalo o normal, en función de la covarianza de todas las variables. Se escogerá este método para la resolución del problema de los aviones de este trabajo porque, al emplear una media y varianza de toda la distribución, resuelve problemas de escala y correlación, además de que es apropiado para grandes

volúmenes de datos. Se detallará más en profundidad esta técnica en la siguiente sección.

- Genéricas
 - Autoencoders: Los autoencoders aprenden una representación de las características de los datos en una dimensionalidad reducida para ser después reconstruida correctamente.

Los autoencoders, abordan varios problemas actuales en la detección de anomalías, en concreto los *challenges* 1,2,4 y 5 explicados anteriormente. Uno de ellos, es detectar puntos anómalos en datos con altas dimensiones. En este caso los autoencoders, al disponer de ilimitadas configuraciones de capas y arquitecturas, permite detectar anomalías en espacios de grandes dimensiones y con datos no independientes entre sí. Además, si los autoencoders han aprendido representaciones de los datos más expresivas, esta técnica también es capaz de reducir falsos positivos, cosa que otras técnicas más clásicas no pueden hacer. Este modelo, al ser sensible al ruido, es capaz de ser entrenado para recordarlo y llevar a cabo un sobreajuste mayor y dar con errores de reconstrucción más bajos [14].

Se han escogido los autoencoders para el diagnóstico de los motores a reacción de este trabajo por abordar la mayoría de los retos presentes en el problema de este proyecto, donde es muy destacable que permitan trabajar cómodamente con grandes volúmenes de datos. Además, porque una de sus grandes ventajas es que son muy genéricos y permiten trabajar fácilmente con diferentes tipos de datos, además de que hay muchas variantes y configuraciones que dan lugar a un amplio rango de mejora [14]. Se explicará su funcionamiento detalladamente en las siguientes secciones.

Un ejemplo de modelo de detección de anomalías actual es el creado por Mei Yuan et al. [20] que utiliza las redes Long Short-Term Memory (LSTM), que son un tipo especial de redes recurrentes que, a diferencia de las normales, pueden aprender dependencias largas, lo que significa que cuentan con una “memoria” a más largo plazo. Esta red es capaz de predecir la vida útil que le queda a un motor y la probabilidad de que éste falle en caso de estar en unas condiciones de operación muy complejas. Para ello, etiqueta los primeros ciclos

del motor como periodos normales, los últimos como atípicos para el set de entrenamiento, y entrena al modelo para que, en base a este set, identifique anomalías en el resto del conjunto. Además, se usa el detector de anomalías con SVM para clasificar el tipo de fallo de cada secuencia [21].

2.3.3.1.- Distancia de Mahalanobis

Una de las técnicas basadas de distancia para la detección de anomalías que se usarán en este proyecto es el uso de la distancia de Mahalanobis, una medida de distancia introducida en 1936 por Prasanta Chandra Mahalanobis, un científico indú destacado por sus estudios de la estadística aplicada. Se utiliza para comprobar la similitud de dos variables aleatorias multidimensionales [22].

Para comprender y emplear correctamente esta medida se recurren a varias definiciones:

Variable aleatoria: Es una variable que cuantifica los resultados de un experimento aleatorio que puede tomar cualquier valor. Por ejemplo, los valores que podemos obtener tras tirar un dado 3 veces (3,1,5).

Matemáticamente es una aplicación $X : \Omega \rightarrow \mathbb{R}$, es decir, para todo suceso en el espacio Ω da un valor numérico real [22].

Se pueden clasificar las variables aleatorias en dos tipos [22]:

- Variable aleatoria discreta: Variable que toma valores numéricos aislados, es decir no puede haber ningún valor intermedio entre 2 consecutivos.
- Variable aleatoria continua: Es una variable que puede tomar cualquier valor dentro de un intervalo.

Distancia: Es la longitud del camino más corto entre dos elementos.

Distancia euclídea: Distancia entre dos puntos A y B a la longitud de la recta que tiene por extremos estos puntos. Su expresión matemática es [22]:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Ecuación 2.1.- Distancia euclídea

La distancia de Mahalanobis y la Euclídea son iguales si las variables no están correlacionadas, es decir, si sus valores cambian sistemáticamente con respecto a los valores semejantes de la otra. Sin embargo, si están correlacionadas, que es lo más frecuente en problemas reales, la medida Euclídea no lo tiene en cuenta. La distancia Euclídea entre un punto y el centro de toda la distribución puede llegar a dar información errónea sobre la cercanía del punto al grupo de puntos [23].

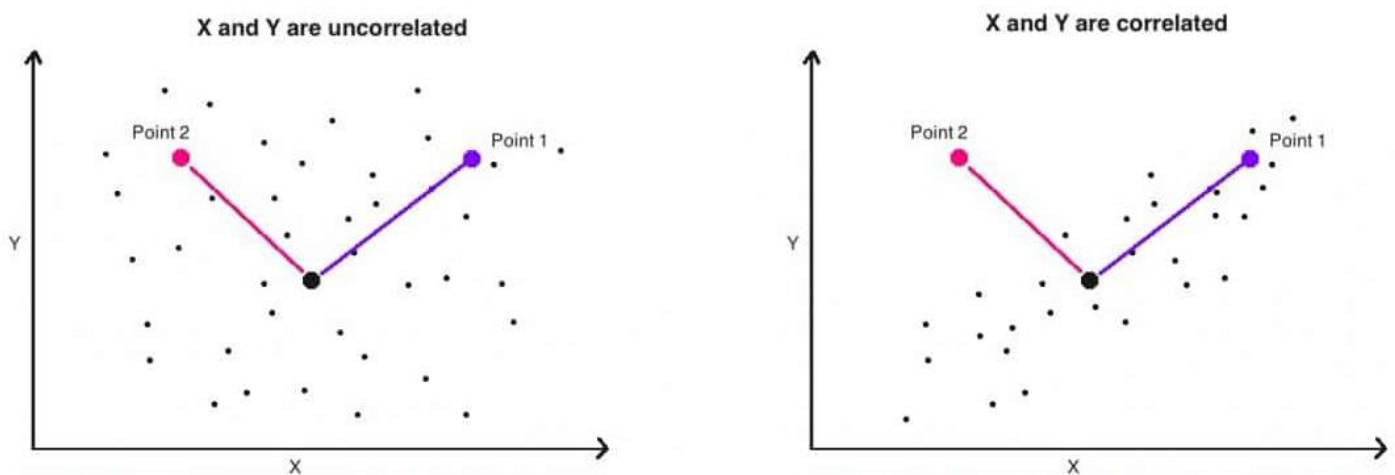


Ilustración 2.10.- Diferencia entre distribuciones con variables con y sin correlación [24]

En la anterior ilustración, se observa que en el primer gráfico no hay correlación entre las variables y la distancia Euclídea puede ser útil para identificar si un punto es parte del grupo. En el segundo gráfico, la distancia es la misma, pero se puede apreciar que el punto 2 no es parte del grupo de puntos [24]. Es por este problema que se quiere recurrir a una distancia más robusta que indique cuanto de distante está un punto de una distribución.

La distancia de Mahalanobis, por tanto, se define como:

$$D(X, \mu) = \sqrt{(X - \mu)^T * C^{-1} * (X - \mu)}$$

Ecuación 2.2.- Distancia de Mahalanobis

Donde X es un vector aleatorio de la observación, μ es el vector de los valores medios de la distribución, y C^{-1} es la matriz de covarianzas. T indica que el vector es traspuesto. La primera parte de la ecuación, $(X - \mu)^T * C^{-1}$, representa la distancia que hay del vector a la

media, dividida por la matriz de covarianza, ya que $C^{-1} = 1/C$. Esto es equivalente a la estandarización ($z = (X - \mu) / \Sigma$). Cuanto más correlacionadas estén las variables, mayor será la covarianza. Por tanto, al dividir la distancia a la media por la covarianza, obtenemos poca distancia si las variables están correlacionadas, y mucha si no lo están.

Como es explico anteriormente, la matriz C contiene la covarianza entre los elementos del vector de la observación X. La representación de C sería:

$$C = \begin{bmatrix} Var(X_1) & Cov(X_1, X_2) & \dots & Cov(X_1, X_n) \\ Cov(X_2, X_1) & Var(X_2) & \dots & Cov(X_2, X_n) \\ \dots & \dots & \dots & \dots \\ Cov(X_n, X_1) & Cov(X_n, X_2) & \dots & Var(X_n) \end{bmatrix}$$

Ecuación 2.3.- Matriz de covarianza

Nótese que en la diagonal se encuentra la varianza de cada elemento ya que la covarianza de una variable consigo misma es su propia varianza. Lo que hace la distancia de Mahalanobis es:

1. Traslada los datos para que la nueva media sea el origen de coordenadas
2. Rota los datos de forma que las correlaciones entre las variables sean 0.
3. Estandariza el resultado del paso anterior para que las varianzas de todas las variables sean iguales a 1.
4. Calcula la distancia euclídea al origen de los puntos resultantes [24].

Para comprender este proceso, partimos de esta distribución de ejemplo:

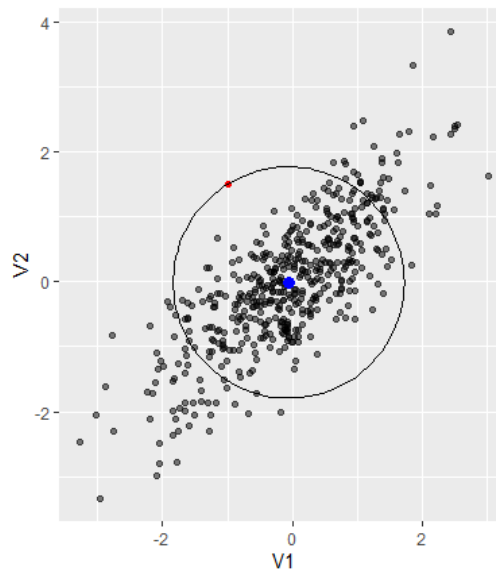


Ilustración 2.11.- Representación gráfica de una distribución de variables aleatorias [25]

El punto azul es la media del conjunto de datos, y el punto rojo es un punto de la distribución que a simple vista se puede considerar atípico, pero si se aplica la distancia euclídea habitual no tendremos en cuenta la relación entre las variables, muchos puntos que parecen más típicos están fuera del círculo [25]. A continuación, una representación gráfica de los 3 primeros pasos:

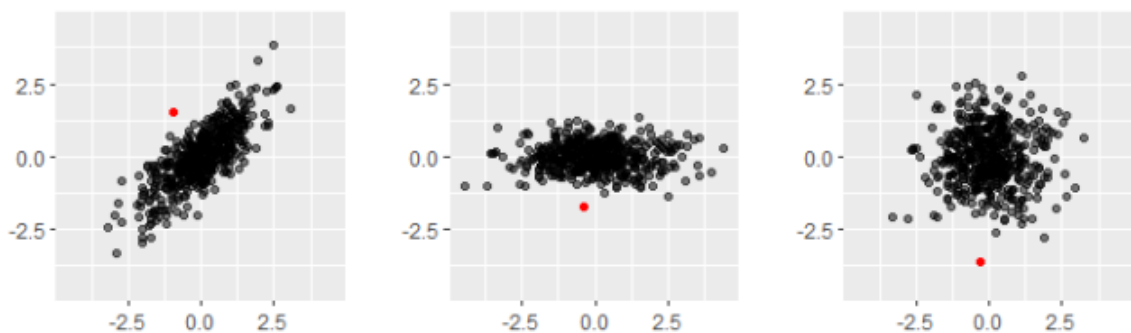


Ilustración 2.12.- Distancia de Mahalanobis en una distribución [25]

Ahora se puede comprobar que el punto rojo es uno de los más alejados de la media.

2.3.3.2.- Autoencoders

El segundo método para detectar anomalías que se usará son las redes neuronales de Autoencoders.

Antes de empezar con el desarrollo de este método, es conveniente comprender que es una red neuronal. Se podría decir que una red neuronal, es un modelo simple inspirado en el funcionamiento del sistema nervioso de un ser humano, pues está formada por varias neuronas artificiales conectadas que emiten señales desde la entrada hasta la salida. Se componen de diferentes capas formadas por las neuronas: la capa de entrada, una o más capas ocultas, y la capa de salida [12]:

- Capa de entrada: Está formada por neuronas o nodos que reciben los datos del entorno.
- Capa oculta: Proporciona a la red neuronal una cierta libertad para representar mejor las características del entorno que se quieran modelar.
- Capa de salida: Está compuesta por nodos responsables de proporcionar el resultado final.

Las neuronas están conectadas unas a otras unidireccionalmente y tienen asociado un peso y un umbral, con la condición de que, si la salida de cualquiera de las neuronas es mayor del umbral especificado, esa neurona se activa y manda a la siguiente capa la información [26], es decir, cada nodo tiene asignado una función de activación (entrada, peso, umbral).

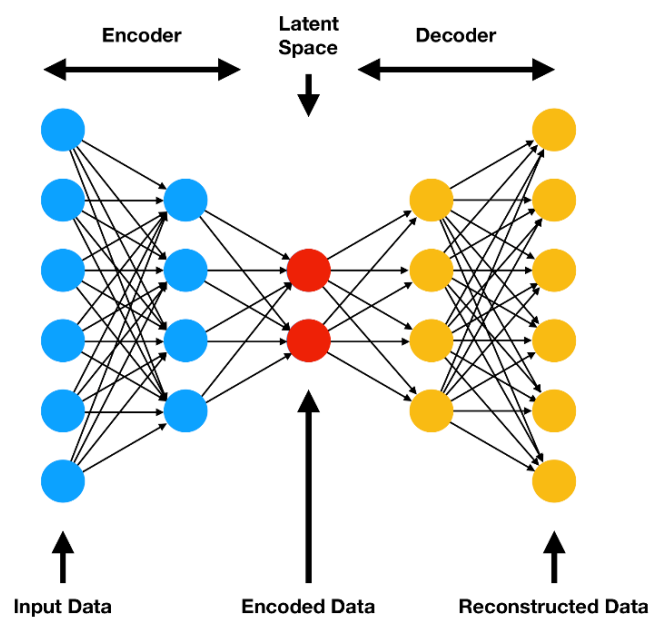


Ilustración 2.13.- Representación gráfica de un autoencoder [27]

Un autoencoder es una red neuronal no supervisada que copia sus entradas en sus salidas. Por tanto, su objetivo es, a parte de aprender una representación (encoding) del conjunto de datos para una reducción de la dimensionalidad, reconstruir lo mejor posible la entrada original [12] [28].

Su funcionamiento se basa en comprimir la entrada en un espacio de estados latente en una representación desde donde se reconstruye la salida [12]. Lo interesante de este funcionamiento está en las capas ocultas de la red, pues si el número neuronas de estas capas es menor que las de entrada, extraerán solo la información esencial y por tanto se “obliga” a la red neuronal a ignorar los datos que no son relevantes (ruido) y aprender los patrones más frecuentes de los datos. Es por esto por lo que la dimensión de la entrada es mayor que la de las capas ocultas, pues se quiere seleccionar solo las características esenciales del conjunto de datos y sino, se daría demasiada capacidad a los nodos para el aprendizaje de los datos [19]. Sin embargo, existen algunas aplicaciones donde es conveniente que las capas ocultas tengan más dimensión que la de entrada, para capturar la riqueza de los datos de la distribución entrante, donde se aplican técnicas en las que se añade ruido a la entrada para anular una parte de los valores.

Los autoencoders están formados por:

- Encoder: se comprime la entrada en una representación latente, es decir, en la capa oculta, podemos expresarlo como $h = f(x)$, donde h es la representación resultante, x la entrada, y f la función que se aplica para llevar a cabo esta representación [12].
- Decoder: Se descomprime o reconstruye la representación en la salida $r = g(h)$.
- Representación latente: Contiene la representación comprimida de los datos de entrada, en la dimensionalidad más reducida posible.

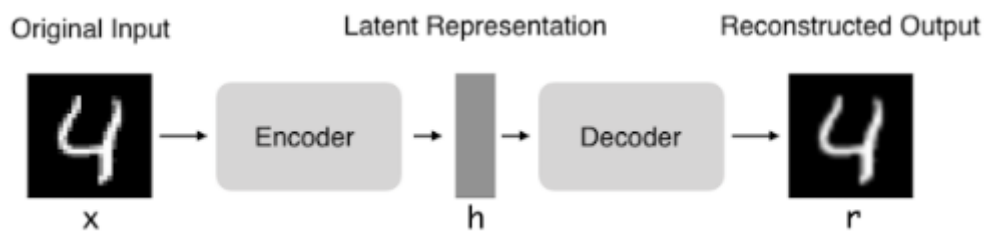


Ilustración 2.14.- Funcionamiento de una red de autoencoders [12]

Por tanto, podríamos definir la función del autoencoder como $g(f(x)) = r$.

Una vez que se obtiene el resultado en la salida, se compara con la entrada original y se computa la diferencia para calcular la pérdida y poder minimizar este valor, para poder llegar a una reconstrucción lo más cercana posible a la original. La función que mide este error o pérdida se llama función de pérdida. El proceso de aprendizaje de los autoencoders, se resume básicamente en minimizar esta función de pérdida:

$$L(x, g(f(x)))$$

Ecuación 2.4.- Función de pérdida

Donde L es una función de pérdida que penaliza $g(f(x))$ (r) el nivel de desigualdad con x [29]. Cuanto menor sea, mejor se ha entrenado y diseñado el modelo y por tanto mejores resultados dará.

Cuando se usa esta función y el decoder es lineal, el autoencoder abarca el mismo subespacio que la técnica de PCA (Principal Components Analysis), pues ha sido entrenado para aprender el subespacio del conjunto de datos. Por otra parte, si el encoder y el decoder usan funciones no lineales, entonces aprenden tareas más útiles y robustas que PCA.

La función objetivo formalmente sería la siguiente:

$$\text{Objetivo} = \min \sum_{x \in X} d(x, g(f(x)))$$

Ecuación 2.5.- Función objetivo

Donde d es una métrica de distancia, que dependiendo del modelo y su contexto se puede escoger una u otra. Una de las más comunes es el error cuadrático:

$$L(x, r) = \frac{1}{n} \sum_{i=1}^n (x - g(f(x)))^2$$

Ecuación 2.6.- Error cuadrático

El entrenamiento de los autoencoders permite generar la mejor reconstrucción posible de los elementos de entrada mediante una configuración óptima de los nodos de la red neuronal. La función objetivo puede obligar al autoencoder a obtener una representación de los datos con unas características concretas, dando lugar a diferentes tipos de autoencoders. Los más frecuentes son [12]:

- Autoencoder simple: Cuenta con 3 capas, capa de entrada, oculta y de salida. Dependiendo de la dimensión de la capa oculta, la compresión será diferente.
- Autoencoder multicapa: En este tipo se puede añadir un número aleatorio de capas, y cualquiera de ellas se puede usar como representación latente, suelen ser redes simétricas.
- Autoencoder convolucional: Se pueden usar capas convolucionales para los encoders y decoders para forzar un nivel de compresión más alto. Las capas convolucionales extraen las características más relevantes y tienen filtros como identificadores de las características.
- Autoencoder regularizado: Este tipo limita la dimensionalidad del espacio regularizando la red, mediante representaciones dispersas donde se busca que el peso de los enlaces sea cero, y representaciones para eliminación de ruido, donde se añade ruido a la entrada para forzar a la red a aprender la red sin ruido y conseguir una representación robusta de los datos [12].
- Autoencoders variacionales: Pueden generar nuevas representaciones mediante una aproximación variacional del aprendizaje de la representación latente.

En este caso, se usarán los autoencoders para la detección de anomalías en este proyecto, pues esta técnica puede realizar transformaciones con funciones no lineales y varias capas, mientras que otras técnicas como PCA solo es capaz de usar álgebra lineal y entrenar una sola capa. Por tanto, es más eficiente entrenar varias capas, sobre todo con conjuntos de datos de gran tamaño [19] donde puede ser muy útil representar los datos en una dimensión reducida y codificada en el espacio de entrada original.

Linear vs nonlinear dimensionality reduction

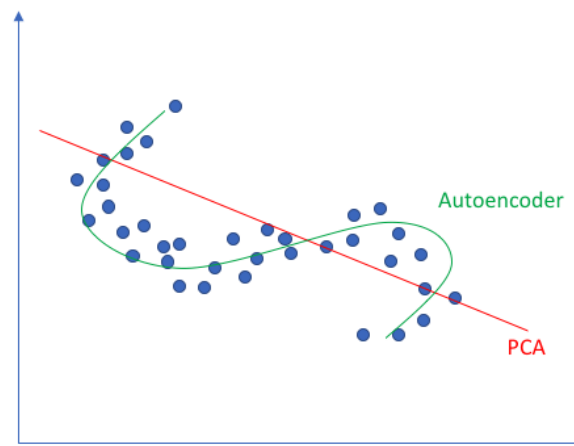


Ilustración 2.15.- Reducción lineal (PCA) y no lineal (Autoencoders) de la dimensionalidad [30]

Para detectar las anomalías, los autoencoders realizan lo siguiente: Primero, se entrena la red introduciendo en la entrada de los autoencoders ciclos que son considerados como normales para que éstos aprendan la representación latente de los datos normales entrantes. Después, el encoder reconstruye los datos originales usando la salida de las capas ocultas, como se explicó anteriormente. Si la entrada es atípica, la red no va a reconstruir ese dato correctamente, pues ha sido entrenada con datos típicos, diferentes a los atípicos, y, por tanto, la salida será diferente a la entrada y se reflejará en el resultado la función de pérdida [31]. Para facilitar la detección de las anomalías, se establece un umbral con el que comparar la salida de la función de pérdida, identificando como valores anómalos los que obtengan como resultado de la función de pérdida un valor mayor que el del umbral establecido.

2.4.- ANÁLISIS DE COMPONENTES PRINCIPALES (PCA)

Una de las técnicas que se usarán para el diagnóstico de motores será el análisis de componentes principales, una técnica añadida a la distancia de Mahalanobis y a los autoencoders, con el fin de describir el nuevo conjunto de datos en términos de nuevas variables no relacionadas. PCA permite simplificar la complejidad de un espacio que cuenta con muchas dimensiones sin perder información, es decir, “condensa” la información

aportada por las variables en pocas componentes. Esto es útil para manejar datos con diversas técnicas de regresión, clustering, técnicas basadas en proximidad, etc... [32].

El cálculo de las componentes principales se obtiene por combinación lineal normalizada de las variables iniciales. La primera componente se calcularía de la siguiente manera:

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \dots + \phi_{n1} X_n$$

Donde x_1, x_2, \dots, x_n es el grupo de variables originales, y $\phi_{11}, \phi_{21}, \dots$ son los pesos o importancia de cada variable para saber qué tipo de información tiene cada variable.

A partir de Z_1 se van calculando el resto de las componentes principales que se deseen obtener, con la condición de que Z_1 y Z_2 no pueden estar correlacionadas, es decir tienen que ser perpendiculares [32]. En este caso, se elige ϕ_1 de manera que la componente calculada, Z_1 , tenga la máxima varianza posible ajustándose a la condición de que tiene que mantener la perpendicularidad. Para mantener la ortogonalidad en la transformación, el módulo del vector de pesos $j = (\phi_{1j}, \phi_{2j}, \phi_{3j}, \dots, \phi_{pj})$, donde $j = 1, 2, \dots, p$ tiene que ser 1 [32]. Representado formalmente:

$$\sum_{k=1}^p \phi_{kj}^2 = 1$$

Ecuación 2.7.- Módulo del vector de pesos ortogonal

Estas componentes principales contienen la mayor proporción de varianza de los datos originales, se pretende que contengan al menos un 95% [12]. Para maximizar la varianza es común usar el método de los multiplicadores de Lagrange, para maximizar la función del siguiente planteamiento:

$$\text{Var}(Z_1) = \text{Var}(\phi'_1, x) = \phi'_1 \sum \phi_1$$

Ecuación 2.8.- Maximización de la varianza

Donde ϕ' es el vector de pesos formado por $\phi_{11}, \phi_{21}, \phi_{31}....$

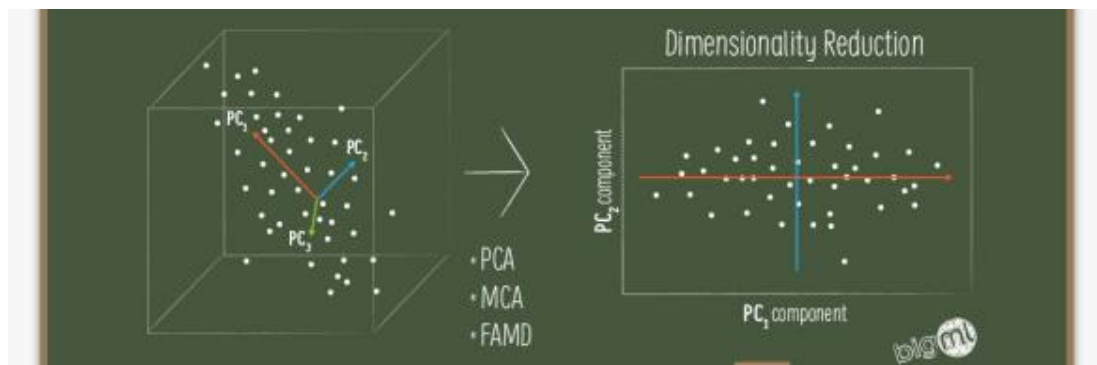


Ilustración 2.16.- PCA [33]

3. Metodología de trabajo

En esta sección se detallará cómo se ha realizado la investigación de este proyecto, detallando técnicas, materiales y procedimientos.

Para el desarrollo se han seguido los siguientes pasos:

- Elección del problema: Primero se ha escogido el tema de este trabajo en base al interés personal y su relevancia en el ámbito informático y en la actualidad.
- Definición y delimitación del problema: Se ha estudiado el problema y sus soluciones actuales para poder llevar a cabo un planteamiento completo de diferentes resoluciones.
- Elección y estudio de los datos: Los datos son una parte importante del proyecto que se han tenido que seleccionar y estudiar para que sean adecuados para el tipo de trabajo que se va a realizar.
- Preprocesamiento de los datos: Tras estudiar la estructura y organización de los datos se someten a un preprocesamiento. Su objetivo es preparar los datos para asegurar que sean compatibles con el modelo.
- Organización y definición de las soluciones al problema: Se han estudiado diferentes técnicas para realizar el diagnóstico de motores de una manera efectiva y acorde con la dificultad del proyecto. Finalmente se han elegido los métodos de distancia de Mahalanobis y Autoencoders tras ser contrastados con otras técnicas y analizar sus ventajas y desventajas.
- Desarrollo: Se ha implementado un programa que lleva a cabo la detección de anomalías en los aviones empleados en este proyecto usando las técnicas descritas anteriormente. Además, se ha realizado una clasificación de los aviones en anómalos

o no anómalos dependiendo del número de puntos atípicos que se encuentren en las señales de salud de sus motores.

- Revisión: Una vez desarrolladas las distintas soluciones para el problema, se analizan las puntuaciones obtenidas para los clasificadores y se comparan con otros métodos actuales para comprobar la calidad del modelo implementado.
- Elaboración de la documentación: Se ha documentado todo el procedimiento, técnicas utilizadas y definiciones teóricas donde se revisa todo el trabajo realizado.

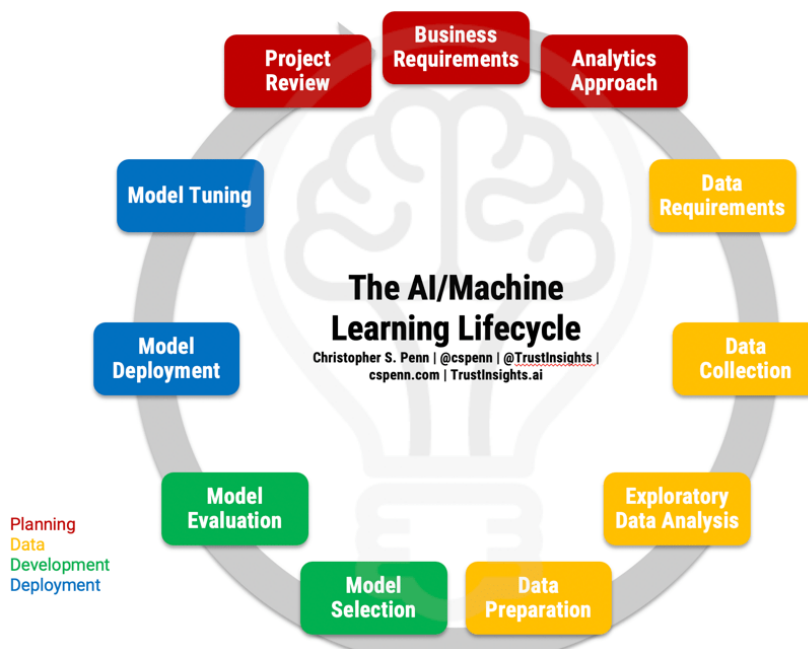


Ilustración 3.1.- Fases de un proyecto de IA [34]

3.1.- DATOS UTILIZADOS

Introduciremos al modelo dos conjuntos de datos diferentes: el conjunto CMAPPS (NASA) y el conjunto Rolls. Ambos conjuntos son dos sets de datos que contienen las diferentes medidas que se han realizado a varios motores de aviones. Las medidas de las señales se realizan cada cierto tiempo, siendo este tiempo un “ciclo”. Por tanto, las columnas serán las señales de salud y las filas representan el ciclo muestreado.

3.1.1.- Conjunto de datos “CMAPPS” (NASA)

Como se explicó anteriormente, el diagnóstico de motores a reacción se realiza mediante el análisis de ciertas variables que se usarán como señales de salud. En este caso, se cuenta con un dataset de la NASA derivado del entorno C-MAPSS (Commercial Modular Aero-Propulsion System Simulation), una simulación de un motor turbofán con un sistema de control realista. El conjunto de datos está dividido en 4 subconjuntos llamados FD001, FD002, FD003, y FD004, previamente divididos en conjuntos de entrenamiento y de test. Este conjunto de datos está organizado de la siguiente manera:

Nombre	FD001	FD002	FD003	FD004
Número muestras entrenamiento	100	260	100	249
Número muestras test	100	259	100	248
Condiciones de operación	1	6	1	6
Modos de fallo	1	1	2	2

Tabla 3.1.- Organización del dataset CMAPPS

En el conjunto de entrenamiento, la degradación del motor va aumentando hasta que se produce un fallo general del sistema, donde la medición finaliza. En el conjunto de test la serie finaliza antes del fallo del sistema, pero se conocen los valores de RUL al final de la serie que nos permiten conocer cuántos ciclos le quedan al motor.

Cada conjunto tiene 26 columnas que corresponden con las señales obtenidas. Los nombres reales son los siguientes:

'unit_number', 'time_in_cycles', 'setting_1', 'setting_2', 'TRA', 'T2': Total temperature at fan inlet, 'T24': Total temperature at LPC outlet, 'T30': Total temperature at HPC outlet, 'T50': Total temperature at LPT outlet, 'P2': Pressure at fan inlet, 'P15': Total pressure in bypass-duct, 'P30': Total pressure at HPC outlet, 'Nf': Physical fan speed, 'Nc': Physical core speed, 'epr': Engine pressure ratio (P50/P2), 'Ps30': Static pressure at HPC outlet, 'phi': Ratio of fuel flow to Ps30, 'NRF': Corrected fan speed, 'NRc': Corrected core speed, 'BPR': Bypass Ratio, 'farB': Burner fuel-air ratio, 'htBleed': Bleed Enthalpy, 'Nf_dmd': Demanded fan speed,

'PCNfR_dmd': Demanded corrected fan speed, 'W31': HPT coolant bleed, 'W32': LPT coolant bleed

A parte de las señales de salud que explicaremos a continuación, las columnas “unit_number” y “time_in_cycles” son relevantes, pues la primera nos indica la unidad o el motor que ha sido monitorizado, y la segunda indica el ciclo muestreado.

Las señales del motor que nos interesan son las siguientes:

Señal	Significado
DP30	Presión de salida del compresor de alta presión
DT30	Temperatura de salida del compresor de alta presión
TGT	Temperatura de salida del compresor
FF	Flujo de combustible (Fuel Flow)
EPRA	Empuje del avión (“acelerador”). No es una señal de salud, pero también sería relevante realizar su medición pues determina el punto de funcionamiento del “acelerador” del avión.

Tabla 3.2.- Señales de salud de los motores del conjunto de datos CMAPPS

3.1.2.- Conjunto de datos “Rolls” (Datos reales)

También se ha usado otro conjunto de datos, llamado Rolls, en el que se muestrean las señales de salud de diferentes motores de avión, esta vez reales. Como en el anterior conjunto, las filas serán los ciclos de vuelo y las columnas las señales de salud.

Libro Excel	Aviones con la turbina en estado X y con el compresor en estado Y					
Hoja	Avión N					
Columna	DT30	DFE	DTGT	DN2	EPRA	P30

Tabla 3.3.- Organización de los datasets del conjunto Rolls

En la tabla anterior se muestra como están organizados los datasets. Las columnas indicadas con las señales de salud son las que interesan para realizar la detección de anomalías en este

proyecto, sin embargo, en el conjunto de datos se incluyen más columnas que se eliminarán en el preprocesamiento.

Las mediciones de los aviones están distribuidas en diferentes archivos de Excel dependiendo del estado de su compresor (estado X en la tabla 4) y su turbina (estado Y). Las etiquetas para definir estado de los componentes son:

- G (Good)
- GN (Good to Normal)
- N (Normal)
- NH (Normal to High)
- H (High)
- B (Bad)

A continuación, la explicación de cada señal de salud.

Señal	Significado
P30	Presión de salida del compresor de alta presión
T30	Temperatura de salida del compresor de alta presión
TGT	Temperatura de los gases de la turbina
FF	Flujo de combustible (Fuel Flow)
N2	Revoluciones en el eje de alta presión
EPRA	Empuje del avión (“acelerador”). No es una señal de salud, pero también es relevante realizar su medición pues determina el punto de funcionamiento del “acelerador” del avión.

Tabla 3.4.- Señales de salud del dataset Rolls

3.2.- HERRAMIENTAS

Para programar el modelo se ha utilizado Python 3.8 con ayuda del entorno PyCharm. Se ha utilizado Python para diseñar e implementar el programa de detección de anomalías por ser uno de los mejores lenguajes que trabajan con la inteligencia artificial y la minería de datos, pues suministra una gran cantidad de recursos que otros no tienen. Python dispone de

numerosas librerías especializadas para deep learning, y aprendizaje automático (Numpy, Scipy, Keras...), tareas imprescindibles en este proyecto.

En concreto, se hará un gran uso de la librería Keras, una biblioteca de redes neuronales de código abierto escrita en Python que ofrece varias implementaciones de bloques modulares sobre los que se pueden desarrollar modelos de Deep Learning. Keras nos ha permitido crear el modelo de autoencoders, así como configurar el número de capas ocultas y el entrenamiento.

Otra biblioteca importante que se ha utilizado para la minería de los datos es Pandas, una de las herramientas más relevantes actualmente para el análisis de datos en Python, pues es esencial para manejar los conjuntos de datos y poder llevar a cabo un correcto preprocesamiento. Pandas se utiliza junto con otras librerías como Numpy, pues ha sido construida sobre esta última.

Numpy se ha utilizado para trabajar con las diferentes estructuras matriciales y vectoriales que se han necesitado a lo largo de la implementación del programa que se detallarán más adelante.

3.3.- PREPROCESAMIENTO

Como se ha explicado anteriormente, el preprocesamiento de los datos es un paso importante antes de llevar a cabo la detección de anomalías en sí, pues hay que suministrar al modelo unos datos compatibles con su funcionamiento y en el formato correcto. El preprocesamiento para los conjuntos de datos se realizará gracias a las librerías pandas y numpy.

Este proceso se puede dividir en 2 áreas: preparación de datos y reducción de datos:

- Preparación de datos: En esta fase se inicializan correctamente los datos para que los datos se ajusten correctamente a la entrada de los algoritmos que se vayan a utilizar. Sin esta fase, los algoritmos podrían no ejecutarse o dar resultados erróneos [4]. Hay diferentes técnicas dentro de la preparación de datos que se realizarán:

- Limpieza de datos: Elimina el ruido y resuelve incoherencias de los datos.
 - Imputación de valores perdidos: Cuando hay valores perdidos en una serie temporal, se sustituirán con otro valor adecuado.
 - Normalización: Su objetivo es escalar los datos para que tengan una forma uniforme realizando un ajuste en la distribución.
- Reducción de datos: Su objetivo es obtener una representación más reducida de los datos iniciales, sin perder la integridad y la información presente en ellos [4]. La selección de atributos es una de las técnicas de esta fase que se emplearán para escoger las características adecuadas para detectar y monitorizar correctamente las señales de salud de los motores. También se hará una selección de instancias, pues al contar con series temporales de tamaño variable se tendrán que ajustar el tamaño de las series para que sean todas iguales y del mismo tamaño.



Ilustración 3.1.- Técnicas para la preparación de datos [4]

Una vez comprendida la clasificación y los detalles de las tareas principales de preprocesamiento se decide que se llevarán a cabo las siguientes técnicas para preparar los datos correctamente:

1. Selección de atributos: Cada conjunto de datos tiene varias columnas, pero solo seleccionaremos las de las señales de salud que nos interese analizar (ya detalladas en la sección anterior), creando un subconjunto nuevo. Si el conjunto que estamos analizando no tiene alguna de las columnas deseadas, se cuenta como un conjunto no válido y se descartará.

- a. Para los datos de la NASA esta tarea se hace 2 veces ya que hay que seleccionar las columnas de cada archivo test y de entrenamiento, al estar previamente separados. En este caso con la función DataFrame es suficiente para seleccionar los atributos y el conjunto de datos correspondiente para cada motor/unidad.
 - b. Para los datos de Rolls hay que tener en cuenta que el conjunto de datos está en un archivo Excel y por tanto hay que usar la función específica de pandas “ExcelFile” que convierte la hoja Excel actual (recorriendo todas las hojas del archivo con un bucle) en un DataFrame.
2. Normalización: A los subconjuntos resultantes se les aplica una normalización o escalado utilizando las librerías y funciones correspondientes para que tengan una forma homogénea. Se usa el escalador MinMax que transforma las características escalándolas a un rango pasado por parámetro.
 3. Imputación de valores perdidos: Esta técnica solo se aplica al conjunto de Rolls, pues al ser datos reales pueden aparecer valores no válidos que tendremos que sustituir o rellenar con un valor correcto. En este caso, se sustituirá por el valor 0 gracias a la función “Fillna”.

Tras estos pasos ya podemos trabajar con un conjunto limpio, con los atributos correspondientes y del tamaño adecuado. El resultado es una estructura de datos tipo DataFrame que a su vez incluye distintos DataFrames para cada avión con los datos de las señales de salud del motor monitorizadas. Recorriendo adecuadamente el DataFrame el modelo identificará las anomalías presentes en los motores.

3.4.- DISEÑO

Para realizar el diagnóstico de los motores y la detección de anomalías, se diseñan dos modelos con una técnica distinta en cada uno: Distancia de Mahalanobis y Autoencoders. El

modelo de Autoencoders ha sido diseñado en base a una red neuronal con varias capas ocultas, con una estructura simétrica.

En cuanto a los datos, serán series temporales que habrá que ajustar y realizar un diseño homogéneo para todos los datos para permitir que los algoritmos trabajen con ellos sin ningún problema.

3.4.1.- Diseño de los datos

Observando el conjunto de datos se puede observar que las series temporales son de diferente tamaño, pues tienen una cantidad de ciclos muestreados diferentes. Esto supondrá un problema a la hora de usar el clasificador pues el funcionamiento será mejor si todos los vectores de señales a clasificar tienen la misma longitud. Para ello, diseñaremos un nuevo vector para cada conjunto de señales monitorizadas para cada motor que tendrá el mismo tamaño para todos. Esto se realizará mediante una interpolación, que hallará los datos dentro del intervalo en el que se conocen los valores extremos, haciendo que el vector sea más grande o pequeño.

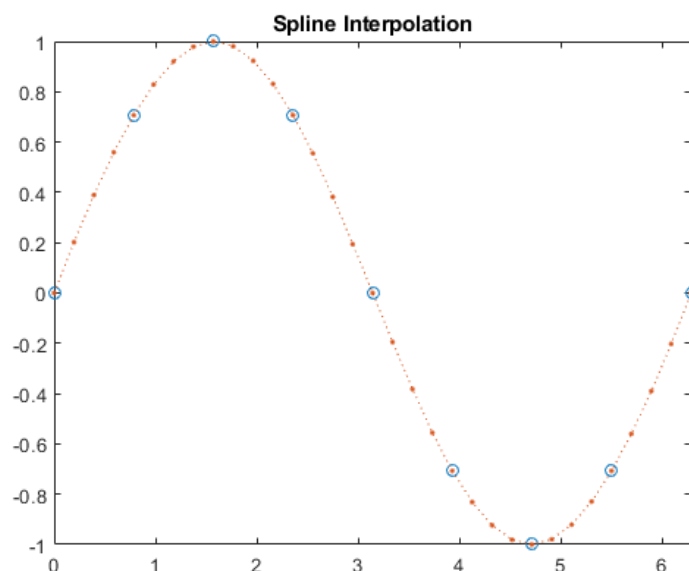


Ilustración 3.2.- Ejemplo de una interpolación [35]

En la anterior ilustración, se muestra un ejemplo de interpolación. Los puntos naranjas son los puntos calculados mientras que los azules son los originales. En este caso, el tamaño del vector está siendo incrementado.

3.4.2.- Diseño experimental

Para que el modelo logre identificar los ciclos anómalos, es necesario dividir los ciclos de cada motor en test y entrenamiento, pues hay que proporcionarle al modelo los ciclos considerados como normales para entrenarse y que pueda después identificar los anómalos. Esta fase es clave para que el modelo pueda clasificar finalmente a los motores dependiendo de su número de ciclos atípicos y comprobar si el avión se está comportando de una forma normal o no.

- Para el conjunto de datos reales Rolls, los primeros 100 ciclos se considerarán como normales y serán parte del conjunto de entrenamiento. El modelo tras ser entrenado con estos datos ya tiene como referencia ciclos normales para poder analizar el resto de los ciclos y poder identificar si son anómalos o no comparándolos con la información que tiene tras el entrenamiento.
- Para el conjunto de la NASA se ha decidido que el 30% de los ciclos (empezando por el principio del dataset, los primeros ciclos) serán considerados como normales. El 70% serán de test.

Recordemos que este proceso es una detección de anomalías semi supervisada, donde solo se etiquetan previamente los puntos típicos.

Para la clasificación de los motores, donde emplearemos un clasificador, que según los ciclos atípicos identificados previamente por nuestros modelos clasificará los aviones como anómalos o no, el conjunto de datos se divide en dos partes, algunos motores serán de entrenamiento, que serán la mayoría, y otra pequeña parte de test para evaluar el modelo entrenado.

Los motores de la NASA ya están separados en entrenamiento y test, pero la distribución de Rolls no. Se empleará una función propia del clasificador empleado que separa los datos en train y test para su evaluación.

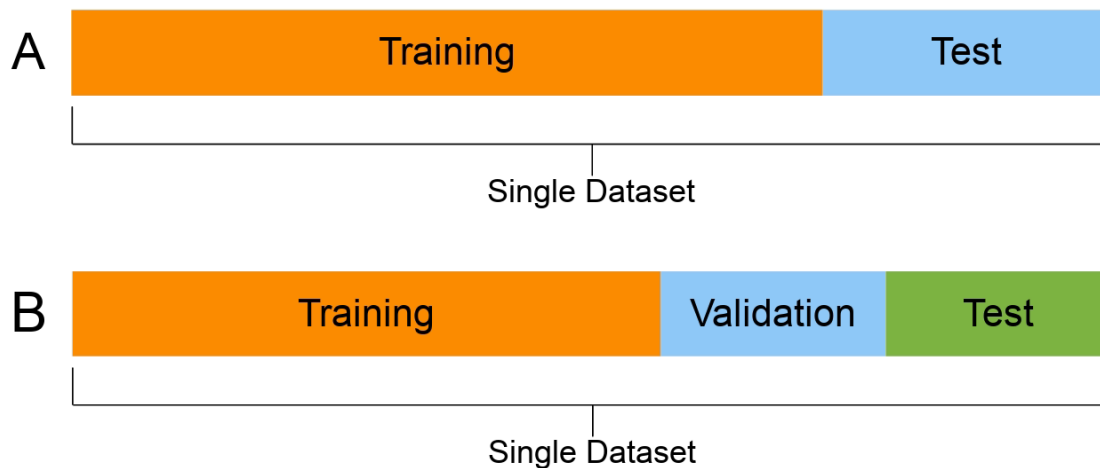


Ilustración 3.3.- 2 ejemplos para la partición de un dataset [36]

La imagen anterior representa gráficamente la división del conjunto de datos en entrenamiento y test. Para el modelo de Autoencoders se usará la partición del dataset B, donde hay una tercera parte que se usa para la validación, en concreto será un 5% de los datos. Esta partición es para realizar una validación que comprueba si el modelo está identificando los puntos anómalos correspondientes en datos que no son de entrenamiento, proporcionando una evaluación más imparcial. También sirve para comprobar si los hiperparámetros son correctos y si no es así, ajustarlos para evitar obtener errores derivados de, por ejemplo, un sobreentrenamiento (valor demasiado alto para el parámetro “epochs”, número de veces que se entrena el modelo). El modelo no aprende nada con esta validación, simplemente ayuda a la etapa de su desarrollo [37].

Para clasificar a los aviones tras haber identificado los ciclos atípicos se usará el clasificador Time Series Forest. Es un clasificador especializado en series temporales basado en Random Forest, un algoritmo de machine learning que ajusta un número determinado de árboles de decisión en diferentes submuestras. Utiliza promedios para mejorar su precisión [38]. Al usar varios árboles se evita el sobreajuste (overfitting) de los datos de entrenamiento que puede causar un árbol individual.

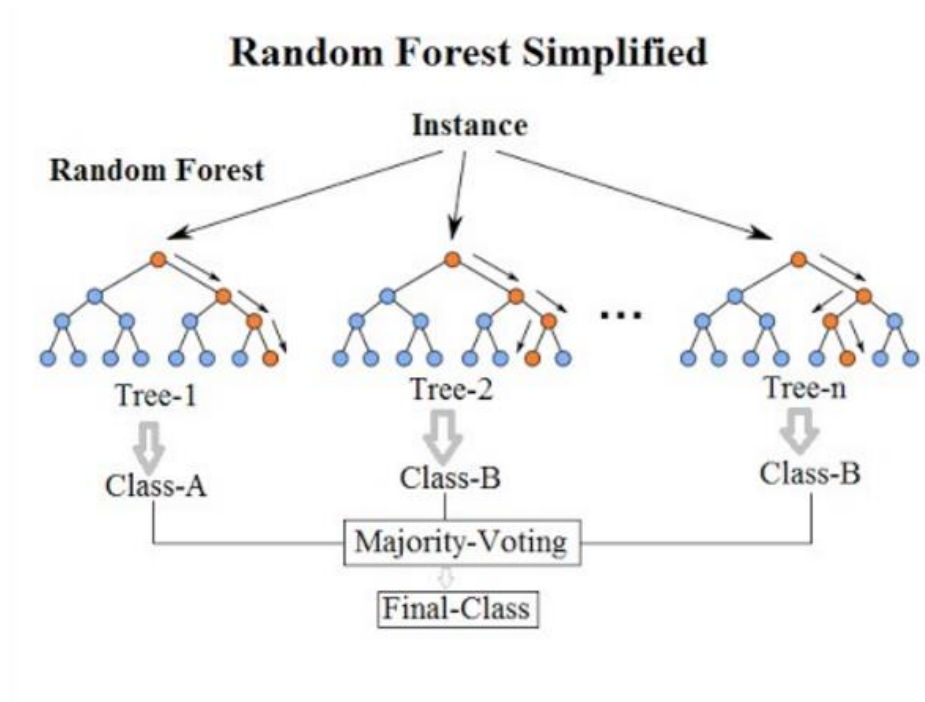


Ilustración 3.4.- Representación de los árboles de decisión Random Forest [39]

En la imagen anterior se puede observar como funciona Random Forest. El algoritmo selecciona muestras de forma aleatoria, después crea un árbol de decisión para cada una, y a continuación cada árbol “vota” a que clase pertenece. En este caso, tendrán que votar si el avión es anómalo (1) o no (0). La mayoría será el resultado final [40].

Por otra parte, para evaluar al clasificador y comprobar cuántos aviones ha clasificado correctamente según las etiquetas que les hemos asignado previamente, se empleará la validación cruzada. Este tipo de validación proporciona una evaluación menos sesgada que otros métodos. Divide el conjunto de entrenamiento en k trozos, que suele estar entre 5 y 10. Cada parte se entrena con el resto menos uno, que será para la evaluación (test). El proceso se repite k veces, y el resultado será la estimación del error medio de todas las repeticiones.

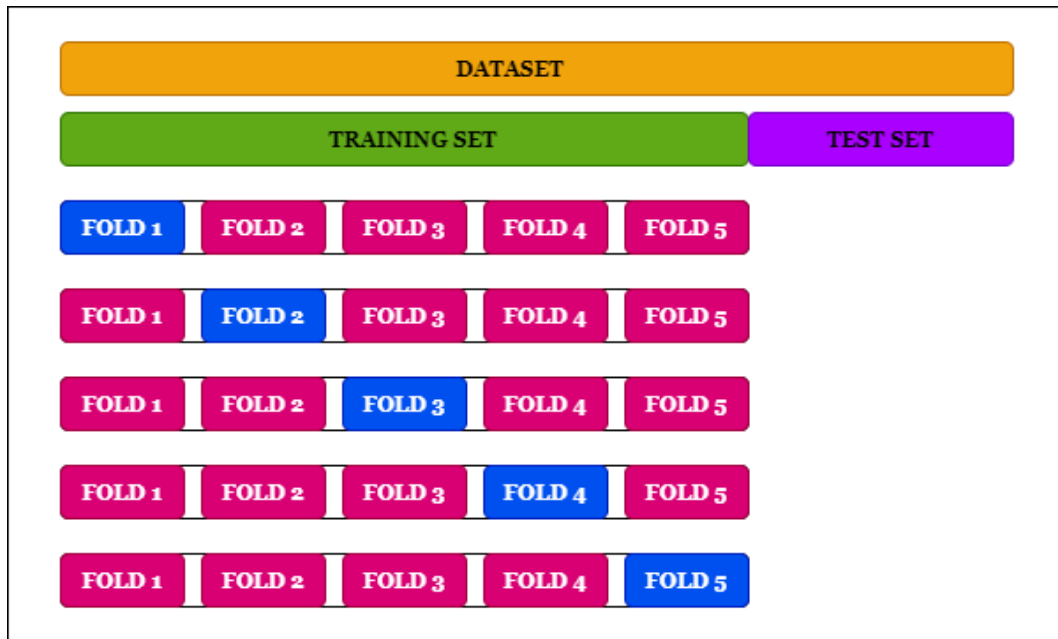


Ilustración 3.5.- Representación de la validación cruzada [41]

En la anterior ilustración, las partes en color azul son para test, y las partes en color rosa para el entrenamiento. En este caso, se ha elegido el valor 5 para el parámetro k , por tanto, se ha dividido en 5 trozos.

Se ha elegido este método de validación porque una de sus ventajas es que reduce el sobreajuste, al dividir el conjunto de datos en diferentes secciones. También ayuda a configurar correctamente los hiperparámetros del algoritmo.

4. Trabajo realizado y resultados obtenidos

4.1.- FORMATO Y REPRESENTACIÓN DE LOS DATOS

- Conjunto de datos CMAPPS: El programa procesa los ficheros de formato csv (archivo .txt) con ayuda de la librería pandas que implementa diferentes métodos para extraer datos de ficheros. Estos ficheros están formados por diferentes muestras obtenidas durante una serie de repeticiones, llamadas series temporales. Tiene un gran número de columnas ya detalladas en la sección anterior, separadas por un espacio. Todos los datos numéricos son números decimales. Cada motor está identificado en la columna ‘unit_number’, con un número entero.
- Conjunto de datos Rolls: Los aviones están distribuidos a lo largo de diferentes archivos tipo Excel, clasificados por el estado de su turbina y compresor (Good, Good to Normal, etc...) como se explicó en la sección anterior. Cada hoja del archivo Excel contiene una serie temporal de un avión. La identificación del avión se encuentra en el nombre de la hoja: N° de Hoja – ID Avión. Este ID del avión también se encuentra en la primera columna de la serie temporal.

4.2.- PREPARACIÓN Y REDUCCIÓN DE LOS DATOS

Para el conjunto de datos de CMAPPS, se han extraído las columnas correspondientes a las señales de salud que se analizarán (unit number, P30, T30, T50, epr) de cada conjunto, entrenamiento y test. También se crea una nueva columna “FF”, que será el producto de phi por PS30. Hay que tener en cuenta que se prescinde de los valores de EPRA, pues son todos iguales y no aportan ninguna información. Se crean dos estructuras de datos diferentes para almacenar el conjunto de datos de entrenamiento y el de test, dos archivos ya separados que serán recorridos para ir almacenando las variables anteriormente mencionadas.

Se creará una estructura de datos DataFrame compuesta de las columnas P30, T30, TGT y FF para los datos de cada motor. Se recorre cada archivo (test y entrenamiento) por separado utilizando un bucle while en el que cada serie temporal para cada motor se va introduciendo a cada modelo y se detectan los ciclos anómalos.

No se dispone de la RUL (vida útil restante) de los datos de entrenamiento, así que habrá que calcularla, pues recordemos que en este conjunto las señales de salud se monitorizan hasta que hay un fallo genérico de motor y éste deja de funcionar, por tanto, no existe RUL, ya que se lleva hasta el límite (quedarían 0 ciclos restantes).

Motor	Ciclo	Señales de salud
1	1	...
1	2	...
1	3	...
1	4	...
1
1	76	...
2	1	...
2	2	...
2
...

Tabla 4.1.- Ejemplo de cómo está representado el conjunto de datos de entrenamiento.

En la anterior tabla se muestra un ejemplo de la organización de los datos de entrenamiento para el conjunto CMAPPS. En este caso, se monitorizan los datos del motor 1 hasta que falla en el ciclo 76. Inmediatamente después aparecen los datos del motor siguiente (el 2).

Para calcular la RUL de los motores de test, eliminaremos los últimos N ciclos de cada motor i. La RUL de cada motor será N, pues son los ciclos restantes antes de que el motor falle. En el ejemplo de la tabla 6, si asignamos 20 a N, la RUL del motor será 20, y en vez de aparecer 76 ciclos en la tabla aparecerán 56. N será un número aleatorio que se generará con la función *randint* entre 0 y el número de ciclos menos diez, para asegurarnos de que no asignamos una

RUL muy próxima a 0. Crearemos un vector que tendrá las x RULs calculadas previamente, donde x será el número de aviones del conjunto de entrenamiento.

Disponer de la RUL para cada motor es imprescindible para comprobar la eficacia del clasificador, pues se etiqueta previamente a cada motor con un 1 (anómalo) o un 0 (normal) de su RUL, que se ha calculado en el caso del conjunto de datos de entrenamiento, o la RUL ya presente en el caso del conjunto de test. Si la RUL es menor de un límite (que escogerá su valor antes de lanzar el programa), el motor se etiqueta como anómalo pues le quedan pocos ciclos y podría necesitar una revisión o reparación. Después, los resultados del clasificador, que clasificará los aviones como anómalos o no según los datos obtenidos de los modelos, se comparan con los datos etiquetados inicialmente y se calculará el % de aciertos que ha tenido.

Para el conjunto de Rolls, hay que extraer también las columnas correspondientes. Al ser un problema real, existen muchos datos no válidos (NaN, Not a Number), vacíos, o que no contengan algunas de las columnas que se quieren monitorizar, por lo que hay que realizar un preprocesamiento para hacer una limpieza de estos datos inválidos. Los datos NaN se sustituirán por el valor 0 con la función.

Antes de proceder con el diagnóstico, se etiqueta cada avión como anómalo o no. Los aviones que tengan los dos componentes con estado Good, o Good to Normal, serán etiquetados como normales (se añade un 0 al vector). Este etiquetado servirá para luego comparar este vector “base” con el resultado del clasificador y ver si sus aviones anómalos coinciden con el etiquetado que se ha realizado manualmente.

4.3.- MODELOS ELABORADOS

- Distancia de Mahalanobis: Se recorren todas las filas de datos, es decir todos los ciclos muestreados para cada avión, y se calcula la matriz de covarianza de todos los puntos del conjunto de entrenamiento, así como su media. Con estos dos valores ya se puede calcular la distancia de Mahalanobis con la fórmula explicada en secciones anteriores para datos de entrenamiento y de test, con ayuda de una función previamente definida.

Se recorren todos los puntos con sus diferentes distancias de Mahalanobis, y si la distancia sobrepasa el umbral, previamente calculado con otra función (media de la distribución * 2), será etiquetado como anomalía.

- Autoencoders: Se ha diseñado una red neuronal simétrica con 3 capas ocultas. La primera capa consta de un tamaño de entrada igual al del vector entrante, y su salida será de tamaño 10. La siguiente capa será de tamaño 2, la siguiente de 10, y la última del tamaño original de entrada, recordemos que el objetivo de los autoencoders es reconstruir los datos iniciales. A continuación, se muestra la estructura de la red:

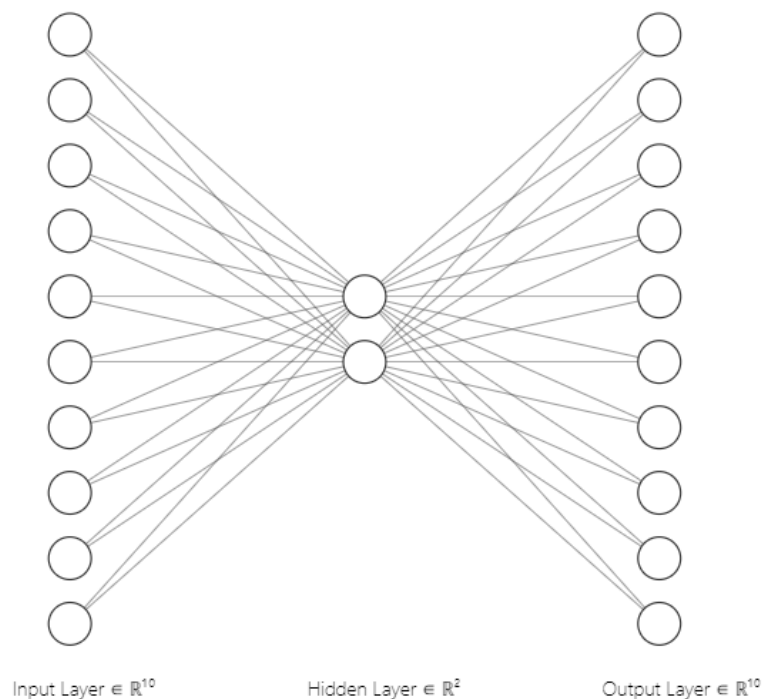


Ilustración 4.1.- Estructura de la red neuronal con autoencoders [42]

Una vez creados y configurados los modelos, se lleva a cabo la fase de experimentación donde se probará su eficacia y se verá si cumple con los objetivos iniciales. Cada experimento será ejecutado variando distintos parámetros para ver cómo actúa el modelo ante situaciones distintas y ver si con qué hiperparámetros rinde mejor. Los experimentos serán lanzados unas entre 3 y 5 veces para cada configuración, para asegurarnos de obtener unos resultados que no varían mucho para cada lanzamiento.

% de anomalías	CMAPPS	Rolls
40	x	x
50	x	
60	x	
70	x	
80	x	x

Tabla 4.2.- Porcentaje de anomalías para cada experimento

Además, para el conjunto de Rolls también se cambia el valor del número de ciclos considerados como normales, experimentando con los valores: 50, 100, 200 y 300.

4.4.- RESULTADOS

En esta sección se muestra la aplicación de las funcionalidades desarrolladas a los conjuntos de datos que se explicaron en el subapartado 4.1 y su correspondiente validación con el fin de comprobar la eficacia del modelo.

4.4.1.- Resultados para el dataset CMAPPS (NASA)

Para validar el clasificador se utilizan 2 tipos de validación para comprobar el porcentaje de aciertos:

1. La función accuracy score del clasificador que se ha usado previamente, que compara los valores dados como correctos con los que ha predicho el modelo.
2. Validación cruzada, explicada anteriormente empleando el método k-fold, que consiste en dividir los datos aleatoriamente en k grupos que se usan para entrenar el modelo, menos uno, el cual se utiliza como validación. Se repite el proceso k veces, variando el grupo de validación, obteniendo k estimaciones del error. La estimación final será la media de todas estas estimaciones. En este caso se usará el valor $k = 5$ [43].

A continuación, se presentan 2 tablas donde se recopilan el porcentaje de aciertos de cada uno de los modelos para cada método de validación, donde se han probado 5 porcentajes de anomalías distintos variando el valor de RUL límite, de manera que a partir de este valor se clasifica un periodo como anómalo o normal.

% anomalías	% de aciertos	
	train	validación cruzada
40%	90%	87%
50%	87%	81%
60%	70%	74%
70%	93%	85%
80%	100%	94%

Tabla 4.3.- Porcentaje de aciertos para el modelo de Distancia de Mahalanobis

Se observa que el desempeño del modelo empleando la distancia de Mahalanobis es correcto, casi un 90% de aciertos para un conjunto de datos en el que la mitad de los periodos son anómalos. Como es natural, con validación cruzada los resultados son algo peores.

% anomalías	% de aciertos	
	train	validación cruzada
40%	63%	87%
50%	76%	78%
60%	53%	75%
70%	73%	80%
80%	100%	84%

Tabla 4.4.- Porcentaje de aciertos para el modelo de Autoencoders (100 EPOCHS)

Por otra parte, el modelo en el que se han usado los autoencoders ha rendido algo peor que el modelo anterior, el porcentaje de aciertos ha disminuido bastante. Este porcentaje de aciertos ha sido obtenido para un modelo en el que se ha configurado con 100 EPOCHS, a continuación, se adjunta una tabla con el porcentaje de aciertos para una configuración con 200 EPOCHS.

% anomalías	% de aciertos	
	train	validación cruzada
40%	83%	85%
50%	87%	76%
60%	66%	78%
70%	100%	83%
80%	100%	84%

Tabla 4.5.- Porcentaje de aciertos para el modelo de Autoencoders (200 EPOCHS)

Con 200 EPOCHS el modelo rinde bastante mejor, pero no más que el modelo en el que se emplea la distancia de Mahalanobis. Este hiperparámetro es el número de veces que los datos de entrada pasan por la red neuronal. Hay que tener especial cuidado con este valor ya que resulta lógico pensar que cuantas más veces se entrene mejores resultados dará, pero hay que tener en cuenta para cada modelo hay un punto en el que la mejoría se estanca e incluso empeora, produciéndose un sobreentrenamiento.

4.4.2.- Resultados para el dataset Rolls

En este subapartado se presentan las tablas de resultados para el dataset de Rolls. Primero se analizará visualmente brevemente la aplicación de cada técnica a cada avión, donde se examina cada ciclo y el modelo lo clasifica como anómalo o no, dependiendo del criterio de cada técnica.

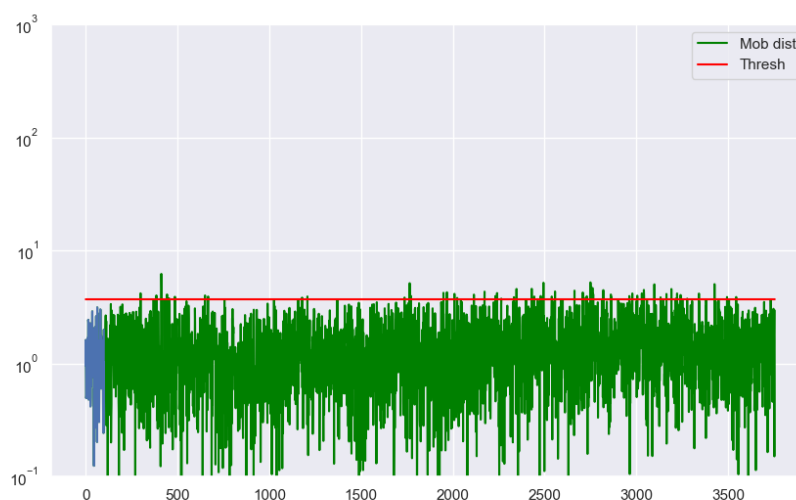


Ilustración 4.2.- Distancia de Mahalanobis para un avión con la turbina y el compresor en estado "Bueno"

En esta imagen se visualiza la distancia de Mahalanobis (en el eje Y) para cada ciclo (en el eje X), de manera que si este valor sobrepasa la línea roja, que simboliza el valor límite, se considera que ese ciclo ha sido anómalo, ya que la distancia de Mahalanobis de este punto es superior al resto. Los puntos en color azul de la serie son los ciclos que se han considerado en el preprocesamiento como normales, es decir 100.

Se observa que a partir del ciclo 1800 aproximadamente, la distancia va aumentando progresivamente, cada vez más puntos sobrepasan el límite y por tanto se consideran atípicos. Esto ocurre porque, naturalmente, el motor del avión se empieza a deteriorar tras un número determinado de ciclos en funcionamiento, el cual depende del estado en el que se encuentre el motor del avión. El motor de la figura estaba inicialmente categorizado con el compresor y la turbina en buen estado (Good compresor and Good turbine), por lo que el número de periodos anormales no es elevado, solo algunos al final.

Ahora se analizarán los ciclos de un avión con el motor en peor estado para ver como se comporta el modelo.

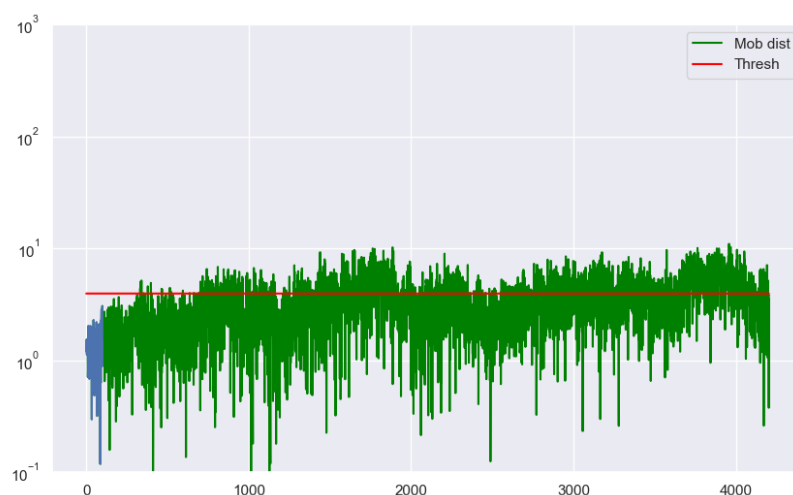


Ilustración 4.3.- Distancia de Mahalanobis para un avión con el estado del motor "Alto"

En este dataset se observa que a partir de los 700 ciclos aproximadamente, la distancia de Mahalanobis comienza a aumentar rápidamente, y sobre el ciclo 1200, la gran totalidad de los puntos son considerados como atípicos al superar el umbral de la distancia de

Mahalanobis calculada. Claramente se aprecia un deterioro del motor en comparación con el analizado anteriormente.

A continuación, como se ha realizado para el conjunto de datos de la NASA, se adjuntan las tablas de los porcentajes de aciertos de la clasificación que ha realizado este modelo.

Distancia de Mahalanobis				
Nº periodos considerados como normales	40% anomalías		90% anomalías	
	train	val. Cruzada	train	val. Cruzada
50 periodos	56%	58%	89%	88%
100 periodos	60%	61%	93%	84%
200 periodos	53%	55%	86%	90%
300 periodos	61%	56%	88%	88%

Tabla 4.6.- Porcentaje de aciertos para el modelo de Distancia de Mahalanobis para el dataset de Rolls

Los resultados no son buenos. Se obtiene un porcentaje de aciertos de un 57% de media para un preprocesamiento con el 40% de anomalías, lo que nos da a entender que no está realizando la clasificación correctamente. Al aumentar el número de anomalías, obtenemos una mayor tasa de aciertos, pero el resultado es el mismo, el modelo no está clasificando los aviones anómalos.

Ahora se analizará visualmente el modelo que se ha realizado con la red neuronal de autoencoders:

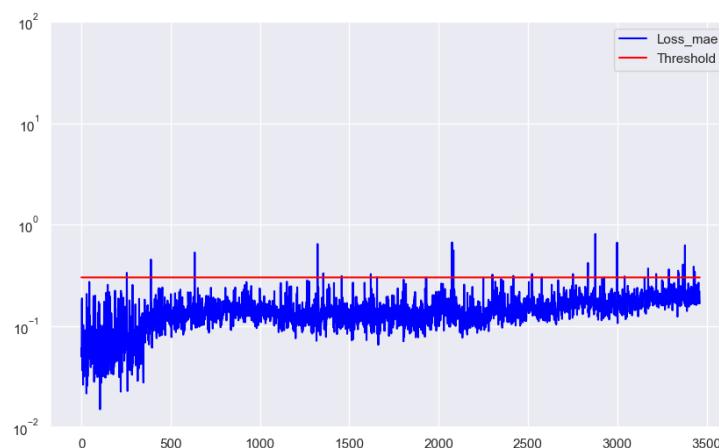


Ilustración 4.4.- Error en Autoencoders para un avión con el estado del motor "Bueno"

El gráfico, muy similar al de la anterior técnica, indica el valor de la función de pérdida, que es el error cuadrático medio que se obtiene al comparar el valor de la salida de la red neuronal con el de la entrada. Cuando más alto sea el error, más diferente es el valor de salida comparado con el de la entrada, que tendría que ser prácticamente igual, y, por tanto, más próximo a ser clasificado como una anomalía, según el límite fijado previamente.

En este caso, el avión tiene el motor en estado “Bueno”, y se puede apreciar que apenas hay cambios a lo largo de los ciclos salvo al final, que el valor de pérdida se incrementa ligeramente y la frecuencia de aparición de puntos atípicos aumenta, pero el rendimiento es bueno como cabría esperar.

En el siguiente avión con un motor más deteriorado, se observa que la cantidad de puntos anómalos es mucho mayor y empiezan a crecer mucho antes que en el anterior.

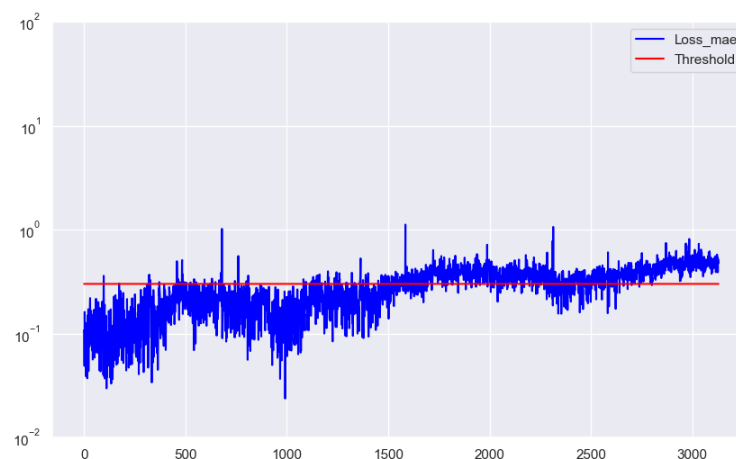


Ilustración 4.5.- Error en Autoencoders para un avión con el estado del motor "Malo"

En cuanto al clasificador, los resultados son similares para el modelo de autoencoders, donde la tasa de aciertos no es la adecuada para considerar este problema como resuelto.

En este caso se han variado distintos valores de los hiperparámetros de la red neuronal, los epochs y el número de capas. pero los resultados son bastante similares.

Autoencoders				
	40% anomalías		90% anomalías	
Hiperparámetros	train	val. Cruzada	train	val. Cruzada
100 EPOCHS	66%	61%	87%	90%
200 EPOCHS	64%	58%	86%	90%
20n -2n - 20n	50%	60%	90%	85%
5n-2n-20n	61%	58%	87%	89%

Tabla 4.7.- Porcentaje de aciertos para el modelo de Autoencoders para el dataset de Rolls

Las razones por las que los resultados no son los esperados son variadas, la principal causa es por la compleja naturaleza del conjunto de datos. Al ser un problema real y sin una solución de referencia asegurada, el problema pierde la sencillez que tiene, por ejemplo, el dataset que se usó anteriormente y donde sí se llegó a una solución razonable.

5. Conclusiones y trabajos futuros

En esta sección se mostrarán las conclusiones a las que se ha llegado y posibles trabajos futuros relacionados con este proyecto.

Tras el análisis de los resultados, se concluye que, se cumplió el objetivo principal que se planteó de detectar anomalías en motores a reacción usando las técnicas de distancia de Mahalanobis y los autoencoders. Sin embargo, en el dataset del problema real de Rolls no se ha llegado a ninguna solución válida. Las técnicas utilizadas son dos opciones válidas, que ayudan a identificar puntos atípicos correctamente. Las dos técnicas han mostrado resultados muy similares en cuanto a la tasa de aciertos en la clasificación de aviones anómalos, pero el modelo en el que se usó Mahalanobis mostró unos porcentajes ligeramente mayores.

Para lograr el objetivo inicial se han seguido los siguientes pasos:

- Primero se analizó el problema propuesto y sus diferentes maneras de resolverlo, la distancia de Mahalanobis, y una red neuronal con autoencoders
- Seguidamente se analizaron y preprocesaron los datasets y se diseñaron los modelos con las herramientas software correspondientes para llevar a cabo la detección de las anomalías
- Se implementó un clasificador para validar la eficacia de las técnicas
- Finalmente se han analizado los resultados y comparado su eficacia con 2 conjuntos de datos distintos

Una ampliación del trabajo sería probar más técnicas, a parte de la distancia de Mahalanobis y Autoencoders, para mejorar los resultados y intentar resolver el problema con el conjunto de datos reales de Rolls. Otra posible mejora podría ser realizar una predicción de la vida útil de los aviones dependiendo del número de ciclos atípicos identificados.

6. Bibliografía

- [1] P. Choudhary, «Introduction to Anomaly Detection,» Oracle Data Science, 5 Febrero 2017. [En línea]. Available: <https://blogs.oracle.com/ai-and-datascience/post/introduction-to-anomaly-detection>. [Último acceso: 3 Marzo 2021].
- [2] B. K. Mustagime Tülin Yildirim, «Aircraft Gas Turbine Engine Health Monitoring System by Real Flight Data,» *International Journal of Aerospace Engineering*, vol. vol. 2018, 14 Marzo 2018.
- [3] T. & M. D. Puranik, «Anomaly Detection in General-Aviation Operations Using Energy Metrics and Flight-Data Records,» *Journal of Aerospace Information Systems*, vol. 15, pp. 1-14, 2017.
- [4] S. R.-G. J. L. F. H. Salvador García, «Big Data: Preprocesamiento y calidad de datos,» *novática*, nº 237, 2016.
- [5] G. Ratto, «¿Qué diferentes tipos de motores de aviones hay?,» AirEuropa, 11 Julio 2016. [En línea]. Available: <https://billete996.aireuropa.com/que-diferentes-tipos-de-motores-de-aviones-hay/>. [Último acceso: 9 Abril 2021].
- [6] P. Sturesson, *Sense, Actuate and Survive: Ceramic Microsystems for High-Temperature Aerospace Applications*, 2018.
- [7] «Como Funciona un Motor a Reacción (I),» 23 Enero 2013. [En línea]. Available: <https://www.gtd.es/es/blog/como-funciona-un-motor-reaccion-i>. [Último acceso: 13 Marzo 2021].
- [8] G. R. Center, «Turbofan Engine,» [En línea]. Available: <https://www.grc.nasa.gov/www/k-12/airplane/aturbf.html>. [Último acceso: 6 Mayo 2021].
- [9] A. T. B. S. I. G. a. K. G. K. Amare D. Fentaye, «A Review on Gas Turbine Gas-Path Diagnostics: State-of-the-art methods, challenges and opportunities,» *Aerospace*, vol. 6, nº 7, p. 83, 2019 Julio 2019.
- [10] F. J. M. Benítez, *Métodos para la Diagnosis de Fallos en Turbinas de Gas de Aviación*, 2012.
- [11] L. T. a. J. D. X. Zhang, «Robust fault diagnosis of aircraft engines: a nonlinear adaptive estimation-based approach,» *IEEE Transactions on Control Systems Technology*, vol. 21, nº 3, p. 861–868, 2013.

- [12] L. S. Ramos, *Inteligencia Computacional aplicada al monitoreo de condición*, 2020.
- [13] A. Suresh, «How to Remove Outliers for Machine Learning?», 30 Noviembre 2020. [En línea]. Available: <https://medium.com/analytics-vidhya/how-to-remove-outliers-for-machine-learning-24620c4657e8>. [Último acceso: 17 Junio 2021].
- [14] G. a. S. C. a. C. L. a. H. A. V. D. Pang, «Deep Learning for Anomaly Detection,» *ACM Computing Surveys*, vol. 54, Abril 2021.
- [15] R. a. G. D. Karbauskaite, «Fractal-Based Methods as a Technique for Estimating the Intrinsic Dimensionality of High-Dimensional Data: A Survey Informatica,» *Informatica*, vol. 27, pp. 257-281, 2016.
- [16] «Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. 2000. Efficient algorithms for mining outliers from large data sets. SIGMOD Rec. 29, 2 (June 2000), 427–438. DOI:<https://doi.org/10.1145/335191.335437>».
- [17] D. M. Esmalifalak, «Anomaly detection with KNN,» 25 Junio 2020. [En línea]. Available: https://www.youtube.com/watch?v=RwmmtGrJs08&t=495s&ab_channel=Dr.MohammadEsmalifalak. [Último acceso: 25 Abril 2021].
- [18] G. P. C. S. C. M. Hu Wang, «Unsupervised Representation Learning by Predicting Random Distances,» 2020. [En línea]. Available: <https://www.ijcai.org/proceedings/2020/0408.pdf>. [Último acceso: 2 Junio 2021].
- [19] «Anomaly Detection with Autoencoders Made Easy,» Towards Data Science, 26 Octubre 2019. [En línea]. Available: <https://towardsdatascience.com/anomaly-detection-with-autoencoder-b4cdce4866a6>. [Último acceso: 28 Mayo 2021].
- [20] Y. W. a. L. L. M. Yuan, «Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network,» de *2016 IEEE international conference on aircraft utility systems (AUS)*, 2016.
- [21] J. I. Garzón, «Cómo usar redes neuronales (LSTM) en la predicción de averías en las máquinas,» 6 Noviembre 2018. [En línea]. Available: <https://blog.gft.com/es/2018/11/06/como-usar-redes-neuronales-lstm-en-la-prediccion-de-averias-en-las-maquinas/>. [Último acceso: 21 Abril 2021].
- [22] P. J. A. S. P. M. Mtra. María Teresa Escobedo Portillo, «P. CH. MAHALANOBIS Y LAS APLICACIONES DE SU DISTANCIA,» *Cultura Científica y Tecnológica*, n° 27, 2008.

- [23] H. Ghorbani, «Mahalanobis Distance And Its Application For Detecting Multivariate Outliers,» *Facta Univ Ser Math Inform*, vol. 34, nº 3, pp. 583-95, 2019.
- [24] S. Prabhakaran, «Mahalanobis Distance – Understanding the math with examples (python),» 15 Abril 2019. [En línea]. Available: <https://www.machinelearningplus.com/mahalanobis-distance/>. [Último acceso: 28 Marzo 2021].
- [25] «Tan cerca y tan lejos: la distancia de Mahalanobis,» 16 Abril 2020. [En línea]. Available: <https://caminosaleatorios.wordpress.com/2020/04/16/tan-cerca-y-tan-lejos-la-distancia-de-mahalanobis/>. [Último acceso: 16 Abril 2021].
- [26] I. C. Education, «IBM,» 17 Agosto 2020. [En línea]. Available: <https://www.ibm.com/cloud/learn/neural-networks>. [Último acceso: 20 Marzo 2021].
- [27] S. Flores, «Variational Autoencoders are Beautiful,» 15 Abril 2019. [En línea]. Available: <https://www.compthree.com/blog/autoencoder/>. [Último acceso: 27 Abril 2021].
- [28] V. Flovik, «How to use machine learning for anomaly detection and condition monitoring. Concrete use case for machine learning and statistical analysis,» 31 Diciembre 2018. [En línea]. Available: <https://towardsdatascience.com/how-to-use-machine-learning-for-anomaly-detection-and-condition-monitoring-6742f82900d7>. [Último acceso: 5 Junio 2021].
- [29] I. G. a. Y. B. a. A. Courville, «Chapter 14, Autoencoders,» de *Deep Learning*, MIT Press, 2016.
- [30] «¿Cómo realizar la compresión de datos utilizando codificadores automáticos?,» [En línea]. Available: <https://ichi.pro/es/como-realizar-la-compresion-de-datos-utilizando-codificadores-automaticos-228547757014583>. [Último acceso: 16 Abril 2021].
- [31] R. Khandelwal, «Anomaly Detection using Autoencoders,» 2021. [En línea]. Available: <https://towardsdatascience.com/anomaly-detection-using-autoencoders-5b032178a1ea>. [Último acceso: 24 Mayo 2021].
- [32] J. A. Rodrigo, «Análisis de Componentes Principales (Principal Component Analysis, PCA) y t-SNE,» Junio 2017. [En línea]. Available: https://www.cienciadedatos.net/documentos/35_principal_component_analysis. [Último acceso: 8 Abril 2021].

- [33] «Introduction to Principal Component Analysis: Dimensionality Reduction Made Easy,» [En línea]. Available: <https://blog.bigml.com/2018/12/07/introduction-to-principal-component-analysis-dimensionality-reduction-made-easy/>. [Último acceso: 12 Mayo 2021].
- [34] «Instant Insights: The AI/Machine Learning Lifecycle,» [En línea]. Available: <https://www.trustinsights.ai/resources/thought-leadership/instant-insights/instant-insights-the-ai-machine-learning-lifecycle/>. [Último acceso: 7 Mayo 2021].
- [35] «interp1 - Interpolación de datos,» [En línea]. Available: <https://es.mathworks.com/help/matlab/ref/interp1.html>. [Último acceso: 1 Junio 2021].
- [36] «Training, validation, and test sets,» Wikipedia, 6 Diciembre 2020. [En línea]. Available: https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets. [Último acceso: 1 Julio 2021].
- [37] T. Shah, «About Train, Validation and Test Sets in Machine Learning,» Towards Data Science, 6 Diciembre 2017. [En línea]. Available: <https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7>. [Último acceso: 16 Junio 2021].
- [38] «pyts.classification.TimeSeriesForest,» [En línea]. Available: <https://pyts.readthedocs.io/en/latest/generated/pyts.classification.TimeSeriesForest.html>. [Último acceso: 12 Julio 2021].
- [39] L. Sinai, 29 Septiembre 2020. [En línea]. Available: <https://liorsinai.github.io/coding/2020/09/29/random-forests.html>. [Último acceso: 12 Julio 2021].
- [40] F. Cardellino, «Tutorial para un clasificador basado en bosques aleatorios: cómo utilizar algoritmos basados en árboles para el aprendizaje automático,» freeCodeCamp, [En línea]. Available: <https://www.freecodecamp.org/espanol/news/random-forest-classifier-tutorial-how-to-use-tree-based-algorithms-for-machine-learning/>. [Último acceso: 12 Julio 2021].
- [41] S. Manna, «K-Fold Cross Validation for Deep Learning Models using Keras,» Medium, 20 Marzo 2020. [En línea]. Available: <https://medium.com/the-owl/k-fold-cross-validation-in-keras-3ec4a3a00538>. [Último acceso: 13 Marzo 2021].

- [42] «NN-SVG - Publication-ready NN-architecture schematics.,» [En línea]. Available: <http://alexlenail.me/NN-SVG/LeNet.html>. [Último acceso: 10 Junio 2021].
- [43] «sklearn.metrics.accuracy_score,» [En línea]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html. [Último acceso: 15 Marzo 2021].
- [44] I. I. A. M. Pedro Larranaga, «Tema 8. Redes Neuronales,» *Redes Neuronales, U. del P. Vasco*, vol. 12, p. 17, 1997.
- [45] J. Jordan, «Introduction to autoencoders.,» 19 Marzo 2018. [En línea]. Available: <https://www.jeremyjordan.me/autoencoders/>. [Último acceso: 26 Marzo 2021].
- [46] P. Wenig, «Local Outlier Factor for Anomaly Detection,» 5 Diciembre 2018. [En línea]. Available: <https://towardsdatascience.com/local-outlier-factor-for-anomaly-detection-cc0c770d2ebe>. [Último acceso: 30 Abril 2021].