



**Escola Politècnica Superior
d'Enginyeria de Vilanova i la Geltrú**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TRABAJO FINAL DE GRADO

**TÍTULO: INDUSTRIAL IoT.
MACHINE LEARNING EN LA INDSTURIA 4.0**

AUTOR: GARCÍA RODRÍGUEZ, DAVID

FECHA: Octubre, 2020

APELLIDOS: García Rodríguez

NOMBRE: David

TITULACIÓN: Grado en Ingeniería Mecánica

PLAN: 2009

DIRECTOR: Ester Simó Mezquita

DEPARTAMENTO: Departamento de Matemáticas

QUALIFICACIÓN DEL TFG

TRIBUNAL

PRESIDENTE

SECRETARIO

VOCAL

FECHA DE LECTURA: mes, año.

Este proyecto tiene en cuenta aspectos medioambientales: Sí No

RESUMEN

Con la llegada de la cuarta revolución industrial, la gran cantidad de datos disponibles y el alto poder de computación y almacenamiento de los dispositivos, el Machine Learning (ML) ha cobrado un gran protagonismo a nivel industrial. El ML permite abordar desafíos de fabricación, tales como la mejora de la calidad de los productos y la optimización de los procesos industriales.

El objetivo principal de este proyecto es destacar la importante herramienta que supone el ML en la industria y en los procesos de fabricación. También, pretende concienciar de la necesidad de añadir en el currículo académico de un ingeniero nociones básicas sobre esta novedosa herramienta.

Por un lado, se muestran y se definen los tres tipos de aprendizaje automático: supervisado, no supervisado y por refuerzo. Esta primera parte va acompañada de la presentación de algunos de los algoritmos más utilizados y los principales elementos para el correcto entrenamiento de estos.

Por otro lado, se recogen y sintetizan 45 casos donde se están aplicando técnicas de ML en el ámbito de la ingeniería mecánica. Estos casos se clasifican dependiendo del objetivo principal del algoritmo entre: diagnóstico y clasificación de fallos, predicción y pronóstico de defectos y optimización de sistemas.

Tras el análisis de los estudios de caso recogidos, se observa que se están aplicando con éxito las diferentes técnicas de ML. Este uso de algoritmos proporciona grandes resultados de precisión y fiabilidad que garantizan la productividad y eficiencia de los procesos.

Palabras clave:

Machine Learning	Industria 4.0	Big data	Caso de uso
Ingeniería mecánica	Predicción	Diagnóstico	Optimización

ABSTRACT

With the arrival of the fourth industrial revolution; large amounts of data, higher computing power and larger storage devices have become available. Machine Learning (ML) has taken on a major role at industrial level. ML enables you to tackle manufacturing challenges, such as improving product quality and optimizing industrial processes.

The main objective of this project is to highlight the importance of ML as a tool in the industry and manufacturing processes. This study also aims to raise awareness of the need for adding basic notions of ML into the engineering curriculum.

On one hand, the study defines three types of ML: supervised, unsupervised and reinforcement. This first part is accompanied by the presentation of some of the most used algorithms and the main elements for their correct training. On the other hand, 45 cases of ML techniques applied in the field of mechanical engineering are collected and synthesized. These cases are classified, depending on the main objective of the algorithm between: diagnosis and classification of failures, prediction and prognosis of defects and optimization of systems.

After analysing the collected case studies, it was observed that the different ML techniques are being applied successfully. The use of these algorithms provides great results of precision and reliability that guarantee the productivity and efficiency of the processes.

Keywords:

Machine Learning	Industry 4.0	Big data	Use case
Mechanical engineering	Prediction	Diagnosis	Optimization

ÍNDICE

RESUMEN.....	3
ABSTRACT.....	4
ÍNDICE DE ELEMENTOS.....	7
GLOSARIO	9
1. INTRODUCCIÓN.....	11
1.1 Contextualización.....	11
1.2 Motivación.....	13
1.3 Metodología seguida para la recopilación bibliográfica	13
1.4 Descripción de los capítulos de la memoria	14
2. MACHINE LEARNING	16
2.1 Datos	18
2.1.1 Ciclo de la vida de los datos.....	19
2.1.2 Metodologías de Data Mining.....	20
2.2 Clasificación de algoritmos de ML.....	24
2.2.1 Aprendizaje automático supervisado.....	24
2.2.2 Aprendizaje automático no supervisado.....	24
2.2.3 Aprendizaje automático por refuerzo.....	25
2.3 Elementos básicos de ML	25
2.3.1 Overfitting y underfitting	25
2.3.2 Métricas	28
2.4 Muestra de técnicas de ML	32
2.4.1 Regresión lineal simple	32
2.4.2 Regresión lineal múltiple	33
2.4.3 Regresión polinómica.....	34
2.4.4 Regresión logística (LR).....	35
2.4.5 Árbol de decisión (DT)	37
2.4.6 Bosque aleatorio (RF)	38
2.4.7 Redes neuronales (NN)	39
2.4.8 Redes neuronales convolucionales (CNN)	41
2.4.9 Autoencoders (AE).....	43

2.4.10	Máquina de vectores de soporte (SVM).....	45
2.4.11	K-Means.....	47
2.4.12	K-Nearest-Neighbor (KNN).....	48
2.4.13	Reinforcement Learning (RL).....	49
3.	ESTUDIOS DE CASO	52
3.1	Diagnóstico y clasificación de fallos	52
3.1.1	Caso de uso analizado.....	52
3.1.2	Recopilación de casos de uso.....	54
3.2	Análisis predictivo para el pronóstico de defectos	59
3.2.1	Caso de uso analizado.....	59
3.2.2	Recopilación de casos de uso.....	61
3.3	Optimización de la productividad.....	66
3.3.1	Caso de uso analizado.....	66
3.3.2	Recopilación de casos de uso.....	68
3.4	Software utilizado.....	72
4.	PRESUPUESTO.....	74
5.	CONCLUSIÓN.....	75
6.	AGRADECIMIENTOS	77
7.	REFERENCIAS.....	78

ÍNDICE DE ELEMENTOS

FIGURAS

Figura 1. Resumen de las cuatro revoluciones industriales.....	11
Figura 2. Evolución de crecimiento de datos anuales. IDC 2018	12
Figura 3. Proceso de aplicación en Ingeniería Mecánica usando ML/AI.....	16
Figura 4. Papel de la inteligencia basada en datos en la fabricación inteligente	17
Figura 5. Diagrama de la metodología de CRISP-DM.....	21
Figura 6. Dinámica general de la metodología SEMMA.....	23
Figura 7. Clasificación de modelos de ML.....	24
Figura 8. Punto óptimo de detención del entrenamiento	27
Figura 9. Proceso esquematizado de validación cruzada k-fold.....	28
Figura 10. Representación de la matriz de confusión.....	29
Figura 11. Ejemplo de modelo de regresión lineal simple	32
Figura 12. Ejemplo de modelo de regresión lineal múltiple	33
Figura 13. Comparación entre aplicación del modelo de regresión lineal y regresión polinómica	35
Figura 14. Modelo de regresión logística con función sigmoide	36
Figura 15. Partes y funcionamiento básico del algoritmo DT.....	37
Figura 16. Modelo de bosque aleatorio.....	38
Figura 17. Modelo de una neurona	39
Figura 18. Proceso de convolución de una imagen con un filtro	41
Figura 19. Arquitectura de dos etapas de una DCNN	43
Figura 20. Arquitectura de un autoencoder	44
Figura 21. Ilustración gráfica de SVM	45
Figura 22. Representación de la transformación sobre los datos	47
Figura 23. Evolución del modelo K-means hasta que se estabilizan los centroides	48
Figura 24. Proceso de toma de decisiones del algoritmo KNN.....	49
Figura 25. Estructura de aprendizaje por refuerzo	50
Figura 26. Diagrama de barras sobre el software utilizado en los casos de uso de ML revisados	72

TABLAS

Tabla 1. Ejemplos de underfitting y overfitting en regresión y clasificación	26
Tabla 2. Casos de uso de diagnóstico de fallos	54
Tabla 3. Parámetros de las máquinas.....	61
Tabla 4. Casos de uso de predicción de defectos.....	61
Tabla 5. Casos de uso de optimización	68
Tabla 6. Resumen de costes del trabajo.....	74

GLOSARIO

AE	Autoencoder
AI	Artificial Intelligence (Inteligencia artificial)
ANN	Artificial Neural Network (Red neuronal artificial)
BP	Back-Propagation (Retropropagación)
CNN	Convolutional Neural Network (Red neuronal convolucional)
CPS	Cyber-physical Systems (Sistemas ciberfísicos)
CRISP-DM	Cross Industry Standard Process for Data Mining (Proceso estándar industrial para minería de datos)
DBN	Deep Belief Networks (Red de creencia profunda)
DL	Deep Learning (Aprendizaje profundo)
DM	Data Mining (Minería de datos)
DNN	Deep Neural Networks (Red neuronal profunda)
DT	Decision Tree (Árbol de decisión)
GBM	Gradient Boosting Machine (Máquina de aumento de gradiente)
GRU	Gated Recurrent Unit (Unidades recurrentes cerradas)
GPR	Gaussian Process Regression (Proceso Gaussiano de regresión)
HMM	Hidden Markov Model (Modelo oculto de Markov)
IF	Isolation Forest (Bosque de aislamiento)
IIoT	Industrial Internet of Things (Internet de las cosas industrial)
IoT	Internet of Things (Internet de las cosas)
KDD	Knowledge Discovery in Databases (Extracción de conocimiento en base de datos)
KNN	K-Nearest-Neighbor (K vecinos más próximos)

LR	Logistic Regression (Regresión logística)
LSTM	Long Short-Term Memory (Memoria a largo plazo)
MAE	Mean Absolute Error (Error absoluto medio)
MARS	Multivariate Adaptive Regression Spline (Spline de regresión adaptativa multivariada)
MC	Multiple Classifier (Clasificador múltiple)
MDP	Markov Decision Process (Proceso de decisión de Markov)
MEISVM	Multivariable Ensemble-based Incremental Support Vector Machine (Máquina de vectores de soporte incremental multivariable basada en conjuntos)
ML	Machine Learning (Aprendizaje Inteligente)
MLP	Multilayer Perceptron (Perceptrón multicapa)
MSE	Mean Squared Error (Error cuadrático medio)
NN	Neural Network (Red neuronal)
RBM	Restricted Boltzmann Machine (Máquina de Boltzmann restringida)
RF	Random Forest (Bosque aleatorio)
RL	Reinforcement Learning (Aprendizaje por refuerzo)
RMSE	Root Mean Square Error (Error cuadrático medio de raíz)
RNN	Recurrent Neural Networks (Red neuronal recurrente)
SEMMA	Sample, Explore, Modify, Model, and Assess (Muestra, explorar, modificar, modelar y evaluar)
SVM	Support Vector Machines (Máquinas de vectores de soporte)
VC-DM	Virtuous Cycle of Data Mining (Ciclo virtuoso de minería de datos)

1. INTRODUCCIÓN

1.1 Contextualización

El término Ingeniería viene definido según el Diccionario de la Lengua de la Real Academia Española como el conjunto de conocimientos orientados a la invención y utilización de técnicas para el aprovechamiento de los recursos naturales o para la actividad industrial [1].

Desde tiempos remotos, los humanos han tenido tendencia a buscar avances tecnológicos, haciendo así que la evolución de la historia de la humanidad haya ido siempre ligada a la evolución de la historia de la tecnología y más concretamente, a la evolución de las máquinas [2].

Nos remontamos a finales del siglo XVIII, cuando la invención de máquinas basadas en el aprovechamiento de energía generada por vapor da pie a la primera revolución industrial [Figura 1]. Posteriormente, a principios del siglo XX, la segunda revolución industrial viene marcada por la introducción de la electricidad y el petróleo como fuente energética para la producción en masa, dando lugar a la cadena de producción y la división de trabajo. Es a principios del año 1970 cuando estalla la tercera revolución industrial, que destaca por la automatización de procesos basada en sistemas electrónicos y tecnologías de la información (TIC). Hoy en día la humanidad se encuentra a las puertas de la cuarta revolución industrial [3, 4].

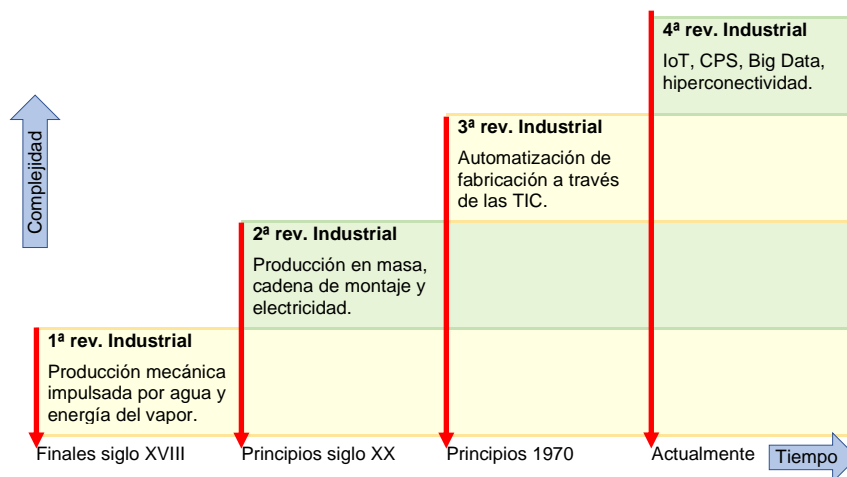


Figura 1. Resumen de las cuatro revoluciones industriales (Fuente propia)

La idea de industria 4.0, también conocida como cuarta revolución industrial, apareció por primera vez en el panorama mundial en la feria industrial alemana de Hannover celebrada el año 2011 [5]. Esta nueva revolución tiene como objetivo la mejora de

calidad de los productos, la optimización de los procesos y la reducción del coste estimado de estos, y se basa en el incremento masivo de datos (Big Data) como consecuencia de la implementación de los sensores y el desarrollo del internet de las cosas (IoT). International Data Corporation [6], tal y como muestra en la Figura 2 pronostica el crecimiento de datos anuales hasta 175 zettabytes en 2025, un incremento increíble comparado con los 33 zettabytes acumulados en 2018.

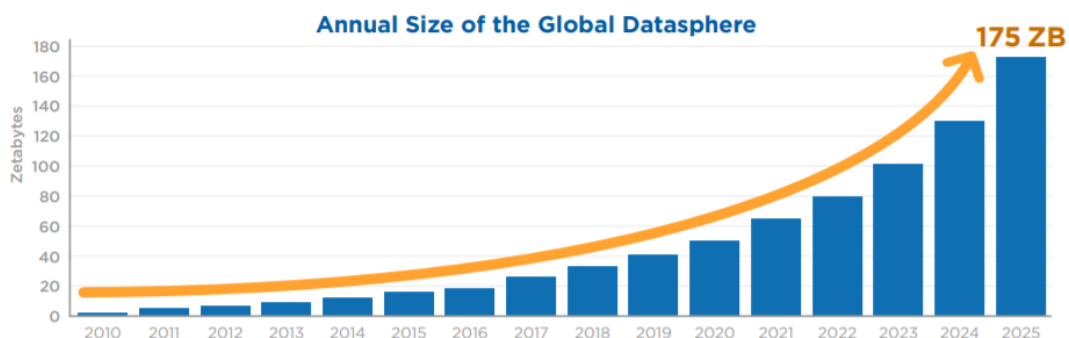


Figura 2. Evolución de crecimiento de datos anuales. IDC 2018. (Extraído de [6])

Derivado del término IoT y también muy importante en la evolución de la cuarta revolución industrial, aparece en 2012 el concepto de Internet de las Cosas Industrial (IIoT). IIoT se refiere a una red de dispositivos industriales conectados por tecnologías de comunicaciones con el propósito de monitorear, recopilar, intercambiar y analizar ideas para impulsar decisiones empresariales más inteligentes y rápidas, es decir, para mejorar la eficiencia de los procesos industriales y de fabricación [7].

La inteligencia artificial (AI), los sistemas ciberfísicos (CPS), la acumulación masiva de información (Big Data) y el uso de algoritmos para automatizar procesos mediante aprendizaje automático (ML) son otros de los elementos fundamentales de la novedosa industria 4.0.

La llegada de la industria 4.0, la gran cantidad de datos disponibles, el alto poder de computación de los dispositivos y la gran capacidad de almacenamiento de los mismos, son la causa de que las técnicas de Machine Learning (ML) se hayan convertido en una solución atractiva para abordar desafíos de fabricación. El ML ofrece una gran cantidad de herramientas para extraer información de datos que puede traducirse en conocimiento para enriquecer y transformar las aplicaciones industriales.

En este trabajo se presenta una visión general de las técnicas de aprendizaje automático y se realiza una revisión de la literatura científica para analizar como las técnicas de ML se están convirtiendo en herramientas clave en el entorno industrial actual.

1.2 Motivación

Hoy en día, la competitividad de las empresas está determinada principalmente por la posibilidad de satisfacer las necesidades del cliente de la mejor manera posible. Es decir, de suministrar al cliente productos o servicios que cumplan con sus requisitos en términos de tiempo, calidad y precio adecuados [8].

La gran mayoría de actividades tienden a la automatización con la finalidad de aumentar la rentabilidad de los procesos. Dado que la industria manufacturera incluye gran cantidad de procesos mecánicos es una muy buena candidata para la automatización.

La motivación principal de este trabajo es remarcar la importancia que supone para la industria la cuarta revolución industrial. Comentar el peso que está teniendo la automatización de procesos y la inteligencia artificial, dado que permiten la recopilación e interpretación masiva de datos para la futura optimización de procesos. Destacar la capacidad de mejorar el mantenimiento preventivo y el control de máquinas, entre otras muchas aplicaciones, para la reducción de costes dentro del proceso de producción.

Por todo esto, por la importancia que significa esta herramienta de cara al futuro y por tratarse de aire innovador dentro del ámbito industrial, el trabajo tiene por objetivo hacer reflexionar sobre la necesidad de incluir nociones básicas sobre las herramientas de ML en el currículo académico de un ingeniero industrial.

1.3 Metodología seguida para la recopilación bibliográfica

En esta sección, se presenta la metodología que se ha llevado a cabo en el proceso de búsqueda y recopilación de información para la elaboración del trabajo. En primer lugar, y como punto de partida, se encuentran los artículos “Methodologies of knowledge Discovery from data and data mining methods in mechanical engineering” [8], “Machine learning algorithms in production: a guideline for efficient data source selection” [9] y “Machine learning in manufacturing: advantages, challenge, and applications” [10]. Estos han sido el punto de inspiración y por lo tanto se han considerado los artículos base del proyecto.

El primer paso para la recopilación de información fue seleccionar artículos donde habían sido citados alguno de nuestros artículos base. Con estos, se creó una base de datos para recopilar los artículos recogidos de forma ordenada, asignando a cada uno de ellos un número y una letra (A/B/C para relacionar al artículo base que hace referencia). Esta base de datos se acompañó por una hoja de cálculo donde figuraba el

número de documento, título, autor y año de publicación.

En segundo lugar, y con el uso de la búsqueda directa de palabras clave, se seleccionaron artículos relacionados con “machine learning”, “data mining”, “Engineering”, “mechanical engineering”, “industrial engineering”, “use case”, “application”, “prognosis fault”, “optimization” o “fault diagnosis” en bases de datos como IEEE Xplore, ScienceDirect, Google Scholar, SpringerLink y ResearchGate.

Por último, y con intención de completar la información obtenida hasta el momento, se identificaron artículos a través de referencias hechas en los documentos previamente seleccionados. Este procedimiento fue muy necesario para poder indagar y profundizar en temas cruciales para el desarrollo del trabajo.

1.4 Descripción de los capítulos de la memoria

Este trabajo de final de grado se abre con una breve introducción histórica de la evolución tecnológica en el ámbito industrial, la contextualización del concepto industria 4.0 y los términos que la engloba. El cuerpo principal del trabajo se divide en dos enfoques. En primer lugar, en el capítulo 2 se encuentra la parte más teórica, donde se sitúa el ML y se exponen los conceptos que lo rodean. Y, en segundo lugar, en el capítulo 3 se hace una revisión analítica de casos donde la industria está aplicando técnicas de ML para obtener beneficios.

El capítulo 2, permite situar el punto en el que se encuentra el ML actualmente, empezando por el Big Data y las metodologías de minería de datos más utilizadas. A continuación, se exponen los tipos de algoritmos de ML, los elementos básicos a tener en cuenta para poder aplicarlos y una descripción de algunas de las técnicas de ML más utilizadas.

Es en el capítulo 3, Estudios de Caso, donde se identifican diferentes artículos que utilizan técnicas de ML para mejorar los procesos industriales y el software que emplean para llevarlas a cabo. Los estudios encontrados se han clasificado en tres categorías, dependiendo del motivo de la aplicación del algoritmo. Estas tres categorías son el diagnóstico y clasificación de fallos, el análisis predictivo para el pronóstico de defectos y la optimización de la productividad.

En capítulo 4, se plasma el presupuesto que supondría la realización del trabajo. Puesto que es un trabajo bibliográfico, principalmente se ha tenido en cuenta el coste que implican las horas de dedicación del autor.

Por último, en el capítulo 5 se concluye el TFG con la síntesis de las ideas básicas extraídas de la realización de este.

2. MACHINE LEARNING

La cuarta revolución industrial comprende el desarrollo de procesos de fabricación basados en automatización, robotización, comunicación entre máquinas y procesos de gestión de productos con soporte digital. El resultado de estas actividades es la creación de fábricas inteligentes. Estas fábricas se basan en la inteligencia artificial, y más concretamente en el aprendizaje automático [11].

El ML proporciona un poderoso marco de procesamiento de información capaz de enriquecer, y posiblemente incluso transformar, las líneas actuales de investigación en ingeniería mecánica y aplicaciones industriales [12].

El objetivo de las técnicas de ML varía en función del tipo de aprendizaje. Este objetivo va desde proporcionar a las máquinas la capacidad de predecir o generar patrones que describan relaciones, hasta centrarse en proporcionar la capacidad de aprender, adaptarse a los cambios y resolver problemas sin haber sido programadas específicamente para ello, tal y como podría hacer un humano. El ML se aplica en diferentes áreas de fabricación, como, por ejemplo, la optimización, el control y la resolución de problemas. La Figura 3 muestra el proceso de aplicación de ML en la ingeniería mecánica [10].

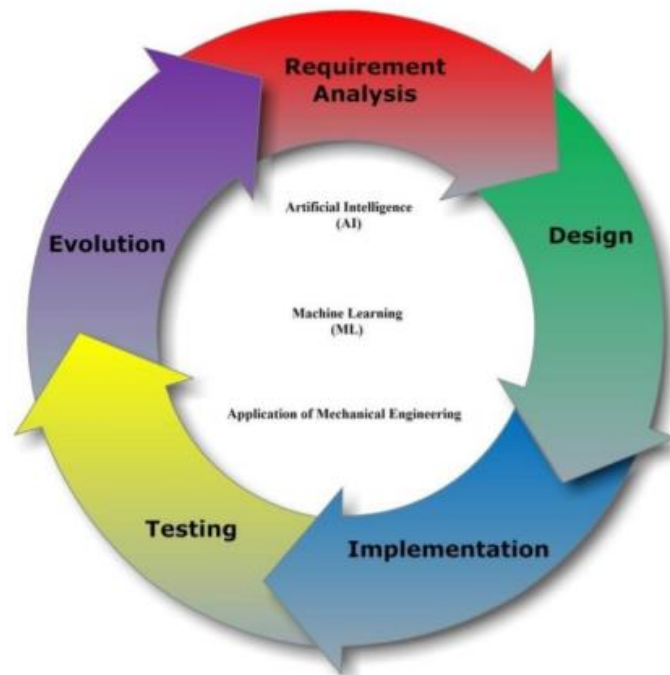


Figura 3. Proceso de aplicación en Ingeniería Mecánica usando ML/AI. (Extraído de [13])

Se considera que el ML es una rama importante de la también conocida inteligencia artificial (AI). Este término apareció en la década de los 50 del siglo pasado, cuando se cuestionó la posibilidad de describir la inteligencia de modo tan preciso que se pudiese construir una máquina capaz de simularla. Es decir, la AI es el conjunto de técnicas, métodos, herramientas y metodologías que nos ayudan a construir sistemas que se comportan de manera similar a un humano en la resolución de problemas concretos. Los objetivos principales de la AI son modelar conocimiento, formalizar los modelos, programar los operadores formales e implementar físicamente el soporte de esos programas [14, 15].

La inteligencia artificial se puede clasificar en función del grado de imitación a la inteligencia humana, diferenciando entre AI débil y AI fuerte. A día de hoy, no se ha programado ninguna máquina capaz de igualar o superar la capacidad humana promedio, que sería lo que se clasificaría como AI fuerte. Por este motivo, únicamente se ha visto este tipo de inteligencia en escenas de ficción.

Otros términos como Deep Learning o Big Data, junto con los que se han ido comentando anteriormente (inteligencia artificial, aprendizaje automático, internet de las cosas, etc.), completan el mapa conceptual que conduce a la fabricación inteligente [Figura 4].

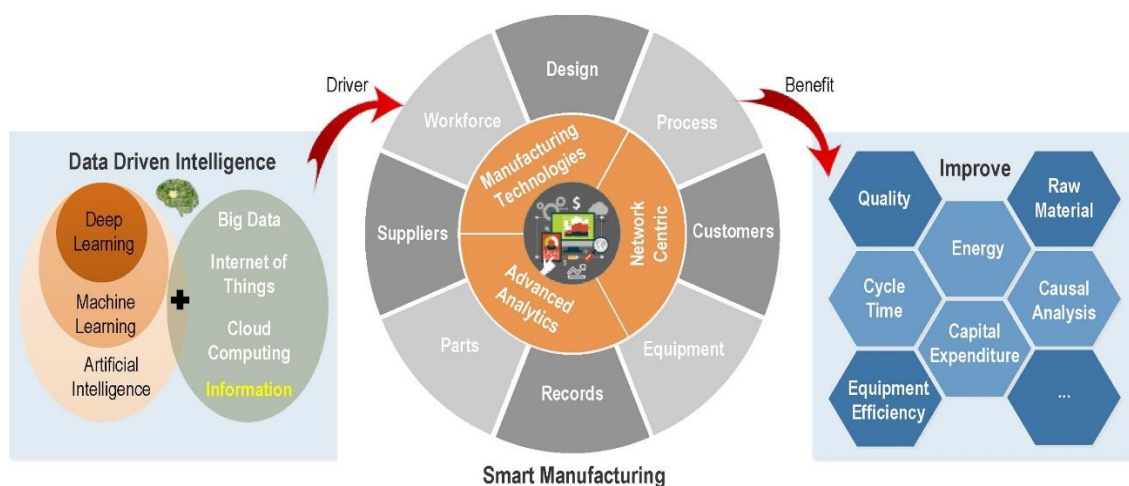


Figura 4. Papel de la inteligencia basada en datos en la fabricación inteligente. (Extraído de [16])

El Deep Learning (DL), en español conocido como aprendizaje profundo, es el término que acuñó Geoffrey Hinton en 2006 para explicar nuevas técnicas que proporcionan herramientas analíticas avanzadas capaces de procesar y analizar grandes cantidades de datos. El DL está catalogado como un subconjunto del aprendizaje automático [16].

El rendimiento de los algoritmos de aprendizaje automático depende en gran parte de la cantidad y la calidad del conjunto de datos. Es decir, el aprendizaje de estos algoritmos puede verse directamente condicionado por los datos de los que disponga. Tal y como se podía observar en la [Figura 2], en la última época ha aumentado de forma exponencial la cantidad de datos recopilados. Este suceso es el llamado Big Data.

El término Big Data hace referencia a la acumulación masiva de datos, llevado al punto de superar la capacidad de las herramientas tradicionales para ser registrados, gestionados y procesados en un tiempo razonable. Por lo tanto, un conjunto de datos se le categoriza dentro de Big Data cuando es demasiado grande para ser manejado de forma apropiada por los programas convencionales de software [17]. Los métodos usados para extraer conocimiento de estos grandes conjuntos de datos son los llamados métodos de Data Mining (DM). Estos métodos están basados en estadística matemática y/o en inteligencia artificial [8].

El primer apartado de este capítulo (2.1), hace referencia a los datos que se aplican en los métodos de ML. En este apartado se comentan las metodologías de procesamiento de datos más utilizadas, necesarias para prepararlos y posteriormente ser usados por los algoritmos de manera óptima. En el apartado (2.2), se presenta una clasificación de los algoritmos de ML basada en el tipo de aprendizaje y el objetivo de la aplicación de estos. En tercer lugar (2.3), se muestra una serie de elementos totalmente necesarios para poder entrenar los algoritmos de ML y disponer del mejor rendimiento posible. Por último, en el apartado (2.4) se hace una breve explicación de las técnicas más citadas en aplicaciones de ámbito industrial.

2.1 Datos

Según información extraída del artículo “Machine learning algorithms in machining: a guideline for efficient algorithm selection” de Ziegenbein et al. [18], y basándose en datos obtenidos en una encuesta hecha a empresarios sobre la aplicación de nuevas tecnologías y capacidades para hacer frente a cambios emergentes: el 60% de los encuestados piensa que son buenos o excelentes recopilando datos, pero solo el 32% se considera capaz de extraer la información valiosa de estos datos recopilados.

Desde el punto de vista industrial, el hecho de capturar y analizar eficientemente grandes cantidades de datos tiene como consecuencia directa la mejora de calidad y productividad respecto a los sistemas tradicionales [19]. La minería de datos es una técnica interdisciplinaria que se está volviendo popular en varias ramas de la ingeniería,

en especial en la ingeniería mecánica, ya que permiten descubrir información oculta en los conjuntos de datos y proporcionar soluciones rápidas para obtener análisis de profundidad sobre productos. En general, las técnicas de DM son capaces de analizar grandes cantidades de datos de forma efectiva en un período de tiempo reducido, siendo esto una tarea complicada para el ser humano [20].

En vista de la importancia y complejidad que supone extraer información útil de los datos analizados, los métodos de adquisición, recopilación, procesamiento y, sobre todo, exploración y análisis, se han convertido en métodos totalmente necesarios.

2.1.1 Ciclo de la vida de los datos

Actualmente, los datos se están convirtiendo en un facilitador clave para mejorar la competitividad de fabricación. Sin embargo, para que los datos sean útiles es necesario que se traduzcan en contenido y contexto de información concretos. El recorrido completo de recopilación, almacenamiento, preprocesamiento, análisis, minería, visualización, transmisión, y aplicación de los datos es lo que se considera el ciclo de vida de los datos [21].

1. Recopilación de datos. Da comienzo al proceso de análisis. Se recopilan y agregan los datos recogidos de diferentes fuentes, mayoritariamente de sensores. Tiene especial protagonismo el IoT, ya que permite medir, monitorear e informar de manera instantánea.
2. Almacenamiento de datos. Es necesario guardar de manera efectiva la gran cantidad de datos recopilados en los procesos de fabricación. Los diferentes tipos de datos se pueden clasificar en estructurados (tablas, símbolos, etc.), semiestructurados (árboles, gráficos, etc.) y datos no estructurados (audios, videos, imágenes, etc.). La computación en la nube facilita el almacenamiento de datos de manera rentable, eficiente y flexible.
3. Preprocesamiento de datos. Se realiza una serie de operaciones para descubrir el conocimiento de un gran volumen de datos. Es muy importante hacer una buena limpieza de los datos, basada en eliminar toda la información redundante, engañosa, duplicada e inconsistente.
4. Análisis de datos. Una vez se tienen los datos limpios y simplificados se exploran a través del análisis. La efectividad del análisis puede incrementar significativamente gracias al uso de técnicas estadísticas.

5. Aplicación del modelo. Una vez realizados los pasos anteriores se aplican los algoritmos de ML. Permite extraer conocimiento de la gran cantidad de datos inicial.
6. Visualización de datos. Permite a los usuarios comprender los datos de manera mucho más explícita, transmitiendo la información a través de medios gráficos (cuadros, diagramas, gráficas, etc.). Gracias al paso de visualización, los resultados se hacen más directos, fáciles de usar y accesibles.
7. Transmisión de datos. Los recursos de fabricación distribuidos se pueden integrar de manera efectiva casi en cualquier momento y lugar gracias a la fluidez continua de datos entre diferentes sistemas.
8. Aplicación de datos. La aplicación de métodos basados en datos hace posibles acciones de precaución, como por ejemplo el mantenimiento preventivo, la clasificación de fallos y la actualización automática.

2.1.2 Metodologías de Data Mining

Las empresas, con intención de explotar las nuevas tecnologías de análisis y aprendizaje para mejorar los procesos basados en datos, necesitan métodos probados capaces de seleccionarlos y analizarlos [9]. Han sido varias las metodologías de DM que han aparecido a lo largo de los años. A continuación, se presentan las cuatro más conocidas: CRISP-DM, KDD, SEMMA y VC-DM. Siendo CRISP-DM la metodología más usada actualmente para proyectos de DM y KDD el enfoque más citado a lo largo de la historia [8, 22].

Una vez revisada la descripción de las diferentes metodologías de DM que se expone a continuación, se puede observar que todas tienen en común la fase de modelado. En esta fase se aplica el algoritmo o algoritmos de ML seleccionado/s para automatizar el procesamiento de información y descubrir conocimiento.

2.1.2.1 CRISP-DM

CRISP-DM (Cross-Industry Standard Process for Data Mining) es el método de minería de datos más utilizado debido a su procedimiento gradual y su aplicabilidad general. El ciclo de vida de un proyecto de exploración de datos con esta metodología consta de seis pasos iterativos que se muestran en la Figura 5 y se explican brevemente a continuación [18, 23, 24].



Figura 5. Diagrama de la metodología de CRISP-DM. (Extraído de [22])

1. Análisis del problema. En primer lugar, se comprende el problema para obtener los objetivos del proyecto. Se diseña y define un proyecto basado en el análisis previo de la situación.
2. Análisis de datos. En esta fase, se recopilan datos y se familiariza con ellos para identificar posibles problemas de calidad de estos.
3. Preparación de datos. Contiene la selección, limpieza, construcción, integración y formateo de los datos de tal manera que queden totalmente preparados para el modelado.
4. Modelado. En la cuarta fase, se calibran los parámetros para obtener valores óptimos y para poder aplicar diferentes técnicas de modelado.
5. Evaluación. Se considera una fase imprescindible ya que se decide si el modelo está listo para ser aplicado. Se evalúa el rendimiento de este y de los pasos anteriores. En caso de no obtener buen rendimiento es necesario retroceder y reajustar el modelo.
6. Desarrollo. Esta última etapa consiste en la aplicación y el control del método, facilitando la obtención de conocimiento a partir de los datos.

2.1.2.2 KDD

El método KDD (Knowledge Discovery in Databases) se describe como el proceso de identificación de patrones válidos, potencialmente útiles y eventualmente comprensibles en los datos [25]. KDD es un proceso iterativo e interactivo que contiene cinco etapas [26].

1. Adquisición. En esta etapa inicial se recopilan los datos.
2. Preprocesamiento. En esta fase se hace un pretratamiento de los datos obtenidos con el objetivo de hacer una limpieza de estos.
3. Transformación. En esta tercera etapa se transforman los datos mediante métodos de reducción, con intención de dejarlos totalmente preparados para su posterior procesamiento.
4. Modelado. El objetivo de esta cuarta etapa es automatizar el proceso de búsqueda de patrones en los datos. Se aplican algoritmos de ML, utilizando técnicas de clasificación, agrupación, regresión y análisis entre las más comunes.
5. Interpretación. Para acabar, se evalúan los patrones encontrados.

2.1.2.3 SEMMA

SEMMA es otra metodología de DM muy frecuente. Se distinguen cinco etapas de descubrimiento de conocimiento. Estas cinco etapas son las siglas que dan nombre a este método (Sample, Explore, Modify, Model y Assess) y que se muestran esquematizadas en la Figura 6.

1. Muestreo. En primer lugar, se realiza la extracción del conjunto de datos sobre el que llevar a cabo el análisis.
2. Exploración. Seguidamente, se ejecuta un análisis en la muestra obtenida y se buscan relaciones entre los datos de dicha muestra. Se utilizan herramientas de visualización de datos o técnicas estadísticas para la exploración de la información. Este paso es muy importante para simplificar el problema y optimizar la eficiencia del modelo.
3. Modificación. En la tercera etapa, se prepara el conjunto de datos para ser aplicado con éxito en el modelo.

4. Modelado. A continuación, se crea un modelo de ML para descubrir conocimiento oculto.
5. Evaluación. Por último, se evalúan los resultados obtenidos observando la fiabilidad y calculando la eficiencia de los modelos.

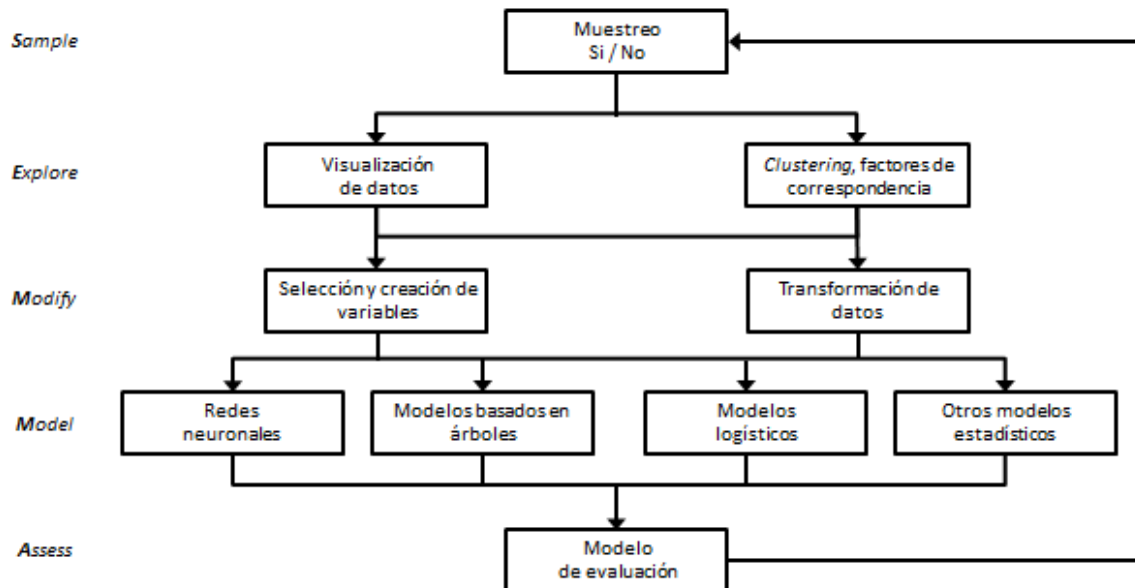


Figura 6. Dinámica general de la metodología SEMMA. (Extraído de [27])

2.1.2.4 VC-DM

El método VC-DM es la cuarta metodología de DM que actualmente se considera significativa a pesar de ser menos utilizada que las anteriores. Este método consta de cuatro etapas.

1. Identificación del problema. Esta etapa se lleva a cabo mediante la exploración de datos, donde se analiza la procedencia y frecuencia del problema.
2. Transformación de datos. En esta fase se preparan los datos para el modelado. También se selecciona el método de ML y se exploran los datos con el método elegido.
3. Actuación sobre la información. Se procesa toda la información para dar cuerpo al modelo.
4. Evaluación de resultados. En último lugar, a partir de un análisis se verifica la efectividad y aplicabilidad del modelo.

2.2 Clasificación de algoritmos de ML

El ML se ha convertido en las últimas décadas en un amplio y variado campo de investigación. Los algoritmos de ML pueden clasificarse según cómo se entrenan los modelos y la forma que tienen de aprender de los datos. Tal y como se muestra en la Figura 7, los algoritmos de ML se pueden clasificar en tres categorías, a saber, aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo. A su vez, pueden subclasificarse teniendo en cuenta la función que tienen a la hora de ser aplicados en: algoritmos de clasificación, regresión, clustering y de recompensa [13].

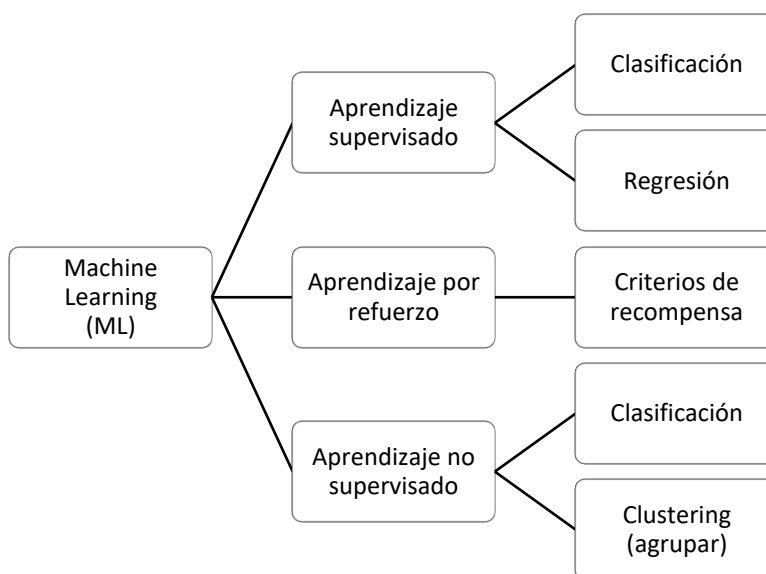


Figura 7. Clasificación de modelos de ML. (Fuente propia)

2.2.1 Aprendizaje automático supervisado

En el aprendizaje supervisado, el modelo se entrena con ejemplos de valores de entrada y de salida. Este proceso de aprendizaje se denomina supervisado porque cada dato de entrada tiene asociada una salida (etiqueta). Se basa en descubrir la relación existente entre las variables de entrada y las variables de salida de tal forma que el modelo pueda identificar cuando las predicciones sean falsas. Para entrenar el modelo, se necesita una gran muestra de datos de entrada y de salida y la supervisión de un humano para etiquetar los datos de entrada. Es por eso, que el proceso es posible que sea de coste elevado.

2.2.2 Aprendizaje automático no supervisado

En el aprendizaje no supervisado el sistema también aprende de los datos de entrada, sin embargo, no dispone de datos de salida. Este tipo de aprendizaje se desarrolla

creando patrones de similitud entre datos de entrada. La ventaja de este tipo de aprendizaje respecto el aprendizaje supervisado es que los datos de muestra pueden crearse con muy poco esfuerzo y no necesitas la supervisión de un humano para visualizar y etiquetar. Por ello, y en el contexto del Big Data, los métodos no supervisados tienen cada vez más relevancia [10].

2.2.3 Aprendizaje automático por refuerzo

El aprendizaje por refuerzo (Reinforcement Learning) ha llamado mucho la atención en los últimos años ya que los modelos bien entrenados han llegado a superar a los humanos en diferentes áreas [28]. La diferencia principal respecto los otros paradigmas del ML es que el RL no requiere de una muestra inicial para entrenar el modelo y que el agente recibe información interactuando continuamente con el entorno [29].

En el aprendizaje por refuerzo los algoritmos se entrenan por recompensas y penalizaciones. Generalmente, los métodos se basan en un sistema de puntuación, donde las recompensas se expresan como un aumento de puntos y las penalizaciones como una disminución. El objetivo de éstos es maximizar la puntuación. El RL es muy recomendado para problemas que incluyen una compensación de recompensa a largo plazo versus a corto plazo.

2.3 Elementos básicos de ML

Con el objetivo de conseguir modelos capaces de aprender de los datos, que no se limiten a memorizar y para proporcionar el buen funcionamiento de estos, es importante someterlos a un buen entrenamiento. Para lograrlo y para poder comparar el rendimiento obtenido entre algoritmos, a continuación, se explican una serie de elementos básicos a tener en cuenta a la hora de trabajar con las técnicas ML.

2.3.1 Overfitting y underfitting

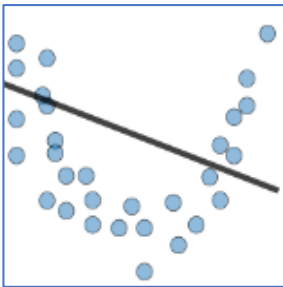
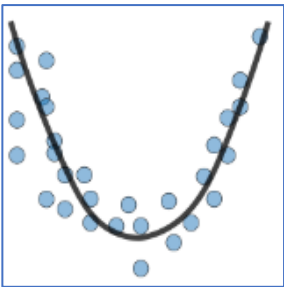
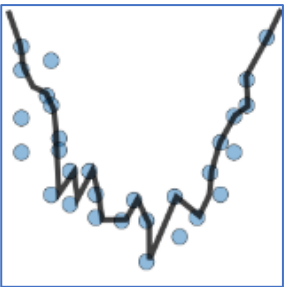
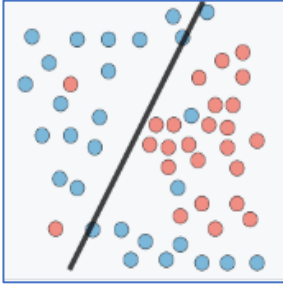
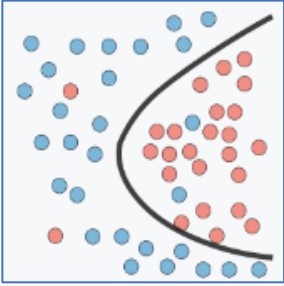
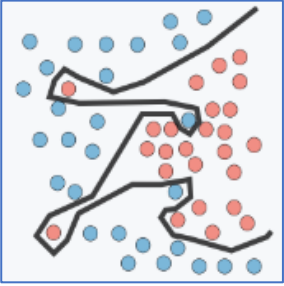
El overfitting (sobreajuste) y underfitting (falta de ajuste) son términos que se refieren a deficiencias que el modelo entrenado podría sufrir, causando como resultado, un rendimiento poco fiable cuando se aplica a nuevos conjuntos de datos. Es por eso, que se consideran uno de los puntos clave en el uso de algoritmos de ML.

Por un lado, se entiende como overfitting cuando el modelo aprende asignaciones incorrectas que funcionan bien en el conjunto de entrenamiento. Acostumbra a pasar con conjuntos de datos pequeños o con modelos que han sido entrenados demasiado tiempo. Por otro lado, se habla de underfitting cuando el modelo no puede capturar bien

las características de los conjuntos de entrenamiento. Acostumbra a pasar con modelos demasiado simples o poco entrenados [30]. Esto hace que en un modelo que sufre overfitting, generalmente se obtiene un sesgo bajo y varianza alta, en cambio, en un modelo que está poco ajustado la varianza acostumbra a ser baja y el sesgo alto [31]. Se entiende el concepto de sesgo como la diferencia entre el valor predicho y el valor observado (valor que tratamos de predecir) para determinados puntos de datos, y el concepto de varianza como la variabilidad de la predicción del modelo para puntos de datos dados [32].

El overfitting y el underfitting son problemas comunes en modelos de regresión y clasificación. En la Tabla 1 se representan ejemplos de estos:

Tabla 1. Ejemplos de underfitting y overfitting en regresión y clasificación. (Extraído de [32])

	Underfitting	Bien ajustado	Overfitting
Ejemplo de regresión			
Ejemplo de clasificación			

A medida que el algoritmo aprende, el error del modelo en los datos de entrenamiento y el error en el conjunto de datos de prueba disminuyen. Si el algoritmo se entrena durante demasiado tiempo, el error puede seguir disminuyendo hasta tal punto que empieza a sobre ajustarse, ya que empieza a aprender detalles irrelevantes y a perder la capacidad de generalizar. Es importante encontrar la tendencia del conjunto de datos que generalice bien el modelo, para ello es necesario detener el entrenamiento justo antes de que el error en el conjunto de datos de prueba empiece a aumentar. En la Figura 8 se ejemplifica la relación entre el tiempo de entrenamiento y el error.

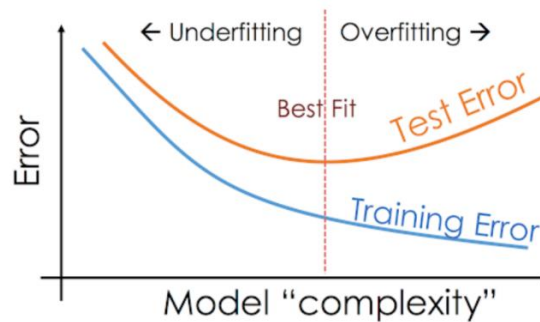


Figura 8. Punto óptimo de detención del entrenamiento. (Extraído de [33])

A continuación, se plantean dos tipos de entrenamiento basados en el conjunto de datos para poder prevenir los problemas de ajuste.

2.3.1.1 Entrenamiento tradicional

El entrenamiento tradicional de algoritmos de ML comprende tres fases:

1. Entrenamiento. El modelo se entrena con una parte de los datos del conjunto, generalmente con el 80% de estos.
2. Validación. El modelo se evalúa con los datos restantes, generalmente el 20% del conjunto.
3. Prueba. Se hace una predicción para probar el modelo con datos nunca vistos por el algoritmo.

Para llevar a cabo este entrenamiento es necesario disponer de una muestra suficientemente grande, de tal manera que sea posible extraer resultados significativos de los datos desde el punto de vista estadístico. Se aconseja que la división se haga de forma aleatoria, para así proporcionar la máxima variabilidad de datos en el conjunto de entrenamiento y validación.

En caso de no disponer de conjunto de datos de prueba, también es común utilizar la repartición 70-20-10, donde es usado este porcentaje de datos para las fases de entrenamiento, validación y prueba, respectivamente.

2.3.1.2 Cross Validation

Con intención de obtener mejores resultados en cuanto a la medición de calidad de la predicción de los modelos, se plantea el método de validación cruzada. Este método además de la fiabilidad que ofrece, también permite mucha flexibilidad al no necesitar tener disponible una gran cantidad de datos en el conjunto de entrenamiento.

2.3.1.2.1 K-fold:

Es el modelo de validación cruzada más utilizado. Este se basa en dividir el conjunto de datos de entrenamiento en k subconjuntos iguales. A la hora de hacer el entrenamiento del modelo, se utiliza cada subconjunto de datos para la validación y el resto de los datos ($k-1$) para el entrenamiento. El proceso se repite k veces, de forma que todos los grupos se hayan utilizado para validar el modelo. De cada iteración se obtiene un error. El error de validación cruzada es el promedio del error obtenido en los k conjuntos. En la Figura 9 se puede observar el proceso K-fold esquematizado.



Figura 9. Proceso esquematizado de validación cruzada k-fold. (Extraído de [32])

2.3.1.2.2 Leave-p-out:

El funcionamiento de este método es similar al anterior. En este caso se excluyen del entrenamiento p datos que serán usados para la validación del modelo. El proceso se repetirá tantas veces como observaciones disponibles haya, excluyendo en cada ocasión un conjunto p diferente. El error total será el promedio de los errores de todas las iteraciones.

La opción más común es que p sea igual a uno, en este caso el método se llama *leave-one-out*.

2.3.2 Métricas

Las métricas son la herramienta que permite comparar diferentes algoritmos para clasificación y regresión. Facilitan información sobre el rendimiento de los diferentes modelos y así permiten seleccionar el más rentable. A continuación, se introducen las métricas más utilizadas [34].

2.3.2.1 Métricas de clasificación

Se utiliza la matriz de confusión (Confusion matrix) para tener una visión completa al

evaluar el rendimiento de un modelo de clasificación. La matriz de confusión se define de la siguiente manera (Figura 10):

		Clase predicha	
		+	-
Clase real	+	TP True Positives	FN False Negatives Type II error
	-	FP False Positives Type I error	TN True Negatives

Figura 10. Representación de la matriz de confusión. (Extraído de [32])

Donde TP es cuando la clase real y la predicha son 1 (verdadero), TN es cuando la clase real y la predicha son 0 (falso), FP es cuando la clase real es 0 (falso) y la predicha es 1 (verdadero) y FN es cuando la clase real es 1 (verdadero) y la clase predicha es 0 (falso).

Las métricas que se utilizan para evaluar los modelos de clasificación y se definen en base a la clasificación realizada en la matriz de confusión son las siguientes:

2.3.2.1.1 Métrica de exactitud

Se utiliza para medir el rendimiento general del modelo. Esta métrica indica el número de elementos clasificados correctamente en comparación con el número total.

$$\frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

2.3.2.1.2 Métrica de precisión

Se utiliza para medir que tan precisas son las predicciones positivas. Esta métrica representa el número de verdaderos positivos que son realmente positivos, comparándolos con el número total de predicciones correctas.

$$\frac{TP}{TP + FN} \quad (2)$$

2.3.2.1.3 Métrica de exhaustividad/sensibilidad

Se utiliza para medir la cobertura de la muestra positiva real. Esta métrica muestra la cantidad de verdaderos positivos que el modelo ha clasificado en función del número

total de valores positivos.

$$\frac{TP}{TP + FN} \quad (3)$$

2.3.2.1.4 Métrica de Especificidad

Se utiliza para medir la cobertura de la muestra negativa real. Esta métrica muestra la cantidad de verdaderos negativos que el modelo ha clasificado en función del número total de valores negativos.

$$\frac{TN}{TN + FP} \quad (4)$$

2.3.2.1.5 Métrica de puntuación F1

Se utiliza para medir la métrica híbrida útil en clases desbalanceadas. Esta métrica es la combinación de las métricas de precisión y exhaustividad y sirve de compromiso entre ellas. La mejor puntuación F1 es igual a 1 y la peor a 0.

$$\frac{2TP}{2TP + FP + FN} \quad (5)$$

2.3.2.2 Métricas de regresión

En los modelos de regresión el objetivo es predecir o estimar el valor de una variable con la ayuda de las variables independientes relacionadas. A diferencia de los modelos de clasificación, en los de regresión se obtiene un valor continuo dentro del rango dado [35, 36].

Las formas más comunes para estimar el rendimiento y evaluar el ajuste del modelo son:

2.3.2.2.1 Error cuadrático medio (MSE)

Es una métrica muy utilizada en modelos de regresión ya que facilita la optimización. Esta métrica es el promedio de la diferencia al cuadrado entre el valor observado y valor predicho.

$$MSE = \frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2 \quad (6)$$

2.3.2.2.2 Error cuadrático medio de raíz (RMSE)

RMSE se considera la métrica más utilizada para modelos de regresión. Genera una alta penalización en errores grandes. Esta métrica representa la raíz cuadrada de la diferencia cuadrática promedio entre el valor observado y el valor predicho.

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2} \quad (7)$$

2.3.2.2.3 Error absoluto medio (MAE)

El error absoluto medio es una puntuación lineal que pondera todas las diferencias por igual. Por lo tanto, no es recomendable para predecir valores atípicos. Esta métrica es el error absoluto entre el valor observado y el valor predicho.

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j| \quad (8)$$

2.3.2.2.4 Coeficiente de determinación (R^2)

El coeficiente de determinación o R^2 se define como el cociente entre la variabilidad explicada por el modelo y la variabilidad total. Siempre tendrá por resultado un valor menor o igual a 1, indiferentemente del tamaño de los valores. De esta manera, es posible comparar el modelo y la evaluación del rendimiento de forma intuitiva y fácil.

$$R^2 = 1 - \frac{\sum_{j=1}^n (y_j - \hat{y}_j)^2}{\sum_{j=1}^n (y_j - \bar{y}_j)^2} \quad (9)$$

2.3.2.2.5 R^2 ajustado

Representa una mejora de la métrica explicada en el apartado anterior (R^2). R^2 ajustado siempre es menor que R^2 , ya que se ajusta a los predictores crecientes y solo muestra una mejora si hay una mejora real.

$$R_a^2 = 1 - \frac{(1 - R^2)(n - 1)}{n - k - 1} \quad (10)$$

donde n es el número de observaciones, k es el número de variables independientes y R_a^2 es R^2 ajustado.

2.4 Muestra de técnicas de ML

Son muchas las técnicas existentes de ML. Dado que el presente trabajo está centrado en la aplicación de estas técnicas en el ámbito industrial, a continuación, se hace una breve descripción de los algoritmos de ML más utilizados en este ámbito.

2.4.1 Regresión lineal simple

La regresión lineal simple es un método estadístico que trata de cuantificar la relación lineal que existe entre dos variables cuantitativas continuas $y = f(x)$. Por un lado, se encuentra la variable independiente o predictiva, representada por x , y por otro lado la variable dependiente o respuesta, representada por y [37]. Se puede apreciar un ejemplo de regresión lineal simple en la Figura 11.

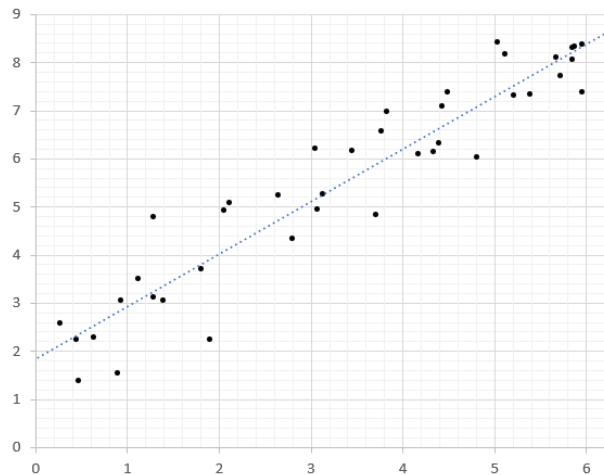


Figura 11. Ejemplo de modelo de regresión lineal simple. (Fuente propia)

La línea que mejor se ajusta viene definida por la función:

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad 1 \leq i \leq n \quad (11)$$

donde β_0 y β_1 son los parámetros del modelo de regresión. β_0 es la intersección con el eje OY, β_1 es la pendiente de la recta y ε_i es el término de error o residuo, la parte de y que no puede ser explicada por el modelo.

La predicción de y_i respecto la variable explicativa x_i se expresa por:

$$\hat{y}_i = \beta_0 + \beta_1 x_i \quad (12)$$

Al utilizar el modelo de regresión para predecir la variable respuesta y_i a partir de la variable explicativa x_i , se obtiene un error de predicción que viene definido por:

$$e = y_i - \hat{y}_i \quad (13)$$

La recta de regresión que mejor se ajusta a los datos muestrales será la que haga los errores de predicción más pequeños $\{(x_i, y_i)\}_{i=1}^n$. Para lograr este objetivo, una forma común es utilizar el criterio de mínimos cuadrados, que permite encontrar los valores de β_0 y β_1 de tal manera que minimicen la suma de los errores de predicción al cuadrado.

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \quad (14)$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (15)$$

Debido a que las fórmulas para β_0 y β_1 se derivan utilizando el criterio de mínimos cuadrados, la ecuación resultante 11 a menudo se conoce como recta de mínimos cuadrados.

Si el modelo de regresión simple es estadísticamente significativo, se puede utilizar para predecir valores de la variable respuesta y , en base a nuevos valores de la variable explicativa x .

2.4.2 Regresión lineal múltiple

La regresión lineal múltiple [Figura 12] es una extensión de la técnica explicada en el apartado anterior. Se utiliza para predecir valores de una variable respuesta y cuantitativa en base a múltiples variables explicativas x_1, x_2, \dots, x_n , siempre que el modelo sea estadísticamente significativo.

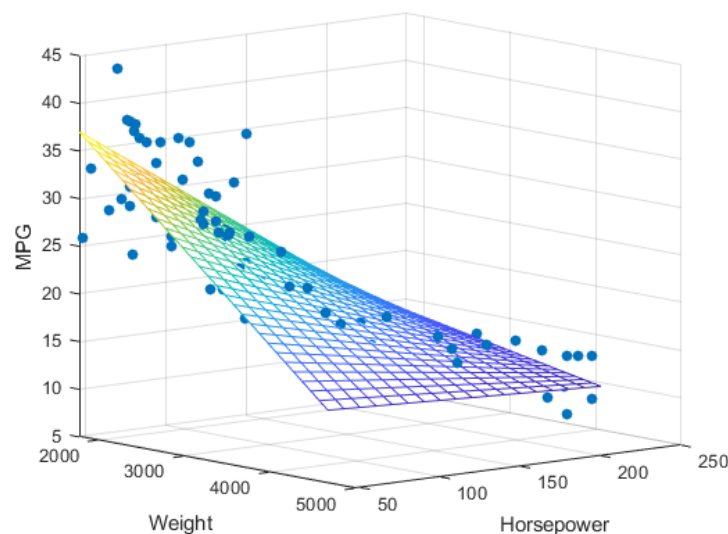


Figura 12. Ejemplo de modelo de regresión lineal múltiple. (Extraído de [38])

El modelo de regresión lineal múltiple tiene la forma:

$$Y_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip} + \varepsilon_i \quad 1 \leq i \leq n \quad (16)$$

donde $\beta_0, \beta_1, \beta_2, \dots, \beta_t$ son los coeficientes o parámetros del modelo de regresión, y ε_i es el término de error, la parte de y que no puede ser explicada por el modelo.

El coeficiente β_i mide la asociación de la variable respuesta y con la variable explicativa x_i . Este coeficiente puede ser interpretado como el efecto medio sobre y de un incremento en una unidad en la variable x_i manteniendo las otras variables explicativas fijas.

A raíz de la ecuación 16, se puede plantear el modelo de forma matricial como:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \dots & x_{1p} \\ 1 & x_{21} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \dots & x_{np} \end{pmatrix} \cdot \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix} \quad (17)$$

Asignando la notación a las matrices respectivas, se puede escribir la expresión anterior:

$$Y = X\beta + \varepsilon \quad (18)$$

Para estimar el vector de parámetros β se puede aplicar el método de mínimos cuadrados igual que en el modelo de regresión lineal simple. Como resultado se obtiene el siguiente estimador:

$$\hat{\beta} = (X^t X)^{-1} X^t Y \quad (19)$$

donde X^t es la matriz transpuesta de X .

2.4.3 Regresión polinómica

La regresión polinómica es un caso especial de regresión lineal que se usa cuando la relación entre la variable dependiente e independiente es curvilínea. La principal ventaja de la regresión polinómica se puede observar en la Figura 13, donde se compara un ejemplo de regresión lineal simple que no captura bien todos los datos con un ejemplo de regresión polinómica que se ajusta mucho mejor. Por lo tanto, este modelo permite minimizar el error de predicción [39, 40].

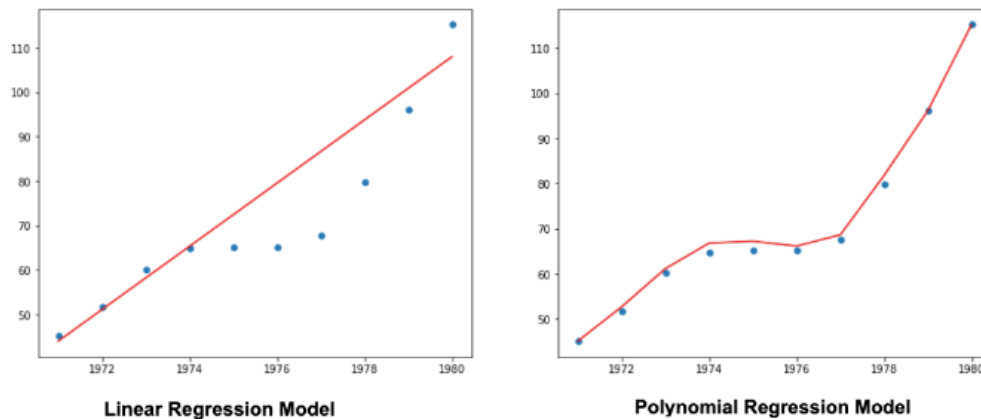


Figura 13. Comparación entre aplicación del modelo de regresión lineal y regresión polinómica. (Extraído de [39])

Para obtener una ecuación polinómica, se transforma la regresión lineal agregando potencia a la variable independiente. La ecuación general de los modelos polinomiales de orden k con una variable se representan como:

$$y = \beta_0 + \beta_1x + \beta_2x^2 + \dots + \beta_kx^k + \varepsilon \quad (20)$$

La regresión polinómica puede usarse para aproximar una relación no lineal compleja. Las técnicas que se usan para ajustar este modelo son las mismas que las usadas para ajustar el modelo de regresión lineal.

La principal desventaja de la regresión polinómica es que es muy sensible a valores atípicos, pudiendo ocasionar overfitting y en consecuencia afectaciones en el rendimiento del modelo.

2.4.4 Regresión logística (LR)

La regresión logística es un algoritmo de aprendizaje supervisado, similar a las técnicas explicadas en los apartados anteriores, que describe y estima la relación entre una variable categórica dependiente y una o múltiples variables independientes. Puede utilizarse para clasificar una observación en dos categorías (regresión logística binaria) o en múltiples categorías (regresión logística multinomial). En esta sección se presenta únicamente la primera por su simplicidad [41].

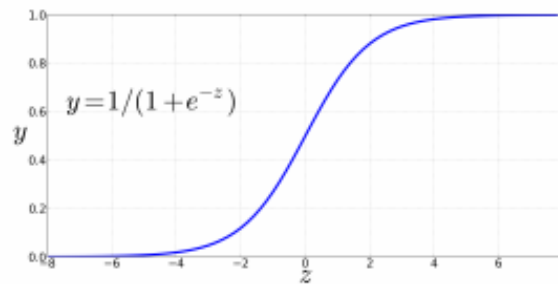


Figura 14. Modelo de regresión logística con función sigmoide. (Extraído de [41])

La regresión logística binaria es un clasificador probabilístico que requiere de un conjunto de datos de entrenamiento $(x^{(i)}, y^{(i)})$, donde cada observación de entrada $x^{(i)}$, será un vector de características $[x_1, x_2, \dots, x_n]$. El objetivo de la regresión logística binaria es entrenar a un clasificador que pueda tomar una decisión binaria sobre cada nueva observación de entrada. Dada una nueva observación de entrada $x = [x_1, x_2, \dots, x_n]$, la salida (y) del clasificador puede ser 1 (si la observación es un miembro de una determinada clase) o 0 (si la observación no lo es).

El algoritmo resuelve esta tarea aprendiendo los parámetros, vector de pesos w_i y término de sesgo b , del modelo a partir de un conjunto de entrenamiento. Cada peso w_i es un número real y está asociado con una de las características de entrada para la decisión de clasificación, mientras que el término de sesgo es otro número real que se agrega a las entradas ponderadas.

$$z = \left(\sum_{i=1}^n w_i x_i \right) + b \quad (20)$$

$$z = w \cdot x + b \quad (21)$$

Para forzar que la salida de clasificador sea una probabilidad, se pasa z a través de la función sigmoide (función logística), $\sigma(z)$. La función sigmoide tiene la siguiente ecuación:

$$y = \sigma(z) = \frac{1}{1 + e^{-z}} \quad (22)$$

Esta función toma un valor real y lo asigna al rango $[0, 1]$.

Para una nueva observación se calcula:

$$P(y = 1|x) = \sigma(w \cdot x + b) \quad (22)$$

Y la decisión se toma en base al valor que toma $P(y = 1|x)$

$$\hat{y} = \begin{cases} 1 & \text{si } P(y = 1|x) > 0,5 \\ 0 & \text{si } P(y = 1|x) \leq 0,5 \end{cases} \quad (23)$$

La estimación de los parámetros del modelo de regresión logística se realiza utilizando el método de máxima verosimilitud. Este método consiste en elegir aquellos valores de los parámetros que hacen máxima la probabilidad de obtener las etiquetas asociadas a cada observación de entrada del conjunto de entrenamiento.

2.4.5 Árbol de decisión (DT)

Árboles de decisión es un algoritmo de aprendizaje supervisado que puede ser utilizado tanto para clasificación como para regresión. Se trata de un gráfico en forma de árbol que sirve para ayudar en la toma de decisiones. Este árbol, tal y como se muestra en la Figura 15, contiene diferentes partes: un nodo raíz (punto de partida del algoritmo, desde el que el conjunto de datos se divide en subconjuntos); nodos intermedios (cuanto más profundos se encuentren en el árbol, será más homogéneo el subconjunto); nodos hoja (se considera el final del árbol de decisión); y ramas que conectan los diferentes nodos [42].

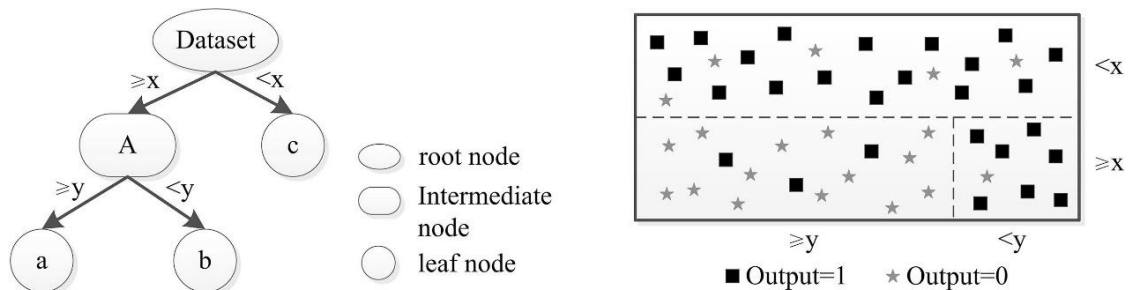


Figura 15. Partes y funcionamiento básico del algoritmo DT. (Extraído de [42])

Para intentar evitar el sobreajuste en estos modelos, se aplica la poda. Se trata de un mecanismo que, como se muestra en la Figura 15, consiste en cortar las ramas que no optimizan la capacidad de generalización del árbol. Partiendo del nodo raíz, los datos se dividen de forma recursiva en subconjuntos y en cada paso se determina la mejor división en base a un criterio [43]. Los dos criterios más utilizados son el índice de Gini y la entropía.

1) Índice de Gini:

$$G(E) = 1 - \sum_{j=1}^c p_j^2 \quad (24)$$

2) Entropía:

$$H(E) = - \sum_{j=1}^c \log p_j \quad (25)$$

2.4.6 Bosque aleatorio (RF)

Bosque aleatorio es un algoritmo de aprendizaje supervisado que está formado por varios árboles de decisión, por ese motivo, RF también puede utilizarse tanto para clasificación como para regresión. Es un método de decisión muy rápido y que opera con grandes cantidades de datos [43].

Se selecciona un subconjunto con k variables de influencia para cada nodo de cada árbol de decisión. Estas variables seleccionadas son diferentes en cada DT, por lo que se garantiza la capacidad de generalización. El resultado final del modelo se calcula dependiendo de si se está utilizando para clasificación o regresión. Los resultados de modelos de clasificación suelen combinarse utilizando “soft voting” o “majority voting”, en el primero se da más importancia a los resultados de los árboles que están muy seguros y en el segundo se tiene en cuenta la mayoría. Para los resultados en modelos de regresión, lo más habitual es combinar los resultados haciendo el promedio de las predicciones de los resultados individuales. Este proceso de combinación de resultados permite que RF sea un método muy resistente al sobreajuste [42].

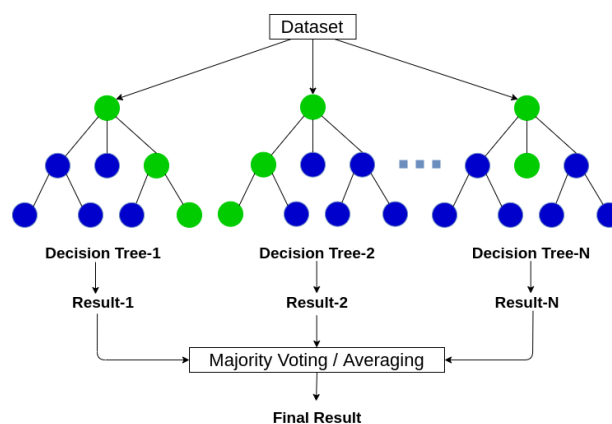


Figura 16. Modelo de bosque aleatorio. (Extraído de [44])

2.4.7 Redes neuronales (NN)

Las redes neuronales (NN) se han convertido en la familia de algoritmos de ML más populares. Se pueden explicar como un modelo extremadamente simplificado del cerebro que comúnmente se utilizan para resolver problemas de clasificación (reconocimiento de patrones, extracción de características, coincidencia de imágenes) y predicción [45].

Las redes neuronales tienen por elemento principal de construcción un número determinado de neuronas. Una neurona [Figura 17] es una unidad básica de procesamiento que recibe una suma ponderada de sus entradas, la procesa a través de una función de activación y produce una salida. La salida de la neurona se puede conectar a las entradas de otras neuronas mediante conexiones ponderadas, llamadas sinapsis. Una red neuronal consiste en un conjunto de neuronas conectadas de un modo concreto [12].

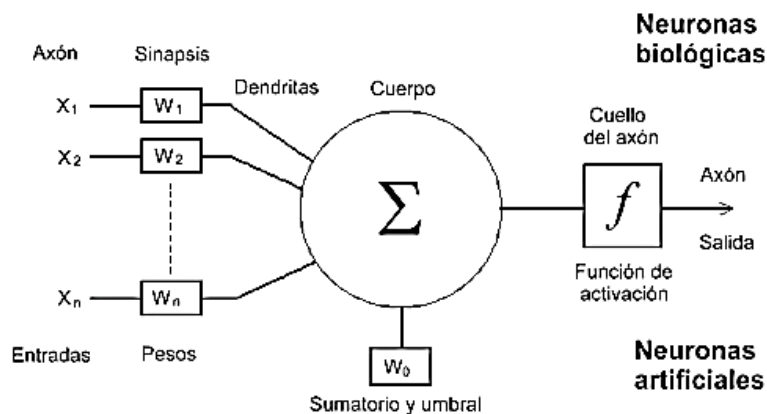


Figura 17. Modelo de una neurona. (Extraído de [46])

El modelo típico de NN tiene tres capas con diferentes neuronas. La capa de entrada, que recibe los datos; la capa oculta, que calcula la entrada de cada neurona a través de una suma ponderada y una función de activación adecuada; y la capa de salida, que produce la respuesta de la red [47]. Matemáticamente las ANN se representan como:

$$y = f(x) = \sum_{j=1}^J w_j \delta \left(\sum_{i=1}^I w_{ij} x_i + b_j \right) + \beta + \varepsilon \quad (26)$$

donde x es el vector de entrada de dimensión I , w_{ij} es el factor de peso que conecta la neurona de entrada i con la neurona oculta j , w_j es el factor de peso que conecta la neurona oculta j con la neurona de salida, β es el sesgo para la neurona de salida, ε es

un error aleatorio y $\delta()$ es la función de activación encargada de devolver una salida a partir de una entrada.

Las NN se incluyen dentro de la categoría de métodos de aprendizaje supervisado y se entrenan utilizando el algoritmo de aprendizaje de Back-Propagation (BP). Este algoritmo se utiliza para actualizar los pesos de la sinapsis de la red y así minimizar el error de la NN. Consta de dos pasos: la propagación directa de la señal (forward-propagation) y la retro-propagación del error (back-propagation).

En un proceso de entrenamiento, las variables de entrada multiplicadas por pesos son transferidas a través de las capas ocultas hacia la capa de salida (forward-propagation). La diferencia, entre los valores observados de la variable respuesta y las predicciones, se pasa a la función de pérdida para evaluar lo bien que la red se está ajustando a su objetivo. A continuación, los errores se retro-propagan para actualizar cada uno de los pesos de las sinapsis de la red, de tal manera que puedan producir salidas más cercanas al valor real. La iteración continúa hasta que el modelo de red predice una salida satisfactoria [48].

La salida prevista $(\hat{y}_i)_k$ de la red neuronal de back-propagation se representa como [49]:

$$(\hat{y}_i)_k = \sigma^{out} \left(\sum_{i=1}^{n_H} w_j^{out} \cdot x_i^H + b_j^{out} \right), \quad k = 1, 2, \dots, l, \quad (27)$$

donde σ^{out} es la función de activación de la capa de salida, w_j^{out} y b_j^{out} son respectivamente los pesos y el sesgo de la capa de salida.

2.4.7.1 Algoritmo Back-Propagation

El algoritmo BP usa el concepto básico de moverse en la dirección del gradiente negativo de la función objetivo (método del descenso del gradiente). Parte de un punto al azar y lleva a terminar en la parte inferior o donde el gradiente es 0 [50]. De esta manera, los parámetros de entrenamiento (w y b) se actualizan capa por capa mediante descenso de gradiente. El gradiente de error se propaga hacia atrás desde la capa de salida hasta la capa de entrada. Al usar el método de descenso de gradiente se debe decidir qué tan lejos se moverá el gradiente negativo en cada paso, lo que se llama longitud del paso. Una forma es mover una cantidad proporcional al gradiente. A continuación, se presenta la fórmula para el método de descenso gradual:

$$W_{n+1} = W_n - \lambda \nabla f(W_n) \quad (28)$$

Donde W_n es el vector actual de pesos, λ es la tasa de aprendizaje y ∇f es el gradiente de la función objetivo a minimizar.

2.4.8 Redes neuronales convolucionales (CNN)

Las redes neuronales convolucionales (CNN) son una categoría de las NN que se utilizan para procesar imágenes de manera eficiente [51]. En una CNN se toma como variable de entrada una imagen, representada por una matriz de valores de píxeles, donde el valor de cada píxel en la matriz oscila entre 0 y 255 (0 indica negro y 255 blanco) [52].

Hay cuatro operaciones básicas en una CNN. Estas cuatro operaciones son la convolución, la no linealidad (ReLU), el pooling (agrupación/submuestreo) y la clasificación (capa completamente conectada). Las capas de convolución y pooling actúan como extractores de características de la imagen de entrada y la capa completamente conectada actúa como un clasificador.

1. Convolución:

La convolución consiste en deslizar una matriz filtro (Kernel) por la imagen de entrada a pasos de longitud k. Para cada posición, se calcula la multiplicación por elementos entre las dos matrices y se suman los resultados de las multiplicaciones. Esto sirve para obtener el entero final que constituye un solo elemento de la matriz de salida (mapa de características)

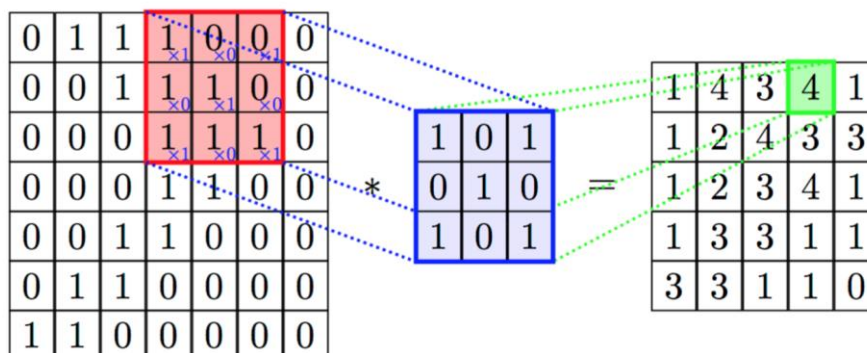


Figura 18. Proceso de convolución de una imagen con un filtro. (Extraído de [53])

Los filtros actúan como detectores de características de la imagen de entrada, por lo que diferentes filtros detectarán diferentes características. Es por eso que cuantos más filtros se apliquen, más características se extraen y, en consecuencia, se tendrá una red

mejor preparada para reconocer patrones en nuevas imágenes.

Hay parámetros que es necesario especificar antes del proceso de entrenamiento, cómo por ejemplo el número de filtros (Depth), el tamaño del filtro, la longitud de paso (Stride) o la arquitectura de la red, entre otros.

2. No linealidad:

Después de cada operación de convolución se utiliza una operación no lineal. Una de las más utilizadas es la Unidad Lineal Rectificada (ReLU).

ReLU es una operación basado en elementos que reemplaza todos los valores de píxeles negativos en el mapa de característica por cero ($output = \max(0, input)$), con el propósito de introducir la no linealidad en la CNN. El mapa de característica resultante al aplicar esta operación se conoce como mapa de características rectificado.

3. Pooling (agrupación/submuestreo)

El pooling es un proceso de discretización basado en muestras. Esta operación es la encargada de reducir la dimensionalidad de cada mapa de características rectificado reteniendo la información más importante. Cuando se aplica una capa de agrupación (pooling), se disminuye la muestra de la capa anterior, deslizando una ventana fija a través de una capa y eligiendo un valor que represente a todas las unidades capturadas por la ventana.

Se puede distinguir entre diferentes tipos de agrupación. Los más comunes son max-pooling y average-pooling. Los dos tipos tienen el mismo funcionamiento, pero en el primer caso a la nueva matriz se le asigna el valor más alto y en el segundo caso se le asigna el valor promedio [54].

4. Clasificación (capa totalmente conectada)

El término “totalmente conectado” implica que cada neurona en la capa anterior está conectada a cada neurona en la capa siguiente.

La salida de las capas convolucionales y de pooling representan características de alto nivel de la imagen de entrada. El propósito de esta capa es usar estas características para clasificar la imagen de entrada en varias clases según el conjunto de datos de entrenamiento. La salida de esta capa será un vector de probabilidades, significando esto una probabilidad por cada una de las clases.

El entrenamiento de la CNN empieza inicializando los filtros y parámetros con valores aleatorios. A continuación, la red toma una imagen de entrada y realiza la convolución, ReLU y las operaciones de agrupación junto con la propagación hacia adelante en la capa totalmente conectada. Esto hace que encuentre las probabilidades de salida para cada clase. En tercer lugar, calcula el error en la capa de salida. El siguiente paso consiste en utilizar el algoritmo BP (explicado en el apartado 2.4.7.1) que calcula los gradientes para actualizar todos los pesos del filtro y los valores de los parámetros para minimizar el error de salida. Y, por último, se repiten los pasos dos, tres y cuatro con todas las imágenes del conjunto de datos.

Al ingresar una nueva imagen en la CNN, la red pasa por el paso de propagación hacia adelante y genera una probabilidad para cada clase (las probabilidades de salida se calculan utilizando los pesos que se han optimizado para clasificar correctamente los ejemplos de entrenamiento anteriores). Si el conjunto de entrenamiento es suficientemente grande, la red generalizará bien las nuevas imágenes y las clasificará en la categoría correspondiente.

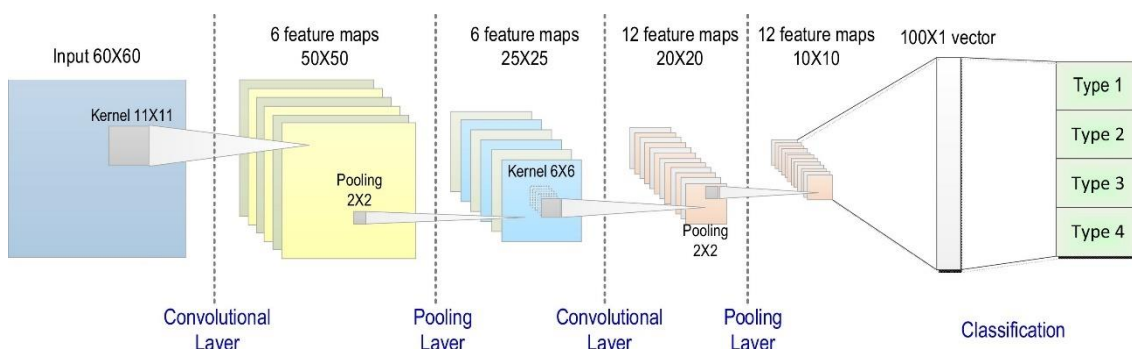


Figura 19. Arquitectura de dos etapas de una DCNN. (Extraído de [55])

2.4.9 Autoencoders (AE)

Un autoencoder es un tipo de red neuronal utilizada para aprender codificaciones eficientes de modo no supervisado. Se aplica en la reducción de dimensionalidad, comprensión, cifrado, etc.

Tal y como se muestra en la Figura 20, el autoencoder está formado por dos partes. La primera parte es la red codificadora y es la encargada de transformar los datos de entrada de un espacio de alta dimensión en códigos a un espacio de baja dimensión. La segunda parte, la decodificadora, reconstruye las entradas de los códigos correspondientes.

Es posible disponer de una serie de autoencoders que permiten el preentrenamiento de

una red neuronal profunda de varias capas ocultas [56].

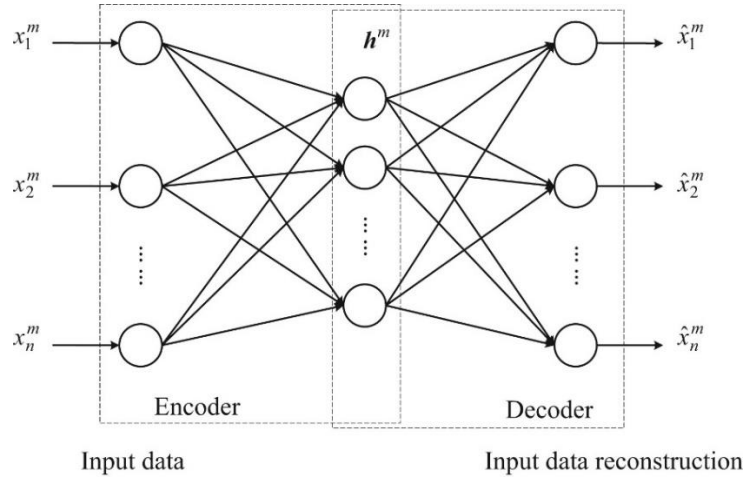


Figura 20. Arquitectura de un autoencoder. (Extraído de [56])

El vector de codificación obtenido representado por h^m , se define para cada señal de entrada (x^m) como:

$$h^m = f_{\theta}(x^m) \quad (29)$$

donde f_{θ} es la función de codificación.

La red de decodificación se define como una función de reconstrucción denotada por $g_{\theta'}$. Produce una reconstrucción llevando h^m desde el espacio de baja dimensión de vuelta al espacio de alta dimensión:

$$\hat{x}^m = g_{\theta'}(h^m) \quad (30)$$

Los parámetros del codificador y del decodificador se aprenden simultáneamente en la tarea de reconstruir lo mejor posible la entrada original, intentando incurrir en el error de reconstrucción más bajo posible. $L(x, \hat{x})$ es una función de pérdida que mide la discrepancia entre la entrada original (x) y la entrada reconstruida (\hat{x}) sobre M ejemplos entrenados.

El entrenamiento del autoencoder tiene por objetivo encontrar conjuntos de parámetros θ minimizando el error de reconstrucción:

$$\Phi_{AE}(\theta, \theta') = \frac{1}{M} \sum_{m=1}^M L(x^m, g_{\theta'}(f_{\theta}(x^m))) \quad (31)$$

Las formas comúnmente utilizadas para el codificador y el decodificador son

asignaciones afines:

$$f_{\theta}(x) = s_f(Wx + b) \quad (32)$$

$$g_{\theta'}(x) = s_g(W^T x + d) \quad (33)$$

$$L(x, \hat{x}) = \|x - \hat{x}\|^2 \quad (34)$$

donde s_f y s_g son las funciones de activación de codificador y decodificador, respectivamente. Por lo tanto, los conjuntos de parámetros del autoencoder son $\theta = \{W, b\}$ y $\theta' = \{W^T, d\}$, donde b y d son vectores de sesgo, y W y W^T son las matrices de peso.

2.4.10 Máquina de vectores de soporte (SVM)

La máquina de vectores de soporte (SVM) es un algoritmo relativamente novedoso que se ha convertido en una herramienta eficaz en tareas de modelado y predicción. Por ello SVM es una de las técnicas de aprendizaje supervisado más utilizada [57].

En la máquina de vectores de soporte se genera un hiperplano de separación que maximiza el margen entre dos conjuntos de datos de acuerdo con sus clases. Se establecen dos hiperplanos paralelos a cada lado del hiperplano de separación para crear el margen. Conseguir el mayor margen posible es desencadenante de lograr una solución óptima. Tal y como se indica en la Figura 21, las muestras de entrenamiento que están muy cerca del límite de decisión se denominan vectores de soporte.

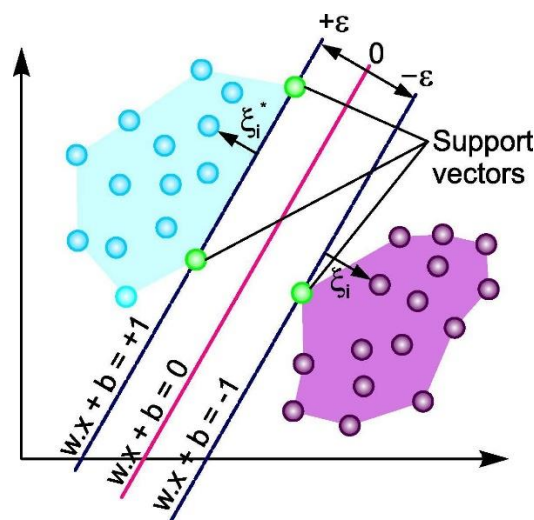


Figura 21. Ilustración gráfica de SVM. (Extraído de [57])

Los hiperplanos positivos y negativos [Figura 21] se representan matemáticamente por

la expresión [58]:

$$w^t x_{pos} + b = 1 \quad (35)$$

$$w^t x_{neg} + b = -1 \quad (36)$$

donde w^t son los pesos, x_{pos} y x_{neg} son las características en el lado derecho e izquierdo del plano de separación respectivamente y b es el vector de sesgo. A partir de las cuales, restando la ecuación 36 a la ecuación 35 y usando el vector w para normalizar, se obtiene:

$$\frac{w^t(x_{pos} - x_{neg})}{\|w\|} = \frac{2}{\|w\|} \quad (37)$$

$$\|w\| = \sqrt{\sum_{j=1}^{norte} w_j^2} \quad (38)$$

donde $\|w\|$ es la norma del vector para el hiperplano. El lado izquierdo de la ecuación 37 se interpreta como el margen y es lo que se debe maximizar. La posibilidad de maximizar este margen se da con la restricción de que las muestras se clasifiquen correctamente [58].

El clasificador de margen óptimo se define como:

$$h(x) = \text{sign}(w^T x - b) \quad (39)$$

donde $(w, b) \in R^n \times R$ es la solución del problema de optimización.

Cuando los datos de entrenamiento no pueden ser separados linealmente, se efectúa una transformación sobre los datos, de forma que se pueda encontrar un hiperplano que clasifique las categorías. La función matemática utilizada para la transformación se conoce como función núcleo (Kernel). Los núcleos son funciones que toman un espacio de entrada y lo transforman en un espacio de dimensión más alta para que el problema en consideración se vuelva linealmente separable.

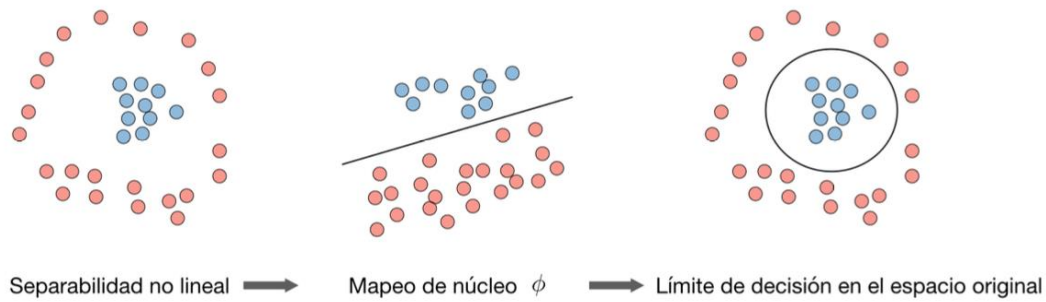


Figura 22. Representación de la transformación sobre los datos. (Extraído de [32])

Dado un mapeo de características ϕ , se define el núcleo K como:

$$K(x^{(i)}, x^{(j)}) = \phi(x^{(i)})^T \phi(x^{(j)}) \quad (40)$$

donde $(x^{(i)}, x^{(j)})$ son un par de muestras o características. El núcleo k definido por:

$$K(x^{(i)}, x^{(j)}) = \exp \left[- \left(\frac{\|x^{(i)} - x^{(j)}\|}{2\sigma^2} \right)^2 \right] \quad (41)$$

donde σ es la desviación estándar de las características, se conoce como núcleo Gaussiano y es frecuentemente utilizado.

2.4.11 K-Means

El Algoritmo K-means es uno de los algoritmos de aprendizaje no supervisado más usado para agrupar muestras de datos debido a su sencillez y rapidez [50]. El algoritmo agrupa objetos en k grupos basándose en sus características. Este agrupamiento se lleva a cabo minimizando la suma de distancias entre cada objeto (x_j) y el centroide (μ_i) de su grupo [59].

$$\min_S \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|^2 \quad (42)$$

donde S es el conjunto de datos.

El proceso del algoritmo empieza con la elección del número de grupos (k). En segundo lugar, se establecen centroides en el espacio de datos, generalmente de manera aleatoria. Posteriormente, cada objeto se asigna al centroide más cercano. El último paso consiste en actualizar los centroides de cada grupo, tomando como nuevo centroide la posición promedio de los objetos de cada grupo.

$$\mu_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \quad (43)$$

Estos dos últimos pasos se repiten hasta que los centroides se estabilizan (no se mueven o se mueven por debajo de una distancia umbral en cada paso).

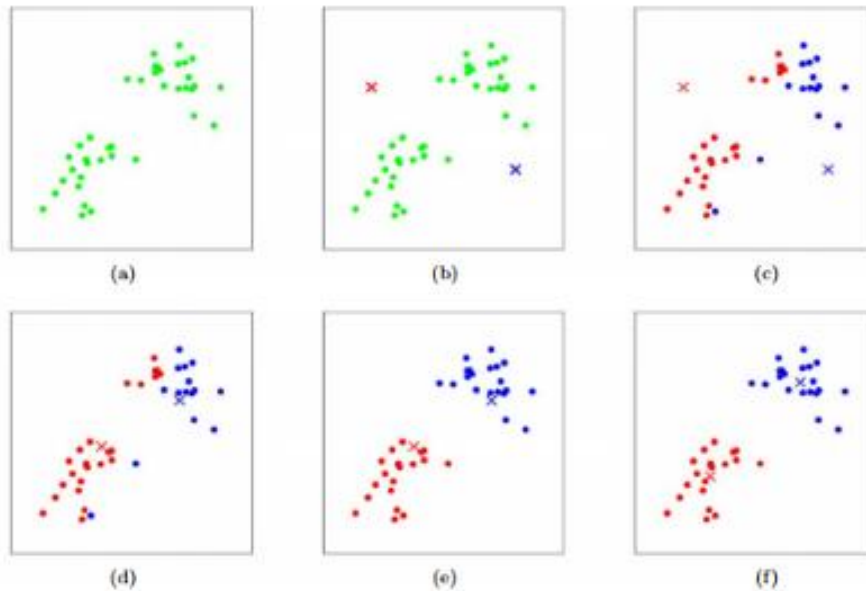


Figura 23. Evolución del modelo K-means hasta que se estabilizan los centroides. (Extraído de [60])

Para realizar el cálculo de la distancia se pueden emplear diferentes métodos, como por ejemplo [61]:

- 1) Distancia Euclidiana: corresponde a la longitud del camino más corto entre dos objetos. Es la distancia más utilizada.

$$d(a, b) = \sqrt{(x_{a1} - x_{b1})^2 + (x_{a2} - x_{b2})^2 + \dots + (x_{an} - x_{bn})^2} \quad (44)$$

- 2) Distancia de Manhattan: corresponde a la suma de distancias a lo largo de cada dimensión. Es menos sensible a valores atípicos.

$$d(a, b) = |x_{a1} - x_{b1}| + |x_{a2} - x_{b2}| + \dots + |x_{an} - x_{bn}| \quad (45)$$

2.4.12 K-Nearest-Neighbor (KNN)

KNN es un algoritmo de ML de aprendizaje supervisado que se utiliza, principalmente, para clasificar ítems. K es la cantidad de puntos que se tienen en cuenta en las cercanías del nuevo valor para clasificarlo. Por lo tanto, si la mayoría de los k vecinos más

cercanos (conjunto de entrenamiento) pertenecen a un grupo, el valor nuevo también pertenecerá a ese grupo, compartiendo las características de los valores de este. [61].

Dada la sencillez de este algoritmo, es ampliamente utilizado para clasificación en diagnóstico de fallos, reconocimiento de patrones y tecnología de visión por computadora. También puede ser usado para predecir valores continuos (regresión). En este caso, el valor que se le asigna al nuevo ítem es el promedio de los k vecinos más cercanos.

El proceso del KNN empieza con el cálculo de la distancia entre el valor a clasificar y el resto de los valores del conjunto de entrenamiento. A continuación, se seleccionan los k elementos más cercanos y, por último, se asigna a este un nuevo valor con la etiqueta de los valores más cercanos entre los k vecinos más cercanos.

En el ejemplo que se muestra en la Figura 24, se puede apreciar la importancia de la elección del valor k , ya que, en función de esta, el nuevo valor pertenecerá a un grupo u otro.

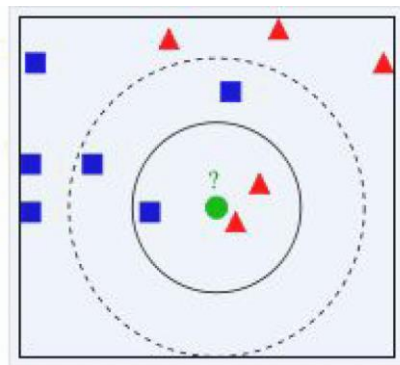


Figura 24. Proceso de toma de decisiones del algoritmo KNN. (Extraído de [61])

Los métodos más populares de medir la cercanía entre valores son (1) la distancia Euclidiana o (2) la distancia Manhattan, explicados y definidos en el apartado anterior.

2.4.13 Reinforcement Learning (RL)

El aprendizaje por refuerzo es uno de los paradigmas de ML y representa una técnica para resolver problemas en los que se realiza una toma de decisiones de forma secuencial. RL asume que un agente (el sujeto que toma decisiones) interactúa con el entorno tal y como se observa en la Figura 25 [28].

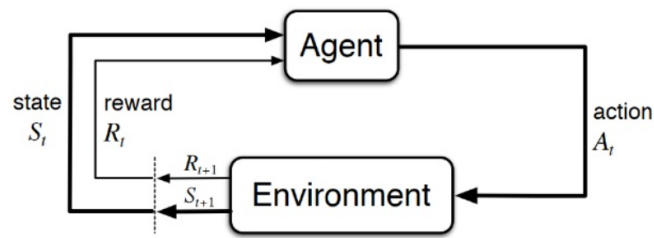


Figura 25. Estructura de aprendizaje por refuerzo. (Extraído de [28])

Las interacciones ocurren secuencialmente en el tiempo $t = 0, 1, 2, \dots, t$. En cada instante de tiempo, el agente obtiene una representación del estado del entorno S_t y selecciona una acción a tomar A_t . Como resultado de esa acción el entorno pasa a un nuevo estado S_{t+1} y el agente recibe una recompensa R_{t+1} en base a la acción previa, que le indicará la bondad de la acción llevada a cabo. El agente interactúa con el entorno y aprende en base a ensayo y error.

Los procesos de decisión de Markov nos dan un modo de formalizar la toma secuencial de decisiones. Un proceso de decisión de Markov (MDP) se define por una 5-tupla (S, A, P, R, γ) donde S es un conjunto de estados. En cada estado se emprende una acción A_t , del conjunto de acciones A , y al realizar la acción A_t se pasa del estado S_t a S_{t+1} , con probabilidad de transición $P(S_{t+1}/A_t, S_t)$. Esta última acción genera una recompensa R_{t+1} . El factor de descuento viene definido por $\gamma \in [0, 1]$, y controla la importancia de la recompensa actual en entornos futuros [29].

En los MDP se considera que el problema es dependiente del tiempo ya que la acción influye en los estados de futuro. También se aprecia que el nuevo estado no es determinista y solo puede predecirse con una cierta probabilidad. Esta sección está enfocada a los algoritmos RL libres de modelo (la distribución de probabilidades de transición P no es conocida), y en tal caso, el agente aprenderá las transiciones de estado y recompensas esperadas interactuando con el entorno.

2.4.13.1 Método de Q-Learning:

1. Inicializa la tabla $Q(s, a)$ a cero para todos los pares (s, a)
2. Observa el estado actual s
3. Repite la secuencia
 - a. Selecciona una acción a y la aplica
 - b. Recibe la recompensa inmediata $R(s')$

- c. Observa el nuevo estado s'
- d. Actualiza la entrada de la tabla para $Q(s, a)$ de acuerdo a

$$Q(s, a) \leftarrow Q(s, a) + \alpha[R(s') + \gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (46)$$

- e. Se mueve al próximo estado i. e. $s \leftarrow s'$

Donde $\gamma \in [0,1]$ es un factor de descuento para equilibrar la recompensa a corto plazo y la recompensa a largo plazo y $\alpha \in (0,1]$ es la tasa de aprendizaje. Una vez actualizada la tabla $Q(s, a)$, se tendrá información de cuál es la acción óptima a realizar desde un estado.

El método Q-learning aprende una política óptima, mientras se elige la acción siguiente, en base a la heurística de ϵ -greedy, que consiste en:

2.4.13.2 Selección ϵ -greedy:

Seleccionar un valor $0 \leq \epsilon \leq 1$

1. Con probabilidad ϵ , intentar una acción aleatoria.
2. Con probabilidad $1 - \epsilon$, elegir la mejor acción actual.

Que significa que el agente debe descubrir el comportamiento óptimo mediante:

1. Exploración: intentar nuevas acciones para aprender sus recompensas y así ganar un mejor entendimiento del entorno (elegir una acción aleatoria).
2. Explotación: utilizar el conocimiento actual (tabla $Q(s, a)$), aunque este podría no ser lo óptimo.

3. ESTUDIOS DE CASO

Son muchos los casos en los que actualmente se están aplicando con éxito técnicas de aprendizaje automático con el objetivo de mejorar los procesos. El mantenimiento, que dependiendo del ámbito puede significar entre el 15% y el 40% de los costes de los bienes producidos, tiene impactos significativos en la rentabilidad de la industria. Por ese motivo, se ha convertido en foco de multitud de estudios automatizar la condición de la máquina y el diagnóstico de fallos, para así, poder reducir potencialmente los costes de mantenimiento y mejorar la productividad [62].

Las técnicas de ML se aplican para obtener diferentes tipos de beneficios. Entre los objetivos más comunes se encuentran el diagnóstico y la clasificación de fallos, la predicción y pronóstico de defectos y la optimización de sistemas.

Este capítulo se divide en cuatro apartados. Por un lado, los tres primeros llevan por título los objetivos mencionados en el párrafo anterior, con la intención de recoger ejemplos donde se están aplicando técnicas de ML con éxito. Cada capítulo se abre con una revisión de uno de los estudios, donde se analiza detalladamente el proceso de aplicación del algoritmo y las mejoras que proporciona. A continuación, el apartado se completa con una tabla que recoge diferentes casos. Esta tabla está ordenada cronológicamente y en ella se puede visualizar el autor, el algoritmo protagonista y un breve resumen sobre la implementación de la técnica y los resultados que proporciona. Y, por otro lado, el cuarto apartado hace referencia al software utilizado para el procesamiento de algoritmos, donde se recogen los programas más utilizados en los artículos revisados en este trabajo.

3.1 Diagnóstico y clasificación de fallos

En la industria, la detección temprana y clasificación de fallos ocupa cada vez más protagonismo. Tener la capacidad de detectar fallos pronto y saber el motivo por el que se ha ocasionado dicho fallo puede evitar grandes pérdidas y largos tiempos de inactividad.

3.1.1 Caso de uso analizado

El artículo “Deep neural networks: a promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data” publicado el año 2016 por la editorial Elsevier, tiene como objetivo aplicar redes neuronales profundas (DNN). El

objetivo de este estudio es procesar rápidamente datos de fallos masivos y proporcionar resultados precisos de diagnóstico en maquinaria rotativa [56].

Se plantea establecer una red de arquitectura profunda para poder superar las deficiencias que tienen los anteriores métodos propuestos en diagnóstico inteligente de fallos, por lo que se espera obtener mayor precisión de diagnóstico comparado con métodos basados en ANN. Las DNN se utilizan para implementar la extracción de características de fallos y el diagnóstico inteligente.

Estas DNN se entrenan previamente mediante un aprendizaje capa por capa sin supervisión y luego se ajustan con un algoritmo supervisado. El nuevo método inteligente de diagnóstico de fallos extrae de manera adaptativa las características de fallos de las señales sin procesar de la maquinaria rotativa y clasifica automáticamente las condiciones de salud de la máquina con estas características de fallos. El modelo que se propone incluye cuatro procedimientos.

1. Obtener los espectros de frecuencia de maquinaria rotativa en diferentes condiciones de funcionamiento. Estos espectros comprenden el conjunto de entrenamiento $\{x^i, d^i\}_{i=1}^M$, donde x^i es el i -ésimo espectro de frecuencia a entrenar, d^i es la condición de buen funcionamiento y M es el número de espectros de frecuencia.
2. Crear una DNN en la cual el número de unidades de entrada sea la dimensión del espectro de frecuencia x^i .
3. Determinar la dimensión de la capa de salida en función del número de condiciones de salud de la maquinaria y aplicar el algoritmo BP para ajustar los parámetros y minimizar el error.
4. Aplicar la DNN para diagnosticar fallos en la maquinaria.

Se llevan a cabo dos estudios de caso, el primero basado en el diagnóstico de fallos de rodamientos y el segundo en el diagnóstico de fallos de cajas de engranajes planetarios.

Caso 1. Diagnóstico de fallos de rodamientos:

Se utilizan cuatro conjuntos de datos de los rodamientos para probar el rendimiento del método propuesto. Se aplica una DNN de cinco capas donde el número de unidad de la capa de entrada se determina por la dimensión de las muestras y el número de unidad de la capa de salida se determina por el número de condiciones saludables. Las

funciones de activación son funciones tangente hiperbólicas.

Se realizan 20 ensayos obteniendo un porcentaje de precisión de más del 99%. Se observa que el método propuesto puede distinguir de manera efectiva y estable las categorías de fallos de los rodamientos e incluso también la gravedad de los fallos.

Caso 2. Diagnóstico de fallos en cajas de engranaje planetarios:

En este caso se llevan a cabo siete experimentos bajo diferentes condiciones de salud. Cada experimento se realiza en ocho condiciones diferentes y contiene un componente dañado. Para realizar los ensayos se utiliza el 50% de las muestras, elegidas al azar. Las muestras restantes se utilizan para la fase de prueba.

Una vez realizados los 20 ensayos pautados, se observa que la precisión promedio tanto del entrenamiento como de las pruebas es del 100%, lo que significa que todas las muestras se han clasificado correctamente.

Se concluye el estudio considerando que el método propuesto es capaz de extraer características de fallos de forma adaptativa y de clasificar las condiciones de salud de la maquinaria de manera efectiva.

3.1.2 Recopilación de casos de uso

En la Tabla 2 se recogen 14 casos más donde se han aplicado diferentes técnicas de ML con el objetivo de diagnosticar y clasificar fallos.

Tabla 2. Casos de uso de diagnóstico de fallos (Fuente propia)

Autor	Técnica utilizada	Breve explicación
J. Hassan et al. [63]	ANN	Se presenta un modelo de ANN que tiene la intención de detectar y clasificar defectos de soldadura mediante características geométricas y estadísticas de los elementos analizados. En la aplicación del algoritmo propuesto se observa que el 86% de los defectos se clasifican y detectan correctamente.
R. Jegadeeshwaran et al. [64]	DT, SVM	Se pretende monitorear el estado de los frenos hidráulicos de los automóviles con el objetivo de diagnosticar fallos mediante aprendizaje supervisado, basado en características de

		<p>vibración. Para la obtención de características se aplica un modelo de árboles de decisión. Posteriormente para la clasificación de los errores se utiliza un modelo de SVM.</p> <p>El estudio se completa comparando diferentes funciones de núcleo para el algoritmo SVM para la clasificación de los fallos. Se obtiene el mejor resultado con la función de base radial, llegando a alcanzar una precisión de acierto del 98,72%.</p>
Nagaraj Bhat et al. [65]	SVM	<p>Se utiliza SVM, con núcleo gaussiano y polinomial, con intención de clasificar imágenes de soldadura que previamente han sido procesadas. Dicha clasificación también se lleva a cabo utilizando Naïve Bayes y árboles decisión para demostrar que con SVM se consigue mayor precisión.</p> <p>Con la técnica elegida se obtiene resultados de precisión superior al 97% en la clasificación entre soldadura buena y defectuosa, y se observa que el núcleo gaussiano muestra mejores resultados que el núcleo polinomial.</p>
XiaoLi Zhang et al. [66]	SVM, MEISVM	<p>Se propone el modelo MEISVM, un método derivado del algoritmo SVM, para detectar fallos en rodamientos y, de esta manera, garantizar la seguridad operativa de la maquinaria.</p> <p>Se analiza el modelo llevando a cabo cuatro estudios de caso diferentes y se demuestra la superioridad de este respecto a otros 7 métodos. Posteriormente, se aplica MEISVM en el diagnóstico de fallos en rodamientos de locomotoras y se logra una precisión final superior al 96%. Se destaca la versatilidad del modelo para ser aplicado con éxito en diferentes tipos de maquinaria y, por lo tanto, se concluye la investigación con éxito.</p>
Olivier Janssens et al. [67]	CNN	<p>Se utiliza el modelo de red neuronal convolucional para la detección automática de fallos de rodamientos.</p> <p>Se observa que al usar el modelo CNN se pueden</p>

		conseguir resultados mejores para la detección de fallos, como por ejemplo fallos de la pista de rodadura externa y variaciones de los niveles de degradación del lubricante. Se obtiene una mejora en la precisión de clasificación de aproximadamente el 6% respecto los métodos previos.
Chen Lu et al. [68]	AE	<p>Se propone la aplicación de una red neuronal profunda (autoencoder) para diagnosticar y clasificar fallos en rodamientos.</p> <p>Se concluye el trabajo apreciando que este modelo, basado en aprendizaje no supervisado, es capaz de extraer de manera adaptativa las características de fallos e identificar con precisión los estados de salud de manera efectiva.</p>
Peng Wang et al. [69]	CNN	<p>Se presenta un sistema híbrido para monitorear las condiciones de salud de cajas de engranajes basado en transformaciones Wavelet y redes neuronales convolucionales. Este sistema permite detectar fallos y clasificarlos en función de la gravedad.</p> <p>Se obtienen resultados de precisión de la clasificación superiores al 99,5%. Se comparan los resultados obtenidos con los que se obtienen aplicando otras tres técnicas y se observa la superioridad de efectividad que proporciona el modelo planteado.</p>
Zhiqiang Chen et al. [70]	NN	<p>Se plantean 3 modelos diferentes de redes neuronales profundas (RBM, DBN y AE) con el objetivo de diagnosticar fallos de rodamientos. Los algoritmos se entrenan definiendo 7 patrones de fallos y realizando 315 pruebas. Se generan 6300 muestras, de las cuales se utiliza el 80% para el entrenamiento del modelo y el 20% para la validación de estos.</p> <p>Se muestra que los modelos estudiados, a pesar de no ser buenos en la extracción de características de datos brutos, son efectivos en el diagnóstico de fallos de rodamientos, alcanzando</p>

		<p>una precisión superior al 99%. Se considera AE ligeramente mejor y se observa que DBN necesita más tiempo de entrenamiento.</p>
Sahir Moreno et al. [71]	CNN	<p>Se desarrolla una técnica para lograr que la inspección de palas de turbinas eólicas se realice de manera más segura y efectiva que manualmente. Se lleva a cabo instalando una cámara en un dron, que vuela alrededor de las palas escaneando la superficie, y a través de una red neuronal convolucional es capaz de clasificar el tipo de daños entre impacto de rayos, desgaste y/o fracturas.</p> <p>A pesar de que la CNN fue entrenada con pocas imágenes, se obtiene una precisión general del sistema del 81,25%.</p>
Bin-Sen Peng et al. [72]	NN, SVM	<p>Se aplica una red neuronal profunda con el objetivo de desarrollar un modelo de diagnóstico de fallos para una central nuclear. El algoritmo se entrena simulando siete estados operativos diferentes y 64 parámetros.</p> <p>Una vez obtenidos los resultados del modelo se comparan con otros dos métodos y se demuestra una precisión notablemente superior. La red neuronal profunda planteada muestra una precisión superior al 99%, mientras que la NN entrenada con BP y el método SVM obtienen aproximadamente una precisión del 95% y 93% respectivamente.</p>
Liangzhi Li et al. [73]	CNN	<p>Se implementa un sistema de inspección para detectar defectos de los productos y clasificar el tipo y grado de la imperfección. El modelo planteado está basado en redes neurales convolucionales y garantiza el computo instantáneo de datos.</p> <p>El método propuesto dispone de un conjunto de datos de 10 categorías, cada una de las cuales contiene 200 imágenes para entrenar el modelo y 50 imágenes para probarlo. En comparación con otros dos métodos anteriores, el modelo</p>

		implementado obtiene resultados superiores en cuanto a precisión y fiabilidad.
Shrinathan Esakimuthu et al. [74]	SVM, CNN	<p>Se selecciona el algoritmo SVM como método para el diagnóstico y la detección de fallos menores en rodamientos, ya que es el fallo más frecuente en motores de inducción. El método seleccionado se compara con datos obtenidos de un modelo CNN.</p> <p>A pesar de que ambos algoritmos diagnostican fallos de los rodamientos de manera efectiva, se obtiene mayor tasa de precisión con el modelo CNN gracias a la profundidad que este ofrece en el análisis de datos.</p>
Jianyu Wang et al. [75]	CNN, SVM	<p>Se plantea un método de aprendizaje conjunto entre CNN y SVM con objetivo de crear un sistema inteligente de diagnóstico de fallos. El modelo trata de ser eficaz evaluando el estado operativo de la máquina y emitiendo alerta temprana en caso de fallo.</p> <p>Se utilizan tres conjuntos de datos de rodamientos y dos de engranajes, y se plantean tres tipos de fallos: interno, externo o fallo del rodillo. El método propuesto permite el ahorro de tiempo y presenta buenos resultados de eficiencia.</p>
Orhan Yaman [76]	SVM, KNN	<p>Se aplican los algoritmos SVM y KNN para la detección y clasificación automatizada de fallos de cojinetes y rotores en motores de inducción trifásicos. Se lleva a cabo a partir de señales sonoras y se observa que, a mayor cercanía del micrófono con el motor, mayor precisión de clasificación.</p> <p>Dado el bajo número de datos, el entrenamiento de los modelos se realizó por validación cruzada en 10 tiempos. Los métodos propuestos se utilizaron para la detección de fallos de 5 motores diferentes, logrando precisiones de clasificación superiores al 99% en ambos casos.</p>

3.2 Análisis predictivo para el pronóstico de defectos

Poder elegir el momento en que interrumpir la producción para el mantenimiento o reemplazo de elementos de falla próxima, puede facilitar una reducción importante en el coste de los procesos. El mantenimiento preventivo es una de las aplicaciones de las técnicas de ML y en los últimos años se le ha otorgado mucha importancia.

3.2.1 Caso de uso analizado

El artículo “Machine Preventive Replacement Policy for Serial Production Lines Based on Reinforcement Learning” publicado el año 2019 por IEEE tiene por objetivo obtener la política óptima de reemplazo de maquinaria en una línea de producción en serie. Para lograrlo se propone la utilización del algoritmo RL basado en Q-Learning [28].

En el sector industrial se producen diferentes fallos inesperados en las máquinas que generan interrupciones en la producción, y a su vez pueden desencadenar grandes pérdidas económicas. Para evitar esta situación es necesario reemplazar la máquina degradada. Se pueden diferenciar dos tipos de mantenimiento: el mantenimiento preventivo (PM), donde se reemplaza la máquina envejecida antes de haber generado fallos; el mantenimiento correctivo (CM), donde se reemplaza la máquina cuando ya ha fallado.

Es preferible reemplazar las máquinas viejas antes de que fallen, ya que en general en una misma máquina, un CM provocará la parada de producción durante más tiempo que un PM. Por este motivo la clave del mantenimiento es determinar el momento óptimo del PM para poder optimizar el beneficio general del sistema de fabricación.

El agente de aprendizaje por refuerzo requiere gran cantidad de muestras de transición paso a paso para entrenar su política. Se aplica el método de modelado basado en datos propuesto en el trabajo “Production system performance identification using sensor data” ya que cumple con el requisito de la necesidad de tener buena eficiencia de cálculo.

A partir del modelo basado en datos se demuestra que en una línea de producción en serie una parada de la máquina puede causar pérdidas de producción del sistema cuando la máquina más lenta (S_{M^*}) se ve bloqueada. Discretizando el tiempo en pasos y teniendo en cuenta los diferentes ítems que condicionan el tiempo de parada $D(t)$, la pérdida de producción del sistema $PL(t)$ durante el paso del tiempo es:

$$PL(t) = \frac{D(t)}{T_{M^*}} \quad (47)$$

donde T_{M^*} es el tiempo de ciclo de la máquina (S_{M^*}). En la simulación gradual la pérdida de producción del sistema $PL(t)$ también puede representarse como:

$$PL(t) = \frac{1}{T_{M^*}} - (Y_{M^*}(t) - Y_{M^*}(t - 1)) \quad (48)$$

donde $1/T_{M^*}$ es el incremento de producción ideal de la máquina más lenta (S_{M^*}) y el segundo término es la producción real durante el paso del tiempo.

Para formular el reemplazo de la máquina como un problema de aprendizaje por refuerzo, la representación de estados s_t , la representación de acciones a_t y la función de recompensa r_t , que se pueden observar en la Figura 25 deben definirse adecuadamente. Se expresan de la siguiente manera:

$$s_t = [g_1(t), \dots, g_M(t), b_2(t), \dots, b_M(t), d_1^r(t), \dots, d_M^r(t)] \quad (49)$$

donde $g_i(t) \in \{0, 1, \dots, N_i\}$ es el estado de degradación de la máquina s_i , $b_i(r) \in [0, B_i]$ es el nivel de la disolución amortiguadora B_i y $d_i^r \in [0, \{d_i^{cm}, d_i^{pm}\}]$ es el tiempo restante de la actividad de mantenimiento en curso.

$$a_t = [a_1(t), a_2(t), \dots, a_M(t)] \quad (50)$$

donde $a_i(t) \in \{0, 1\}$, siendo $a_i(t) = 1$ cuando la máquina está apagada para tiempo t .

$$r_t = - \sum_{i=1}^M w_i(t)c_i - \sum_{i=1}^M a_i(t)c_i - c_p \cdot PL(t) \quad (51)$$

donde $\sum_{i=1}^M w_i(t)c_i$ y $\sum_{i=1}^M a_i(t)c_i$ es el coste total de nuevas máquinas para CM y PM en el paso de tiempo t respectivamente. $c_p \cdot PL(t)$ es la pérdida de ingreso debido a la pérdida de producción en el paso de tiempo t . La función de recompensa es negativa ya que el objetivo es maximizar las recompensas y en consecuencia minimizar los costes. Un gran proceso de decisión de Markov (MDP) está definido y puede ser resuelto con Q-learning y el simulador de modelo basado en datos.

Se lleva a cabo una simulación de una línea de producción en serie de 3 máquinas para demostrar la efectividad de la técnica de resolución planteada. Esta simulación se basa en los parámetros recogidos en la Tabla 3.

Tabla 3. Parámetros de las máquinas (Extraído de [28])

Parámetros	Máquinas		
	S ₁	S ₂	S ₃
Tiempo de ciclo t_i (min)	1,25	0,8	1,00
Coste de máquina c_i (\$)	1200	1600	1500
Duración de PM d_i^{pm} (min)	5,0	5,0	5,0
Duración de CM d_i^{cm} (min)	20,0	20,0	20,0

Se entrena el agente para 27.000 iteraciones y cada una de éstas supone 10.000 minutos de duración. Cada iteración empieza con estados iniciales aleatorios. El factor de descuento γ se establece en 0,99. La tasa de aprendizaje α se establece en 0,1 y el parámetro de ϵ -greedy es 0,2. El paso de tiempo es 1 minuto. Se tiene en cuenta que la posible vida útil de la máquina es mucho mayor que el paso de tiempo, para evitar que esto suponga un problema se decide omitir el ϵ -greedy.

Se compara la política obtenida mediante RL basado en Q-learning con otras dos políticas (run-to-failure y state-dependent-PM) no basadas en algoritmos de ML para demostrar que la propuesta es realmente efectiva. Se obtiene que el coste de mantenimiento por minuto es mucho inferior a las políticas planteadas anteriormente.

3.2.2 Recopilación de casos de uso

En la Tabla 4 se recogen 14 casos más donde se han aplicado diferentes técnicas de ML con el objetivo de predecir y pronosticar defectos.

Tabla 4. Casos de uso de predicción de defectos. (Fuente propia)

Autor	Técnica utilizada	Breve explicación
Piotr Bilski [77]	SVM	Se elige la técnica SVM para identificar los parámetros de la máquina de inducción, con el objetivo de monitorear el sistema de línea para detectar y clasificar fallos. Con este proceso se pretende facilitar las reparaciones rápidas. El modelo demostró ser resistente a pequeños cambios.
Gian Antonio Susto et al.	SVM, KNN	Se presenta la metodología MC, basada en las técnicas de clasificación SVM y KNN con la intención de gestionar de manera eficiente actividades de

[78]		<p>mantenimiento preventivo y reducir costes asociados al tiempo de inactividad. Los clasificadores proporcionan información del estado de salud del proceso y por lo tanto el rendimiento de la metodología aumenta con el número de clasificadores.</p> <p>Se lleva a cabo un estudio donde se aplican 6 enfoques de mantenimiento diferentes y se demuestra que la metodología propuesta basada en las técnicas de ML ofrece costes operativos más bajos que las técnicas clásicas. Se destaca también la superioridad de rendimiento del algoritmo SVM respecto KNN</p>
Yu Ding et al. [79]	RL	<p>Se estudia un modelo basado en regresión logística y regresión softmax con el objetivo de analizar la evolución de salud de una bomba hidráulica. El modelo estudiado se aplica en una bomba hidráulica con dos fallos comunes y se obtienen 12 muestras en condiciones normales y 4 para cada uno de los fallos. Cuando el valor de confianza (0 a 1), es menor a 0,7 se activa la alerta de fallo para la clasificación de este.</p> <p>Se considera que el método es eficaz y eficiente para la evaluación del estado de salud y la clasificación dinámica de fallos. Se contempla la opción de incorporar fallos adicionales para lograr mejor rendimiento.</p>
Nagaraj N. Bhat et al. [80]	HMM	<p>Se utiliza el modelo oculto de Markov para clasificar los estados de desgaste de la herramienta empleada para el torneado de una pieza de acero de carbono medio. Los datos de entrada del algoritmo son las características extraídas del análisis de textura realizado, basado en la matriz de coincidencia de nivel gris, a las imágenes de la superficie de la máquina.</p> <p>En este estudio se obtiene una precisión de clasificación superior al 94%.</p>
Wu Yuting et al. [81]	LSTM, RNN, GRU	<p>Se propone el uso de LSTM Vanilla, un modelo de LSTM con una única capa de unidades LSTM, con el objetivo de obtener estimación de vida útil y predecir</p>

		<p>degradaciones del sistema para facilitar el mantenimiento previo a la fallida. Para realizarlo se llevan a cabo experimentos con cuatro conjuntos de datos de simulación de motores de aeronaves Turbofán en los que se inyectan fallos desconocidos por los analizadores de datos.</p> <p>El modelo elegido se compara con otros dos tipos de RNN. El rendimiento de predicción de LSTM Vanilla ofrece superioridad tanto en predicción a corto plazo como a largo plazo.</p>
Chongchong Qi et al. [42]	SVM	<p>Se aplican seis técnicas de aprendizaje automático (LR, DT, GBM, MLP, SVM y RF) con la intención de analizar la estabilidad de taludes, para poder predecir movimientos y prevenir posibles fallos. El modelo utiliza un conjunto de datos de 78 casos de pendiente estable y 70 casos de pendiente inestable. Del conjunto de datos se usa el 70% de datos para el entrenamiento del algoritmo y el 30% restante para las pruebas de análisis.</p> <p>Se obtienen altos valores de precisión con los modelos de ML, destacando entre ellos los resultados obtenidos con el algoritmo SVM, que predijo más del 95% de los casos correctamente.</p>
Mohamed Elforjani et al. [82]	ANN, SVM, GPR	<p>Se estudian los métodos de regresión SVM, ANN y GPR para el pronóstico de fallos. El objetivo es mantener las máquinas a niveles óptimos y predecir la vida útil restante de los componentes. Los modelos se basan en la relación de características de emisión acústica con el desgaste natural correspondiente de cojinetes de baja velocidad. El conjunto de datos se extrae de cuatro casos. Se utilizan datos del caso 1 para el entrenamiento de los modelos y datos de los casos 2, 3 y 4 para la validación de estos.</p> <p>Se aprecia que el modelo ANN con BP es altamente competitivo, obteniendo el error más bajo y siendo capaz de mejorar exitosamente la predicción de vida útil restante.</p>
Guillermo R.	CNN, SVM,	<p>Se propone una CNN de clasificación, basada en aprendizaje supervisado, para predecir fallos en</p>

Moreira et al. [83]	MLP	<p>anclajes industriales. Se logra reducir un 10,9% el tiempo de ejecución de la tarea de subprocesos.</p> <p>Se compara esta técnica con los algoritmos de aprendizaje supervisado SVM y MLP. El modelo propuesto se muestra como el mejor clasificador obteniendo buena precisión y el menor número de falsos positivos.</p>
Guangxing Niu et al. [84]	LSTM, ANN, RNN	<p>Se presenta un enfoque novedoso basado en memoria a largo plazo (LSTM), un tipo de red neuronal recurrente, y filtrado de partículas con el objetivo de pronosticar el estado de la máquina y predecir el inicio y la evolución de posibles fallos.</p> <p>Con la intención de evaluar el rendimiento del modelo se lleva a cabo un estudio en una caja de cambios agrietada, donde se comparan los métodos ANN, RNN y LSTM con el modelo propuesto. Los resultados obtenidos muestran que el modelo propuesto es claramente superior en cuanto a precisión de predicción de estado de máquina.</p>
Ido Amihai et al. [85]	RF	<p>Se desarrolla un estudio basado en monitoreo de vibraciones para la supervisión del estado de los elementos imprescindibles en una planta industrial. Se recopilan datos de 30 bombas industriales en una planta química durante 2 años y medio para validar el modelo propuesto y garantizar la correcta predicción de fallos.</p> <p>Se utiliza el algoritmo RF y se compara con la técnica de persistencia. Se observa que puede llegar relativamente rápido a resultados razonables con datos reales, siendo el modelo capaz de predecir fallos de activos antes de que ocurran.</p>
Uferah Shafi et al. [86]	SVM, KNN, DT, RF	<p>Se propone un enfoque para la predicción de fallos de cuatro sistemas principales del vehículo (sistema de combustible, de encendido, de escape y de enfriamiento). El conjunto de datos que se utiliza para el entrenamiento de las técnicas proviene de 70 vehículos. Se trata de un conjunto de datos de naturaleza binaria, es por eso que se seleccionan cuatro algoritmos con buen funcionamiento</p>

		<p>procesando datos binarios: DT, SVM, KNN y RF.</p> <p>Una vez obtenidos los resultados de la precisión de predicción, se demuestra la capacidad de los métodos de aprendizaje automático para predecir correctamente. Se destaca la superioridad de SVM e inferioridad de DT. SVM obtiene el mejor rendimiento en los cuatro sistemas, superando el 95% en todos ellos.</p>
Xiao-long Chen et al. [87]	KNN	<p>Se plantea, en centrales eléctricas, un método de monitoreo de condición y alerta temprana llamado CMEW-EKNN, una variación del algoritmo K-vecinos más cercanos. El modelo propuesto adopta el factor de descuento adaptativo, que le permite mejorar la precisión y solidez.</p> <p>Se llevan a cabo dos estudios de aplicación del método y se demuestra tener superioridad en la presentación del estado de las condiciones del proceso y en el rendimiento de alerta temprana de fallos en caso de posibles anomalías.</p>
Kyu Han Kim et al. [88]	NN, K-means	<p>Se implementa un método basado en aprendizaje automático para la detección de fugas en el tubo de la caldera en una central térmica. Se emplean NN para detectar fallos, se analizan los casos de fallos causados por fugas y se clasifican los datos de estado utilizando el algoritmo K-means (con distancia euclidiana).</p> <p>Los resultados muestran que los fallos se detectaron con éxito antes de que ocurrieran.</p>
Nikolaos Kolokas et al. [89]	IF	<p>Se selecciona el algoritmo Isolation Forest (IF), método de aprendizaje no supervisado derivado de RF, para la previsión o pronóstico de fallos en equipos industriales. Se utilizan datos industriales reales relacionados con la producción de aluminio y plástico. El objetivo del método propuesto es detectar valores atípicos para anticiparse a los fallos y permitir el mantenimiento predictivo.</p> <p>Se obtienen resultados satisfactorios y se confirma que el modelo es capaz de pronosticar con éxito fallos</p>

antes de que ocurran. Se destaca la necesidad de re-entrenar ocasionalmente el modelo propuesto, dado que las condiciones pueden variar.

3.3 Optimización de la productividad

Los procesos de optimización son importantes para garantizar la fabricación competitiva, ya sea mejorando la calidad del producto, rentabilizando los procesos de fabricación, como reduciendo tiempo invertido i/o inversión económica.

3.3.1 Caso de uso analizado

El artículo “Applications of artificial intelligence principles in mechanical engineering”, publicado el año 2018 por MATEC Web of Conferences, tiene por objetivo optimizar el proceso de producción de neumáticos. Se basa en el estudio de la viabilidad de la automatización completa del control de calidad mediante redes neuronales convolucionales profundas [51].

Actualmente todo el proceso de control de calidad se basa en los conocimientos de un operador sobre el producto. En primer lugar, se analizan las tareas que realiza dicho operador y que será necesario automatizar:

- 1) Manipulación del neumático a evaluar.
- 2) Automatización del control de calidad.
- 3) Detección automática y eliminación de posibles residuos del proceso de moldeo y vulcanización del neumático.

En segundo lugar, se selecciona el método que se aplicará. Se decide aplicar redes neuronales convolucionales profundas (DCNN), técnica de aprendizaje no supervisado, para detectar y clasificar los errores de la superficie de los neumáticos evaluados. Aplicando este tipo de redes neuronales se espera conseguir un aumento significativo del rendimiento del sistema. Para poder hacer un análisis mediante DCNN, previamente se necesita incorporar al puesto de inspección una cámara que genere imágenes de alta calidad del neumático testado.

Para el entrenamiento de la DCNN se usa el método de descenso de gradiente estocástico en mini lotes (minibatch stochastic gradient descent). Calcular el gradiente de error para una pequeña porción de muestras en cada iteración es la principal diferencia del método de gradiente estocástico en minilotes respecto el método de

descenso de gradiente estándar, donde se calcula el gradiente error de todo el conjunto provocando así costes elevados en redes grandes.

La aplicación del método se detalla a continuación:

El número total de iteraciones durante el entrenamiento viene representado por $T = n / b$, donde b es el tamaño de mini lotes y n el número total de muestras de entrenamiento. Por lo tanto, los parámetros w se obtienen optimizando el valor aproximado esperado de la función de error f definida como:

$$E_t[f(w)] = \frac{1}{b} \sum_{i=(t-1)b+1}^{tb} f(w; x_i) \quad (52)$$

dónde $t \in \{1, \dots, T\}$ es el índice de iteración y x_i es la i -ésima muestra de entrenamiento. En cada iteración el peso se ajusta por regla de descenso de gradiente:

$$w^{(t+1)} = w^{(t)} - \mu \nabla_w E_t[f(w^{(t)})] \quad (53)$$

donde μ es la tasa de aprendizaje.

Una vez implementada la modificación de la construcción y el montaje en el puesto de inspección se lleva a cabo la fase de implementación. Esta fase permite recopilar la definición inequívoca de las imágenes adquiridas para el proceso de definición de errores que pueden surgir en la producción de neumáticos. En esta etapa también será necesario definir los rangos de desviaciones aceptables del estado ideal del producto. Todas las imágenes capturadas deberán dividirse en clases con aparente similitud.

Una vez completada con éxito la fase de implementación, se realizan pruebas para revelar una variación adecuada del modelo de redes neuronales convolucionales profundas. Estas pruebas se diferencian en la combinación de parámetros básicos (número de mapas de elementos, tamaño de los filtros aplicados, número de capas de red, número de capas interconectadas). Después de realizar estas pruebas se seleccionan los parámetros que permiten la máxima precisión de clasificación a través de las DCNN.

Se concluye el estudio con buenas sensaciones respecto a la propuesta realizada. La implementación del método generará una gran cantidad de datos de entrada mediante imágenes, se utilizarán las DCNN para extraer y reconocer efectivamente los elementos de la imagen y posteriormente, se podrá detectar automáticamente la presencia de defectos.

3.3.2 Recopilación de casos de uso

En la Tabla 6 se recogen 14 casos más donde se utilizan técnicas de ML con el objetivo de optimizar los procesos de fabricación.

Tabla 5. Casos de uso de optimización. (Fuente propia)

Autor	Técnica utilizada	Breve explicación
Girish Kant et al. [90]	ANN	<p>Se realiza un experimento de mecanizado con la intención de investigar la capacidad de una red neuronal artificial para predecir los valores de energía específica de corte. La ANN aplica el algoritmo BP y se entrena usando el 80% de las muestras y el 20% restante se utiliza para la validación.</p> <p>Se concluye con éxito el experimento ya que, tras 324 iteraciones, el error cuadrático medio se estabiliza y se obtiene una diferencia entre el modelo predicho y el valor real de tan solo el 1,5%.</p>
Jianbin Xiong et al. [61]	KNN	<p>Se presenta un método para diagnosticar fallos en maquinaria rotativa de equipos petroquímicos a gran escala. El método se basa en un factor de descuento estático y en el algoritmo KNN con indicadores adimensionales.</p> <p>Con las 1024 muestras de datos de fallos se demuestra que el modelo planteado puede reducir eficazmente la influencia de factores poco fiables en los resultados para facilitar la toma de decisiones de forma precisa.</p>
Daniel Weimer et al. [91]	CNN	<p>Se investiga el diseño de arquitecturas CNN para llevar a cabo el control de calidad de forma automática mediante la inspección óptica industrial. El objetivo de este estudio es poder detectar los defectos y clasificarlos en categorías previamente establecidas.</p> <p>Los resultados demuestran que agregando capas, donde se extrae conocimiento, se mejora la precisión del modelo llegando a superar el 97% de éxitos.</p>
Carlos A.	LR	Se aplica regresión logística con la intención de

Escobar et al. [19]		<p>detectar eventos de calidad poco frecuentes en sistemas de fabricación. El objetivo del modelo es identificar y clasificar las características más importantes sobre el producto y garantizar una calidad casi perfecta.</p> <p>Según los resultados, el 100% de los defectos se pueden detectar de manera efectiva.</p>
Slawomir Klos et al. [92]	ANN	<p>Se elabora un modelo sobre la base de un sistema de fabricación en serie de piezas de repuesto de amortiguadores. Se utilizan redes neuronales artificiales con el objetivo de reducir costes y de mejorar el rendimiento.</p> <p>Una vez realizados 100 experimentos de simulación se concluye el estudio habiendo podido garantizar un nivel aceptable de rendimiento del sistema y de la vida útil del producto.</p>
Lukas Lingitz et al. [93]	RF	<p>Se estudia la predicción del tiempo de entrega, optimización de calidad y eficiencia de la planificación y programación basada en aprendizaje supervisado. Se evalúan 11 métodos de regresión (modelos basados en árbol, regresión lineal, ANN, MARS, SVM y KNN). Los modelos se entrenan con un conjunto de 18.532 datos, de los cuales se utilizan 70% y 30%, para entrenamiento y evaluación respectivamente.</p> <p>Se obtiene que los modelos basados en árbol pueden superar a los demás, destacando RF como el mejor de todos ellos.</p>
Xinyian Ou et al. [94]	RL	<p>Se pretende automatizar un pórtico para cargar y descargar materiales, con la intención de encontrar la política de movimiento en tiempo real que maximice la producción. Se lleva a cabo mediante aprendizaje por refuerzo basado en Q-learning.</p> <p>Se demuestra que el algoritmo RL puede reducir la pérdida de producción de manera efectiva.</p>
Durga Prasad Penumuru et	SVM	<p>Se utiliza visión artificial y el algoritmo SVM para identificar y clasificar con precisión materiales planos, con el objetivo de facilitar la selección de parámetros de mecanizado en función del material identificado.</p>

al. [58]		<p>Se usan 3559 muestras de aluminio, cobre, fibra y acero, que se distribuyen 70-30 en datos de entrenamiento y prueba respectivamente.</p> <p>Se compara el modelo propuesto con otras técnicas y destaca como la mejor, dado que consigue mejores valores de precisión en tiempos de entrenamiento más cortos.</p>
Minxing Si et al. [95]	NN	<p>Se realiza la investigación de un sistema de medición de emisiones de óxido de nitrógeno usando la biblioteca de código abierto Keras. Se seleccionan 5 estructuras de red al azar (con diferente número de capas y neuronas por capa cada una) y 3 métodos de optimización diferentes.</p> <p>Se registran mejores resultados en modelos que tienen más número de capas y han sido optimizados, llegando a solo el 0,06% de diferencia entre los valores predichos y obtenidos.</p>
Zilong Ti et al. [48]	ANN	<p>Se utiliza un modelo ANN entrenado con el algoritmo BP para mejorar las predicciones de estela tridimensional de una turbina.</p> <p>El modelo entrenado muestra eficiencia excelente y puede predecir con precisión los campos de velocidad y turbulencia. La ANN obtiene valores de error inferiores al 5%.</p>
Jesus Romero et al. [96]	RL	<p>Se lleva a cabo una simulación de una secuencia de soldadura con ocho cordones, donde se aplica el algoritmo RL basado en Q-learning con la intención de minimizar la deformación estructural. El algoritmo de aprendizaje por refuerzo tiene por función de recompensa esta deformación estructural.</p> <p>El modelo propuesto se compara con otras técnicas. Se observa que el algoritmo RL es mucho más rápido y a la vez más eficiente que las demás, llegando a conseguir reducir las deformaciones estructurales hasta un 71%.</p>
Panagiotis D. Paraschos et	RL	<p>Se propone la implementación de un modelo basado en RL, para abordar problemas de optimización y obtener las políticas óptimas conjuntas de</p>

al. [97]		<p>producción, mantenimiento y control de calidad del producto, con el objetivo de optimizar la rentabilidad de la instalación de fabricación.</p> <p>El modelo propuesto se evalúa en 18 escenarios diferentes y a partir de 15 iteraciones. El beneficio de la política obtenida con RL es notablemente alto en comparación con otras políticas de control en los mismos escenarios. Por lo tanto, se concluye resaltando la capacidad del método RL de mejorar de manera efectiva la rentabilidad de un sistema de fabricación.</p>
Yan He et al. [98]	CNN	<p>Se utilizan técnicas de aprendizaje profundo para la predicción de energía utilizada. El objetivo de este estudio es mejorar la eficiencia energética de las máquinas para optimizar los procesos de fabricación y desarrollar una industria sostenible. El modelo se basa en una CNN (no supervisado) para la extracción de características de consumo, y en una 1D-CNN (supervisado) para la predicción de energía consumida.</p> <p>Se expone que el método propuesto, tras realizarse un experimento en una fresadora y en una rectificadora, es capaz de aumentar la precisión de la predicción ya que puede extraer información de consumo de manera más completa respecto a los métodos tradicionales.</p>
Ying Wang et al. [99]	RL	<p>Se presenta un modelo basado en el método Monte Carlo de RL, con el objetivo de predecir la capacidad de producción de una industria hidroeléctrica. Dado que el conjunto de datos para el entrenamiento es reducido, se utiliza el método Monte Carlo para generar una expansión de estos. Se evalúan los datos expandidos y se demuestra que la diferencia con los datos originales no es significativa. Para el entrenamiento del método propuesto se utilizan 900 conjuntos de datos y 100 para probarlo.</p> <p>Según los análisis realizados, aplicar el método Monte Carlo puede aumentar la generación de energía en aproximadamente 12.967.400kW/h</p>

3.4 Software utilizado

En este apartado se recoge información sobre el software utilizado para la aplicación de algoritmos de ML de los 45 estudios de caso revisados en los apartados anteriores.

Gran parte de los artículos científicos se limitan a plasmar el objetivo del estudio, el trabajo que se ha llevado a cabo y el resultado obtenido, sin especificar detalles del proceso ni las herramientas utilizadas. De hecho, se observa que prácticamente en la mitad de los casos seleccionados (51,1%) no se especifica el software utilizado para el procesamiento de datos.

Por este motivo, y a pesar de que para este trabajo sirve de referencia para determinar la tendencia, el diagrama de barras que se muestra a continuación (Figura 26), no se considera 100% significativo.

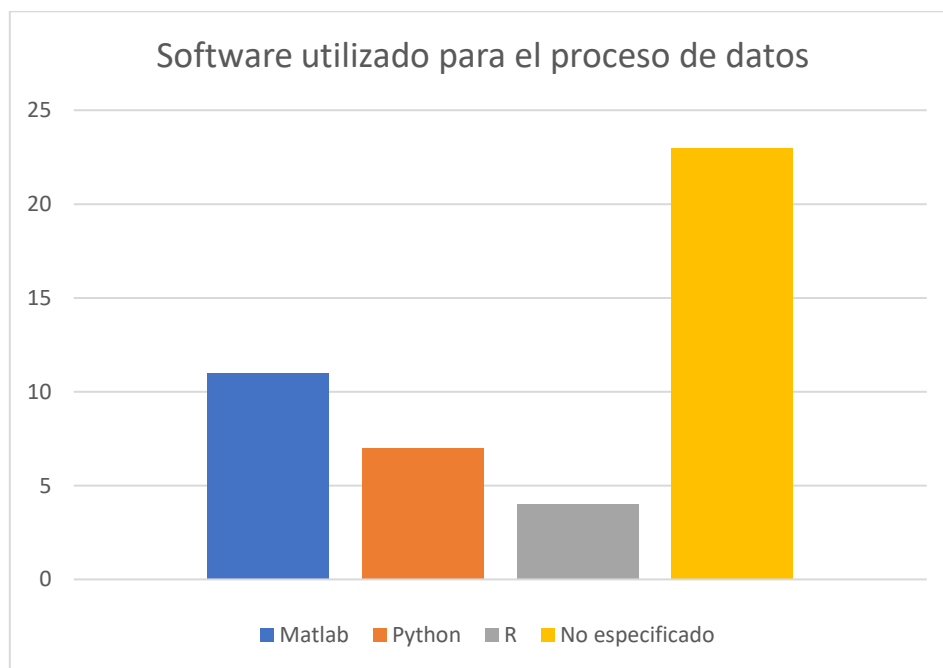


Figura 26. Diagrama de barras sobre el software utilizado en los casos de uso de ML revisados. (Fuente propia)

Matlab es, aparentemente, el software más utilizado, seguido de Python y R. No sorprende que puedan ser los más mencionados, ya que son tres softwares con lenguaje de programación propio muy conocidos y con mucho recorrido en investigación científica.

Matlab facilita el aprendizaje automático ya que da acceso inmediato a funciones prediseñadas, toolboxes exhaustivas y aplicaciones especializadas para la regresión, clasificación y clustering. Permite, entre otras muchas cosas, facilidades para entrenar

y comparar enfoques diferentes y técnicas de perfeccionamiento de modelos para crear modelos precisos con mayor capacidad de predicción [100].

Python es un lenguaje de código abierto que se ha convertido en uno de los lenguajes más utilizados desde su aparición. Los principios de Python se basan en facilitar la legibilidad y favorecer la simplicidad del código. La particularidad de Python es ser un lenguaje interpretado y multiparadigma. Tensorflow, MXNet y Scikit-learn son tres herramientas de Python enfocadas para el área de ML [101].

R también es un entorno de software libre que se caracteriza por ser un lenguaje de programación interpretado. Tiene un conjunto de características que permite tener amplia versatilidad en el manejo de elementos estadísticos, manipulación efectiva de datos, cálculo y gráficos. R destaca en ML por la velocidad que proporciona al análisis de datos [102].

A pesar de haber encontrado mención de únicamente estos tres softwares, son muchas las plataformas que permiten el desarrollo de modelos de aprendizaje automático y minería de datos. A parte de las ya mencionadas, SPSS Modeler, SAS, WEKA y KNIME se encuentran entre las más destacadas.

4. PRESUPUESTO

Ya que se trata de un trabajo 100% teórico, las horas empleadas por el alumno en la realización de este ocupan la parte más importante del presupuesto. Cursar el trabajo de final de grado de ingeniería mecánica en la EPSEVG consiste en un total de 24 créditos. En base a información extraída de la página web de la escuela, 1 crédito equivale a aproximadamente 25-30 horas de dedicación [103]. Cogemos por referencia las 30 horas por crédito.

Por lo que hace al precio por hora, en base al artículo 16 de la normativa de prácticas externas de la Universidad Politécnica de Catalunya, aprobada el 18 de diciembre de 2014 y modificada el 10 de febrero de 2015, el estudiante que realiza prácticas académicas externas tiene derecho a percibir una aportación económica con un precio orientativo de 8€ la hora [104]. Estos 8€/hora sirven de guía para el cálculo del presupuesto.

Además se ha tenido en cuenta dos cargos adicionales que se hacen en la matrícula. Por un lado, el seguro escolar, se trata de 1,12€. Y, por otro lado, el soporte para el aprendizaje escolar (35€), ya que, sin las suscripciones a las principales bases de datos de búsqueda científica, como SpringerLink, Ieee Xplore y ScienceDirect, el desarrollo del trabajo no hubiera posible.

Por último, se computa también la licencia para Microsoft 365 y gastos varios de copistería.

Tabla 6. Resumen de costes del trabajo. (fuente propia)

Concepto	Unidades / horas	Precio por unidad / hora (€)	Total (€)
Dedicación del estudiante	720 h	8€	5.760€
Seguro escolar (UPC)	1	1,12€	1,12€
Soporte al aprendizaje (UPC)	1	35€	35€
Licencia Microsoft 365 personal	1	69€	69€
Impresión (2 copias a color)	200	0,18€	36€
Encuadernación espiral	2	2,90€	5,80€
		TOTAL	5.906,92€

5. CONCLUSIÓN

En este trabajo se ha querido hacer visible el impacto que están teniendo las técnicas de aprendizaje automático en los procesos de fabricación de ingeniería mecánica.

El cuerpo principal del trabajo se ha dividido en dos partes, cada una de ellas con sus objetivos asociados. Por un lado, en el capítulo de Machine Learning, se pretendía contextualizar la situación actual del aprendizaje automático y lo que engloba. Creemos que hemos mostrado de forma clara y concisa en que consiste el ML y cuáles son las técnicas principales. Hemos abierto el capítulo presentando los conceptos clave para el desarrollo de la cuarta revolución industrial. Los tipos de ML, la breve explicación de los algoritmos más comunes y los elementos principales para el uso correcto de los mismos, concluyen esta primera parte del trabajo.

Cada una de las técnicas de ML que hemos citado, podría ser única protagonista de un trabajo de final de grado, por este motivo era inviable hacer un análisis exhaustivo de cada una de ellas. En este trabajo nos hemos ceñido a presentar el algoritmo y pinceladas de información sobre este.

Por otro lado, el apartado bibliográfico tenía por objetivo reflejar como las técnicas de ML son herramientas que se están utilizando en el entorno industrial con éxito. Para lograrlo hemos creído interesante llevar a cabo la selección de una amplia lista de estudios.

Desde 2015, ha crecido exponencialmente el número de publicaciones que presentan técnicas de ML para la mejora de procesos de fabricación o el diagnóstico y pronóstico de defectos. Continuamente aparecen algoritmos nuevos, en la mayoría de los casos derivados de los tradicionales, que favorecen la obtención de resultados mejores y, por lo tanto, abren la puerta a nuevos objetivos.

El objetivo principal de la aplicación de técnicas de ML en el ámbito de la ingeniería es proporcionar la fabricación competitiva, entendiéndose como beneficios más significativos de esto la reducción de costes de fabricación, reducción de tiempos de procesos y la mejora de calidad de productos. Estos beneficios generales, vienen desencadenados de otros más específicos, como el pronóstico, diagnóstico y clasificación de defectos, y a su vez, la programación de trabajos de mantenimiento y la detección temprana y precisa de los fallos producidos.

Actualmente, se utilizan principalmente algoritmos de aprendizaje supervisado y no supervisado. Estos dos tipos son capaces de procesar grandes cantidades de datos, en la mayoría de los casos obtenidos de sensores, para devolver especificaciones del estado de la máquina o proceso, y poder tomar decisiones en base a estas. En este momento, la meta del Machine Learning es seguir creciendo en la aplicación de algoritmos de aprendizaje reforzado. Aplicar con éxito este tipo de algoritmos supondría la capacitación de las máquinas de tomar decisiones por sí solas basándose en los resultados y las recompensas que van obteniendo.

Consideramos, que la ayuda de los algoritmos de ML puede marcar la diferencia a la hora de tomar decisiones, y que el IoT industrial ya está marcando un antes y un después a pesar de que todavía falte mucho por descubrir. Por este motivo, destacamos la importancia de integrar conocimientos básicos sobre Machine Learning en el currículo académico de un ingeniero.

Como conclusión final, haber dedicado tanto tiempo a la realización de este trabajo me ha llevado a un gran crecimiento personal, a nivel de organización de datos y programación de tempos. En especial, este TFG me ha dado a conocer un campo que desconocía totalmente y me ha generado inquietudes para seguir aprendiendo sobre el ML. Finalmente, que se tratase de un trabajo bibliográfico ha supuesto la lectura de muchísimos estudios, por lo que me ha ayudado a desarrollar nuevas técnicas de búsqueda y cribaje de información, así como de interpretación de recursos.

6. AGRADECIMIENTOS

A la familia. Por el amor y cariño incondicional, por acompañarme durante todo mi crecimiento académico y especialmente estos últimos meses.

Al equipo docente. Ingrid, gracias por escucharme, entenderme y apoyarme. Y, sobre todo, especial agradecimiento a Ester, gracias por tutorizar este trabajo, por guiarme tanto y tan bien y por la gran cantidad de horas dedicadas para que este TFG saliese adelante.

7. REFERENCIAS

- [1] Real Academia Española, «Diccionario de la Lengua Española,» 2020. [En línea]. Available: <https://dle.rae.es/ingenier%C3%ADa>. [Último acceso: 28 Abril 2020].
- [2] J. L. Muñoz Sanz, J. Echávarri Otero, M. Ceccarelli y P. Lafont Morgado, «The evolution and development of Mechanical Engineering Through Large Cultural Areas,» pp. 69-82, Enero 2009.
- [3] K.-D. Thoben, S. Wiesner y T. Wuest, «"Industrie 4.0" and Smart Manufacturing – A Review of Research Issues and Application Examples,» vol. 11, nº 1, pp. 4-19, 2017.
- [4] D. Preuveneers y E. Ilie-Zudor, «The intelligent industry of the future: A survey on emerging trends, research challenges and opportunities in Industry 4.0,» vol. 9, pp. 287-298, 2017.
- [5] I. Sittón Candanedo, E. Hernández Nieves, S. Rodríguez González, M. T. Santos Martín y A. González Briones, «Machine Learning Predictive Model for Industry 4.0,» vol. 877, pp. 501-510, Julio 2018.
- [6] D. Reinsel, G. John y J. Rydning, «Data Age 2025: The Digitization of the World From Edge to Core,» IDC White Paper, 2018.
- [7] P. M. LaCasse, W. Otieno y F. P. Maturana, «A Survey of Feature Set Reduction Approaches for Predictive Analytics Models in the Connected Manufacturing Enterprise,» *spp/sci*, vol. 9, p. 843, 27 Febrero 2019.
- [8] M. Rogalewicz y R. Sika, «Methodologies of knowledge discovery from data and data mining methods in mechanical engineering,» *Management and Production Engineering Review*, vol. 7, nº 4, pp. 97-108, Diciembre 2016.
- [9] P. Stanula, A. Ziegenbein y J. Matternich, «Machine learning algorithms in production: A guideline for efficient data source selection,» de *6th CIRP Global Web Conference – Envisaging the future manufacturing, design, technologies and systems in innovation era (CIRPe 2018)*, 2018.
- [10] T. Wuest, D. Weimer y C. T. L.-D. Irgens, «Machine learning in manufacturing: advantages, challenges, and applications,» *Production & Manufacturing Research*, vol. 4, nº 1, pp. 23-45, 24 Junio 2016.
- [11] J. Kozłowski, R. sika, F. Górski y O. Ciszak, «Modeling of Foundry Processes in the Era of Industry 4.0,» *Advances in Design, Simulation and Manufacturing. DSMIE 2018. Lecture Notes in Mechanical Engineering.*, pp. 62-71, 16 Junio 2018.
- [12] S. L. Brunton, B. R. Noack y K. Petros., «Machine Learning for Fluid Mechanics,» vol. 52, pp. 477-508, 12 Septiembre 2019.
- [13] C. Dhandapani y R. Sivaramakrishnan, «Implementation of Machine Learning (ML) in Mechanical Engineering Application using Artificial Intelligence (AI),» *Science, Technology and Development*, vol. 8, nº 10, p. 93, Octubre 2019.
- [14] J. T. Palma y R. Marin, *Inteligencia artificial. Técnicas, métodos y aplicaciones*, Madrid: McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S.A.U., 2008.
- [15] R. Cioffi, M. Travaglioni, G. Piscitelli y A. Petrillo, «Artificial Intelligence and Machine Learning Applications in Smart Production: Progress, Trends, and Directions,» *Sustainability*, vol. 12, nº 492, 8 Enero 2020.
- [16] J. Wang, Y. Ma, L. Zhang, R. X. Gao y D. Wu, «Deep learning for smart manufacturing: Methods and applications,» *Journal of Manufacturing Systems*, vol. 48, nº Parte C, pp. 144-156, Julio 2018.

- [17] J. F. Aldana Montse, A. Baldominos Gómez, J. C. García Nieto, F. Mochón Morcillo y I. Navas Delgado, *Introducción al Big Data*, Madrid: García Maroto editores, 2016.
- [18] A. Ziegenbei, P. Stanula, J. Metternich y E. Abele, «Machine learning algorithms in machining: a guideline for efficient algorithm selection,» de *Advances in Production Research. WGP 2018*, Cham, 2018.
- [19] C. Escobar y R. Morales-Menendez, «Machine learning techniques for quality control in high conformance manufacturing environment,» *Machine Learning in Automotive Industry*, vol. 10, nº 2, 27 Febrero 2018.
- [20] S. P. Potharaju y M. Sreedevi, «A novel subset feature selection framework for increasing the classification performance of SONAR targets,» *Procedia Computer Science*, vol. 125, pp. 902-909, Enero 2018.
- [21] F. Tao, Q. Qi, A. Liu y A. Kusiak, «Data-driven smart manufacturing,» *Journal of Manufacturing Systems*, vol. 48, nº Part C, pp. 157-169, Julio 2018.
- [22] F. Schäfer, C. Zeiselmaier, J. Becker y H. Otten, «Synthesizing CRISP-DM and quality management: a data mining approach for production processes,» de *2018 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD)*, Marrakech, Morocco, 2018.
- [23] J. Lenz, T. Wuest y E. Westkämper, «Holistic approach to machine tool data analytics,» *Journal of Manufacturing Systems*, vol. 48, nº Part C, pp. 180-191, Julio 2018.
- [24] J. Romero, «Metodologías de Minería de Datos,» Jorge Romero, 11 Junio 2019. [En línea]. Available: <https://jorgeromero.net/metodologias-de-mineria-de-datos/>. [Último acceso: 16 Julio 2020].
- [25] Rohit, B. Gupta, R. Kumar y A. Kumar, «Towards information discovery on large scale data: state-of-the-art,» de *International Conference on Soft-computing and Network Security (ICSNS)*, Coimbatore, India, 2018.
- [26] D. Mishra, R. B. Roy, S. Dutta, S. K. Pal y D. Chakravarty, «A review on sensor based monitoring and control of friction stir welding process and a roadmap to Industry 4.0,» *Journal of manufacturing processes*, vol. 36, pp. 373-397, Diciembre 2018.
- [27] F. C. Peralta, «Proceso de conceptualización del entendimiento del negocio para proyectos de explotación de información,» *Revista Latinoamericana de Ingeniería de Software*, vol. 2, nº 273, pp. 273-306, 2014.
- [28] J. Huang, Q. Chang y N. Chakraborty, «Machine preventive replacement policy for serial production lines based on reinforcement learning.,» de *15th International Conference on Automation Science and Engineering (CASE)*, Vancouver, 2019.
- [29] N. Pröllochs y S. Feuerriegel, «Reinforcement Learning in R,» 29 Septiembre 2018. [En línea]. Available: <https://arxiv.org/abs/1810.00240>. [Último acceso: 4 Junio 2020].
- [30] H. Zhang y L. J. Y. Zhang, «Overfitting and Underfitting Analysis for Deep Learning Based End-to-end Communication Systems,» de *11th International Conference on Wireless Communications and Signal Processing (WCSP)*, Xian, China, 2019.
- [31] H. K. Jabbar y R. Z. Khan, «Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study),» Aligarh, India, 2014.
- [32] A. Amidi y S. Amidi, «CS 229 — Aprendizaje automático,» 6 Octubre 2018. [En línea]. Available: <https://stanford.edu/~shervine/l/es/teaching/cs-229/hoja-referencia-aprendizaje-supervisado>. [Último acceso: 15 Junio 2020].
- [33] S. Saxena, «Underfitting vs. Overfitting (vs. Best Fitting) in Machine Learning,» 7

- Febrero 2020. [En línea]. Available: <https://www.analyticsvidhya.com/blog/2020/02/underfitting-overfitting-best-fitting-machine-learning/>. [Último acceso: 22 Junio 2020].
- [34] FAYRIX, «Selección de métricas para aprendizaje automático,» [En línea]. Available: https://fayrix.com/machine-learning-metrics_es. [Último acceso: 24 Junio 2020].
- [35] D. Mishra, «Regression: An explanation of regression metrics and what can go wrong,» Medium, 6 Diciembre 2019. [En línea]. Available: <https://towardsdatascience.com>. [Último acceso: 28 Junio 2020].
- [36] L. González, «Evaluando el error en los modelos de regresión,» 23 Noviembre 2018. [En línea]. Available: <https://ligdigonzalez.com>. [Último acceso: 28 Junio 2020].
- [37] T. P. S. University, «PennState Eberly College of Science,» [En línea]. Available: <https://online.stat.psu.edu/stat462/node/79/>. [Último acceso: 11 Junio 2020].
- [38] [En línea]. Available: <https://es.mathworks.com/help/stats/regress.html#References>. [Último acceso: 21 Septiembre 2020].
- [39] Benai, «Polynomial Regression — explained,» Medium, 8 Junio 2020. [En línea]. Available: <https://medium.com/next-gen-machine-learning/polynomial-regression-7f27fae98197>. [Último acceso: 11 Junio 2020].
- [40] Shalabh, «Regression Analysis,» Indian Institute of Technology Kanpur, [En línea]. Available: <http://home.iitk.ac.in/~shalab/course5.htm>. [Último acceso: 11 Junio 2020].
- [41] D. Jurafsky y J. H. Martin, «Stanford: Speech and Language Processing,» 16 Octubre 2019. [En línea]. Available: <https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf>. [Último acceso: 26 Junio 2020].
- [42] C. Qi y X. Tang, «Slope stability prediction using integrated metaheuristic and machine learning approaches: A comparative study,» *Computers – Industrial Engineering*, vol. 118, pp. 112-122, Abril 2018.
- [43] C. Reinders, H. Ackermann, M. Ying Yang y B. Rosenhahn, «Chapter 4 - Learning convolutional neural networks for object detection with very little trainint data,» de *Multimodal Scene Understanding*, Academic Press, 2019, pp. 65-100.
- [44] A. Sharma, «Decision tree vs random forest - Which algorithm should you use?,» Analytics Vidhya, 12 Mayo 2020. [En línea]. Available: <https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/>. [Último acceso: 21 Septiembre 2020].
- [45] I. Pavlenko, J. Trojanowska, V. Ivanov y O. Liaposchenko, «Scientific and Methodological Approach for the Identification of Mathematical Models of Mechanical Systems by Using Artificial Neural Networks,» de *Innovation, Engineering and Entrepreneurship. HELIX 2018*, Cham, 2018.
- [46] F. Sancho Caparrini, «Redes Neuronales: una visión superficial,» 14 Diciembre 2019. [En línea]. Available: <http://www.cs.us.es/~fsancho/?e=72>. [Último acceso: 18 marzo 2020].
- [47] Q. Xiao, C. Li, Y. Tang y X. Chen, «Energy Efficiency Modeling for Configuration-Dependent Machining via Machine Learning: A Comparative Study,» *IEEE Transactions on Automation Science and Engineering*, pp. 1-14, 14 Enero 2020.
- [48] Z. Ti, X. Wei Deng y H. Yang, «Wake modeling of wind turbines using machine learning,» *Applied Energy*, vol. 257, 1 Enero 2020.
- [49] Y. Lei, B. Yang, X. Jiang, N. Li y A. K. Nandi, «Applications of machine learning

- to machine fault diagnosis: A review and roadmap,» *Mechanical Systems and Signal Processing*, vol. 138, Abril 2020.
- [50] J. Dean, *Big Data, Data Mining, and Machine Learning: Value Creation for Business Leaders and Practitioners*, WILEY, 2014.
- [51] I. Zajačko, T. Gál, Z. Ságová, V. Mateichyk y D. Wiecek, «Application of artificial intelligence principles in mechanical engineering,» *Innovative Technologies in Engineering Production (ITEP'18)*, vol. 244, Diciembre 2018.
- [52] ujjwalkarn, «An Intuitive Explanation of Convolutional Neural Networks,» 11 Agosto 2016. [En línea]. Available: <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>. [Último acceso: 20 Junio 2020].
- [53] S. Nasiriany, G. Thomas, W. Wang y A. Yang, «A comprehensive guide to machine learning,» 2019.
- [54] D. D. Ciresan, U. Meier, J. Masci, L. M. Gambardella y J. Schmidhuber, «Flexible, high performance convolutional neural networks for image classification,» de *IJCAI'11: Proceedings of the Twenty-Second international joint conference on Artificial Intelligence*, 2011.
- [55] P. Wang, Ananya, R. Yan y R. X. Gao, «Virtualization and deep recognition for system fault classification,» *Journal of Manufacturing Systems*, vol. 44, n° Part 2, pp. 310-316, Julio 2017.
- [56] F. Jia, Y. Lei, J. Lin, X. Zhou y L. Na, «Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data,» *Mechanical Systems and Signal Processing*, Vols. %1 de %272-73, pp. 303-315, Mayo 2016.
- [57] d. Cica, B. Sredanovic, S. Teic y D. Kramar, «Predictive modeling of turning operations under different cooling/lubricating conditions for sustainable manufacturing with machine learning techniques,» *Applied Computing and Informatics*, 13 Febrero 2020.
- [58] D. P. Penumuru, S. Muthuswamy y P. Karumbu, «Identification and classification of materials using machine vision and machine learning in the context of industry 4.0,» *Journal of Intelligent Manufacturing*, vol. 31, 16 Noviembre 2019.
- [59] «El algoritmo k-means aplicado a clasificación y procesamiento de imágenes,» [En línea]. Available: https://www.uniovi.es/compnum/laboratorios_py/kmeans/kmeans.html. [Último acceso: 7 Junio 2020].
- [60] C. Piech, «K Means,» Stanford CS221, 2013. [En línea]. Available: <https://stanford.edu/~cpiech/cs221/handouts/kmeans.html>. [Último acceso: 21 Septiembre 2020].
- [61] J. Xiong, Q. Zhang, G. Sun, X. Zhu, M. Liu y Z. Li, «An information fusion fault diagnosis method based on dimensionless indicators with static discounting factor and KNN,» *IEEE Sensors Journal*, vol. 16, n° 7, pp. 2060-2069, Abril 2016.
- [62] A. Widodo y B.-S. Yang, «Support vector machine in machine condition monitoring and fault diagnosis,» *Mechanical Systems and Signal Processing*, vol. 21, n° 6, pp. 2560-2574, Agosto 2007.
- [63] J. Hassan, A. Majid Awan y A. Jalil, «Welding defect detection and classification using geometric features,» de *10th International Conference on Frontiers of Information Technology*, Islamabad, 2012.
- [64] R. Jegadeeswaran y V. Sugumaran, «Fault diagnosis of automobile hydraulic brake system using statistical features and support vector machines,» *Mechanical Systems and Signal Processing*, Vols. %1 de %252-53, pp. 436-446, Febrero 2015.

- [65] N. N. Bath, k. Kumari, S. Dutta, S. Pal y S. Pal, «Friction stir weld classification by applying wavelet analysis and support vector machine on weld surface images,» *Journal of Manufacturing Processes*, vol. 20, nº Part 1, pp. 274-281, Octubre 2015.
- [66] X. Zhang, B. Wang y X. Chen, «Intelligent fault diagnosis of roller bearings with multivariable ensemble-based incremental support vector machine,» *Knowledge-Based Systems*, vol. 89, pp. 56-85, Noviembre 2015.
- [67] O. Janssens, V. Slavkovikj, B. Vervisch, K. Stockman, M. Loccufier, S. Verstock, R. Van de Walle y S. Van de Hoecke, «Convolutional Neural Network Based Fault Detection for Rotating Machinery,» *Journal of Sound and Vibration*, vol. 377, pp. 331-345, 1 Septiembre 2016.
- [68] C. Lu, Z.-Y. Wang, W.-L. Qin y J. Ma, «Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification,» *Signal Processing*, vol. 130, pp. 377-388, Enero 2017.
- [69] P. Wang, Ananya, R. Yan y R. X. Gao, «Virtualization and deep recognition for system fault classification,» *Journal of Manufacturing Systems*, vol. 44, nº Part 2, pp. 310-316, Julio 2017.
- [70] Z. Chen, S. Deng, X. Chen, C. Li, R.-V. Sanchez y H. Qin, «Deep neural networks-based rolling bearing fault diagnosis,» *Microelectronics Reliability*, vol. 75, pp. 327-333, Agosto 2017.
- [71] S. Moreno, M. Peña, A. Toledo, R. Treviño y H. Ponce, «A New Vision-Based Method Using Deep Learning for Damage Inspection in Wind Turbine Blades,» de *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, Mexico City, 2018.
- [72] B.-S. Peng, H. Xia, Y.-k. Liu, B. Yang, D. Guo y S.-M. Zhu, «Research on intelligent fault diagnosis method for nuclear power plant based on correlation analysis and deep belief network,» *Progress in Nuclear Energy*, vol. 108, pp. 419-427, 5 Julio 2018.
- [73] L. Li, K. Ota y M. Dong, «Deep learning for smart industry: Efficient manufacture inspection system with fog computing,» *IEEE Transactions on Industrial Informatics*, vol. 14, nº 10, pp. 4665-4673, Octubre 2018.
- [74] S. Pandarakone Esakimuthu, S. Gunasekara, K. Asano, Y. Mizuno y H. Nakamura, «A Study on Machine Learning and Artificial Intelligence Methods in Detecting the Minor Outer-Raceway Bearing Fault,» de *IEEE International Conference on Industrial Technology (ICIT)*, Melbourne, Australia, 2019.
- [75] J. Wang, Z. Mo, H. Zhang y Q. Miao, «Ensemble diagnosis method based on transfer learning and incremental learning towards mechanical big data,» *Measurement*, vol. 155, Abril 2020.
- [76] O. Yaman, «An automated faults classification method based on binary pattern and neighborhood component analysis using induction motor,» *Measurement*, vol. 168, p. 108323, 15 Enero 2021.
- [77] P. Bilski, «Application of support vector machines to the induction motor parameters identification,» *Measurement*, vol. 51, pp. 377-386, Mayo 2014.
- [78] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone y A. Beghi, «Machine learning for predictive maintenance: a multiple classifier approach,» *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, vol. 11, nº 3, pp. 812-820, Junio 2015.
- [79] Y. Ding, J. Ma y Y. Tian, «Health assessment and fault classification for hydraulic pump based on LR and softmax regression,» *Journal of Vibroengineering*, vol. 17, nº 4, pp. 1805-1816, Junio 2015.

- [80] N. N. Bhat, S. Dutta, S. K. Pal y S. Pal, «Tool condition classification in turning process using hidden Markov model based on texture analysis of machined surface images,» *Measurement*, vol. 90, pp. 500-509, Agosto 2016.
- [81] W. Yuting, Y. Mei, D. Shaopeng, L. Li y L. Yingqi, «Remaining useful life estimation of engineered systems using vanilla LSTM neural networks,» *Neurocomputing*, vol. 275, pp. 167-179, Enero 2018.
- [82] M. Elforjani y S. Shanbr, «Prognosis of bearing acoustic emission signals using supervised Machine Learning,» *IEEE Transactions on Industrial Electronics*, vol. 65, nº 7, pp. 5864-5871, Julio 2018.
- [83] G. R. Moreira, G. J. B. T. Lahr y J. O. C. G. A. P. Savazzi, «Online prediction of threading task failure using Convolutional Neural Networks,» de *RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, 2018.
- [84] G. Niu, S. Tang y B. Zhang, «Machine condition prediction based on long short term memory and particle filtering,» de *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, Washington, DC, USA, 2018.
- [85] I. Amihai, R. Gitzel, K. Muzaffar, D. Pareschi, S. Subbiah y G. Sosale, «An industrial case study using vibration data and Machine Learning to predict asset health,» de *2018 IEEE 20th Conference on Business Informatics*, Viena, Austria, 2018.
- [86] U. Shafi, A. Safi, A. R. Shahid, S. Ziauddin y M. Q. Saleem, «Vehicle remote health monitoring and prognostic maintenance system,» *Journal of Advances TRansportation*, 2018.
- [87] X.-l. Chen, P.-h. Wang, Y.-s. Hao y M. Zhao, «Evidential KNN-based condition monitoring and early warning method with applications in power plant,» *Neurocomputing*, vol. 315, pp. 18-32, 13 Noviembre 2018.
- [88] J. H. Kim, H. S. Lee, J. H. Kim y J. H. Park, «Detection of Boiler Tube Leakage Fault in a Thermal Power Plant Using Machine Learning Based Data Mining Technique,» de *IEEE International Conference on Industrial Technology (ICIT)*, Melbourne, Australia, 2019.
- [89] N. Kolokas, T. Vafeiadis, D. Loannidis y D. Tzovaras, «A generic fault prognostics algorithm for manufacturing industries using unsupervised machine learning classifiers,» *Simulation Modelling Practice and Theory*, vol. 103, p. 102109, Septiembre 2020.
- [90] G. Kant y K. S. Sangwan, «Predictive modelling for energy consumption in machining using artificial neural network,» *CIRPe 2015 - understanding the life cycle implications of manufacturing*, vol. 37, pp. 205-210, 2015.
- [91] D. Weimer, B. Scholz-Reiter y M. Shpitalni, «Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection,» vol. 65, nº 1, pp. 417-420, 2016.
- [92] S. Klos y Patalas-Maliszewska, «An analysis of simulation models in a discrete manufacturing system using artificial neural network,» *Innovation, Engineering and Entrepreneurship. HELIX 2018. Lecture Notes in Electrical Engineering*, vol. 505, pp. 315-322, 3 Junio 2018.
- [93] L. Lingitz, V. Gallina, F. Ansari, D. Gyulai, A. Pfeiffer, W. Sihn y L. Monostori, «Lead time prediction using machine learning algorithms: A case study by a semiconductor manufacturer,» *Procedia CIRP*, vol. 72, pp. 1051-1056, 27 Junio 2018.
- [94] X. Ou, Q. Chang, J. Arinez y Z. Jing, «Gantry Work Cell Scheduling through Reinforcement Learning with Knowledge-guided Reward Setting,» *IEEE Access*, vol. 6, pp. 14699-14709, 2018.

- [95] M. Si, T. J. Tarnoczi y B. M. D. K. Wiens, «Development of predictive emissions monitoring system using open source machine learning library – Keras: a case study on a cogeneration unit,» *IEEE Access*, vol. 7, pp. 113463-113475, 2019.
- [96] J. Romero Hernández, B. Nath Saha, S. Tstutsumi y R. Fincato, «Incorporating domain knowledge into reinforcement learning to expedite welding sequence optimization,» *Engineering Applications of Artificial Intelligence*, vol. 91, p. 103612, Mayo 2020.
- [97] P. D. Paraschos, G. K. Koulina y D. E. Koulouriotis, «Reinforcement learning for combined production-maintenance and quality control of a manufacturing system with deterioration failures,» *Journal of Manufacturing Systems*, vol. 56, pp. 470-483, 14 Julio 2020.
- [98] Y. He, P. Wu, Y. Li, Y. Wang, F. Tao y Y. Wang, «A generic energy prediction model of machine tools using deep learning algorithms,» *Applied Energy*, vol. 275, 1 Octubre 2020.
- [99] Y. Wang, J. Liu y Y. Han, «Production capacity prediction of hydropower industries for energy optimization: Evidence based on novel extreme learning machine integrating Monte Carlo,» *Journal of Cleaner Production*, vol. 272, p. 122824, 1 Noviembre 2020.
- [100] «Machine Learning,» Mathworks, [En línea]. Available: <https://es.mathworks.com/discovery/machine-learning.html>. [Último acceso: 25 Septiembre 2020].
- [101] «Machine Learning en Python,» GRAPH everywhere, [En línea]. Available: <https://www.grapheverywhere.com/machine-learning-en-python/>. [Último acceso: 25 Septiembre 2020].
- [102] «Machine Learning con R,» GRAPH Everywhere, [En línea]. Available: <https://www.grapheverywhere.com/machine-learning-con-r/>. [Último acceso: 26 Septiembre 2020].
- [103] E. P. S. d. d. V. i. I. Geltrú, «Grado en ingeniería mecánica,» [En línea]. Available: <https://www.epsevg.upc.edu/ca/estudis/grau-en-enginyeria-mecanica>. [Último acceso: 24 Septiembre 2020].
- [104] U. P. d. Catalunya, «Normativa de prácticas externas de la Universitat Politècnica de Catalunya,» 10 febrero 2015. [En línea]. Available: <https://www.upc.edu/normatives/ca/butlleti-upc/hemeroteca/2014-2015/butlleti-upc-161/bupc-161-docs/docs-consell-govern/9.26-aprovacio-modificacio-normativa-practiques>. [Último acceso: 24 septiembre 2020].