

UNIVERSIDAD ALFONSO X EL SABIO

Business Tech

Máster Universitario en Inteligencia Artificial



TRABAJO DE FIN DE MASTER

**Diseño Inteligente: Aplicación de IA en el
Prediseño de Motores Eléctricos para
Aeronáutica**

Estudiante: Gonzalo Vargas Tendaro
Director: Guillermo Vieira de Santiago

Junio 2025

RESUMEN

La creciente demanda de eficiencia y reducción de peso en motores eléctricos para aeronáutica ha impulsado la implementación de técnicas inteligentes basadas en Inteligencia Artificial (IA) para optimizar el proceso de prediseño. Este trabajo desarrolla una metodología data-driven que integra simulaciones FEA, modelos predictivos y algoritmos metaheurísticos para el diseño automatizado de Motores Síncronos de Imán Permanente (PMSM) aplicados a actuadores electromecánicos (EMAs). Se presentan resultados comparativos que evidencian mejoras en densidad de potencia, eficiencia energética y reducción de peso respecto a métodos tradicionales. Además, se discuten las implicaciones tecnológicas y futuras extensiones del método para otros tipos de motores y plataformas multidominio. Los resultados validan la viabilidad y ventajas del uso de IA en ingeniería aeronáutica.

PALABRAS CLAVE

Inteligencia Artificial, IA, Motores Eléctricos, PMSM, Actuadores Electromecánicos, EMA, Data-Driven.

ABSTRACT

The growing need for efficiency and weight reduction in electric motors for aeronautics has driven the application of Artificial Intelligence (AI) techniques to optimize the pre-design process. This work develops a data-driven methodology integrating FEA simulations, predictive models, and metaheuristic algorithms for automated design of Permanent Magnet Synchronous Motors (PMSM) in electromechanical actuators (EMAs). Comparative results demonstrate improvements in power density, energy efficiency, and weight reduction compared to traditional methods. Technological implications and future extensions to other motor types and multidomain platforms are discussed. Results validate AI's feasibility and benefits in aeronautic engineering.

KEYWORDS

Artificial Intelligence, Electric Motors, PMSM, Electromechanical Actuators, Data-Driven

CONTENIDO

Diseño Inteligente: Aplicación de IA en el Prediseño de Motores Eléctricos para Aeronáutica	1
1. Introducción	6
1.1. Motores Síncronos de Imán Permanente (PMSM)	7
1.2. Importancia en Actuadores Electromecánicos (EMAs) en aeronáutica	12
1.3. Limitaciones de métodos tradicionales	16
1.4. Objetivos del trabajo	17
2. Marco Teórico	19
2.1. Fundamentos de Inteligencia Artificial	19
2.2. Técnicas de IA en el Diseño de Motores Eléctricos	20
2.3. Metodología Data-Driven	23
3. Datos y Generación de la Base de Conocimiento	26
3.1. Definición de Variables de Diseño y Especificaciones	26
3.2. Generación de Datos mediante Simulación FEA	31
3.3. Exploración y Análisis de Datos (EDA)	33
4. Modelo de Inteligencia Artificial Aplicado	37
4.1. Modelos subrogados y modelos predictores	38
4.2. Algoritmos regresores y predictivos usados	40
4.3. Optimización del Modelo mediante Metaheurísticas	48
5. Procedimiento del Diseño Inteligente	50
5.1. Implementación del Proceso Data-Driven	52
6. Resultados y Análisis	54
6.1. Estructura del dataset final	55
6.2. Comparativa del Desempeño del Modelo	59
6.3. Análisis de Sensibilidad y Correlaciones	63
6.4. Generación de Diseño Óptimo mediante IA	65
6.5. Validación del modelo y comparativa.	67
6.6. Otras Bases de Datos	70
7. Conclusiones	72
7.1. Beneficios de la IA en el Prediseño de Motores	73
7.2. Implicaciones para la Aeronáutica	74

8. Trabajos Futuros	76
8.1. Extensión a Otros Tipos de Motores	76
8.2. Integración de Modelos Multidominio	77
8.3. Desarrollo de Plataformas Automatizadas Completas	78
8.4. Ecosistema de diseño inteligente.....	80
9. Bibliografía	81
ANEXOS	84
ANEXO I: Repositorio.....	84

1. Introducción

La evolución tecnológica hacia sistemas más eficientes y sostenibles ha situado a los motores eléctricos en una posición estratégica clave dentro de múltiples sectores industriales. En particular, en el ámbito aeronáutico, estos motores desempeñan un papel crucial, especialmente en el desarrollo y optimización de actuadores electromecánicos (EMAs). La creciente demanda en términos de rendimiento, eficiencia energética y reducción de peso ha impulsado la investigación en técnicas avanzadas de diseño basadas en inteligencia artificial (IA), con el objetivo de optimizar significativamente el proceso tradicional del prediseño de motores eléctricos.

Los motores eléctricos han adquirido una importancia central en el desarrollo tecnológico actual debido a la creciente necesidad de sistemas más eficientes, sostenibles y compactos. Particularmente, la búsqueda constante de mejorar el rendimiento energético y reducir las emisiones de contaminantes ha llevado a estos motores a un lugar privilegiado en aplicaciones que van desde vehículos eléctricos hasta sistemas aeronáuticos. En este contexto, la inteligencia artificial (IA) emerge como un recurso clave que permite revolucionar el diseño preliminar de motores eléctricos mediante la automatización y optimización del proceso de diseño basado en grandes volúmenes de datos y modelos predictivos avanzados (Wang et al., 2024). [REF_4](#)

La IA aplicada al diseño preliminar de motores eléctricos permite superar las limitaciones inherentes a los métodos tradicionales, caracterizados por enfoques empíricos y dependientes de múltiples iteraciones mediante análisis de elementos finitos (FEA). Estos métodos suelen ser lentos y costosos, especialmente en situaciones donde es necesario evaluar numerosas configuraciones potenciales. En contraste, la implementación de metodologías basadas en datos (Data-Driven) posibilita explorar de forma rápida y sistemática múltiples soluciones, acelerando significativamente los procesos de diseño y permitiendo encontrar configuraciones óptimas con un esfuerzo computacional considerablemente menor (Qiu et al., 2024). [REF_2](#)

La metodología propuesta en este trabajo pretende establecer un marco innovador para el diseño automático de motores eléctricos aplicados al sector

aeronáutico. Para lograrlo, se utilizarán técnicas avanzadas de inteligencia artificial, tales como modelos predictivos basados en redes neuronales artificiales (ANN), regresión por Kriging (GPR) y algoritmos de optimización como Random Forest (RF). Además, se integrarán algoritmos metaheurísticos para la exploración eficiente del espacio de diseño. La utilización conjunta de estas técnicas permite obtener soluciones altamente adaptadas a especificaciones concretas, particularmente en sectores críticos como el aeronáutico, donde aspectos como peso, eficiencia energética y fiabilidad son vitales (Sayeb et al., 2021). [REF_3](#)

Finalmente, el trabajo contribuirá a la creación de una extensa base de datos generada mediante simulaciones de elementos finitos estratégicas, que posteriormente servirá para entrenar y validar modelos predictivos. Esta base de datos, además de facilitar el diseño preliminar automático, constituirá una herramienta valiosa para futuros desarrollos e investigaciones en motores eléctricos. Así, el presente trabajo tiene un enfoque integral, abordando tanto el desarrollo tecnológico inmediato como la creación de recursos y conocimientos que permitan el avance continuo en la optimización y aplicación de motores eléctricos para la industria aeronáutica (Contreras Novoa, 2022). [REF_1](#)

1.1. Motores Síncronos de Imán Permanente (PMSM)

Los Motores Síncronos de Imán Permanente (PMSM) son ampliamente utilizados debido a sus altas prestaciones en términos de eficiencia energética, densidad de potencia y dinámica operacional. Estos motores se caracterizan por poseer imanes permanentes en el rotor, proporcionando una elevada densidad de flujo magnético constante sin requerir energía adicional para la magnetización del rotor. Esta configuración permite mejorar notablemente la eficiencia energética y disminuir las pérdidas térmicas y eléctricas, aspectos fundamentales en aplicaciones aeronáuticas donde la optimización del peso y la eficiencia son críticas. (Contreras Novoa, 2022). [REF_1](#)

1.1.1. Motores eléctricos

Definición:

Un motor eléctrico es una máquina que convierte la energía eléctrica en energía mecánica mediante la interacción de campos magnéticos. Se utilizan en una gran variedad de aplicaciones, desde electrodomésticos hasta sistemas industriales avanzados.

Tipos de motores eléctricos:

Los motores eléctricos se pueden clasificar en varias categorías, entre ellas:

- Motores de corriente continua (DC): Usan corriente continua para su funcionamiento y son comunes en aplicaciones como vehículos eléctricos y sistemas de control de precisión.
- Motores de corriente alterna (AC): Utilizan corriente alterna y pueden ser de dos tipos principales:
 - Motores síncronos: Operan a una velocidad constante determinada por la frecuencia de la corriente de alimentación.
 - Motores asíncronos (o de inducción): Son los más utilizados en la industria debido a su bajo costo y mantenimiento sencillo.

A continuación, se muestra un diagrama donde se describe los tipos de motores más usados en la industria:

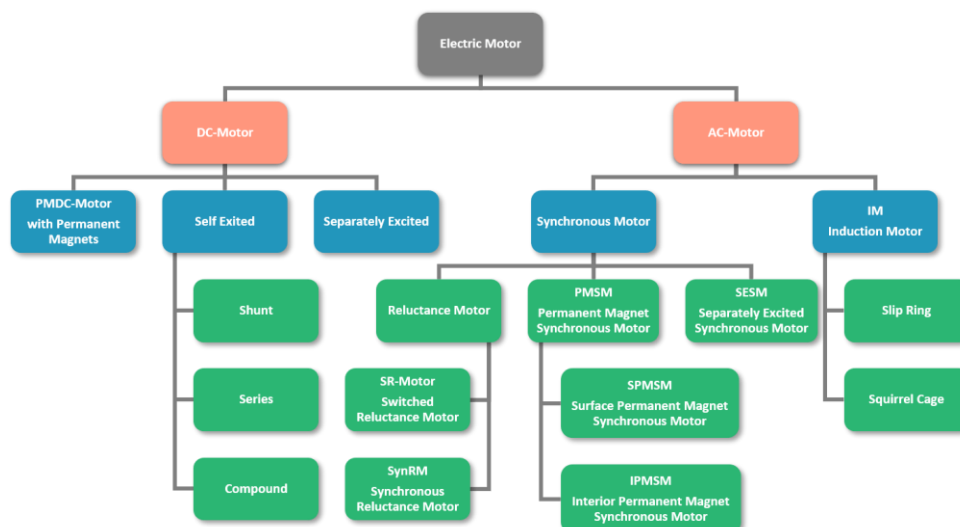


Figura 1: Taxonomía de motores eléctricos. [REF_ 13]

1.1.2. Topología PMSM

Los motores síncronos de imanes permanentes (PMSM) son un tipo de motor eléctrico en el que el campo magnético del rotor es generado por imanes permanentes en lugar de devanados con corriente. A diferencia de los motores de inducción, donde la velocidad del rotor varía ligeramente con respecto a la del campo magnético giratorio, los PMSM operan en sincronía con la frecuencia de alimentación, lo que significa que la velocidad del rotor es exactamente la misma que la velocidad del campo magnético del estator. (Senevirathna, 2020). [REF_14](#)

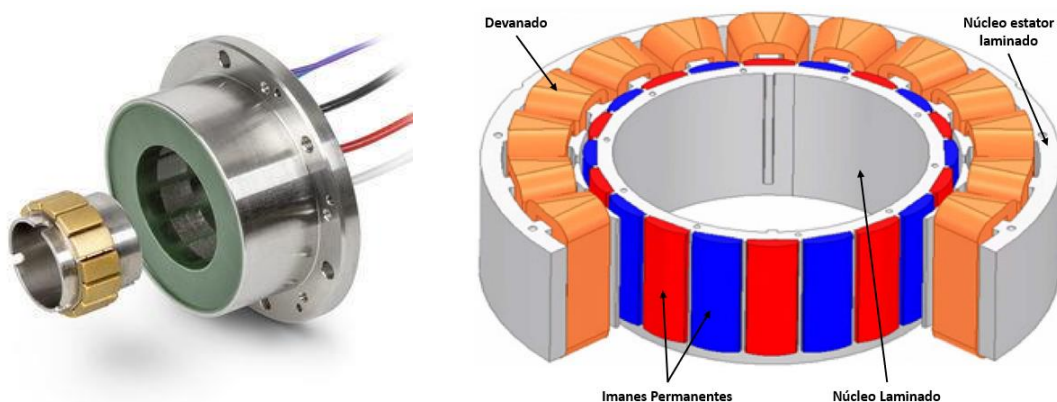


Figura 2: A la izquierda las partes activas de un PMSM. A la derecha un esquema de las partes activas de un PMSM. Izq.: [\[REF_5\]](#). Der.: [\[REF_6\]](#)

Los PMSM se utilizan en diversas aplicaciones industriales y comerciales, incluyendo vehículos eléctricos, robots, sistemas de control de movimiento y compresores de alta eficiencia. Sus principales ventajas incluyen una mayor eficiencia energética, un mejor control del par motor y una densidad de potencia superior en comparación con otros tipos de motores eléctricos. Sin embargo, requieren sistemas de control más avanzados debido a su necesidad de sincronización precisa con la fuente de alimentación.

Existen dos tipos principales de motores PMSM según la disposición de los imanes en el rotor: (Senevirathna, 2020). [REF_14](#)

1. Motores de superficie (SMPMSM - Surface-mounted PMSM): Los imanes están montados en la superficie del rotor. Este diseño facilita la manufactura y permite altas velocidades, pero tiene menor capacidad para soportar esfuerzos mecánicos elevados.

2. Motores de imanes enterrados (IPMSM - Interior PMSM): Los imanes están embebidos dentro del rotor, lo que mejora la robustez mecánica y permite aprovechar el efecto de reluctancia para aumentar el par motor.

Además, los motores PMSM presentan una estructura compacta y robusta que favorece su integración en sistemas aeronáuticos, especialmente en aplicaciones donde las dimensiones y el peso son parámetros restrictivos, como ocurre con los actuadores electromecánicos utilizados en superficies de control y otros sistemas de gestión dinámica del vuelo.

1.1.3. Partes principales de un motor PMSM

Un motor síncrono de imanes permanentes está compuesto por varios elementos fundamentales que contribuyen a su funcionamiento eficiente. (Senevirathna, 2020). [REF_ 14](#)

1. Estátor:

El estátor es la parte fija del motor y se compone de un núcleo laminado y devanados de cobre. Su función principal es generar el campo magnético giratorio cuando se le aplica una corriente alterna trifásica.

- Núcleo laminado: Fabricado en acero al silicio para reducir pérdidas por corrientes parásitas y mejorar la eficiencia magnética.
- Devanados del estator: Dispuestos en forma de bobinas, generan el campo magnético giratorio cuando se alimentan con corriente alterna.

2. Rotor:

El rotor es la parte móvil del motor y contiene los imanes permanentes que generan el campo magnético necesario para la sincronización con el estator. Dependiendo del diseño, los imanes pueden estar montados en la superficie del rotor o embebidos en su estructura.

- Imanes permanentes: Generalmente de neodimio-hierro-boro (NdFeB) debido a su alta densidad de energía magnética. En aeronáutica se suelen usar los de Samario-Cobalto (SmCo) por sus propiedades anticorrosivas y aguante a altas temperaturas.

- Núcleo del rotor: Generalmente fabricado en acero laminado para minimizar pérdidas por corrientes de Foucault.

3. Carcasa y Sistema de Enfriamiento:

La carcasa protege los componentes internos del motor y permite la disipación de calor. Dependiendo de la aplicación, se pueden utilizar sistemas de refrigeración por aire o líquido para mejorar la disipación térmica y evitar sobrecalentamientos.

4. Cojinetes (Rodamientos):

Los cojinetes o rodamientos permiten que el rotor gire suavemente dentro del estator, minimizando la fricción y maximizando la eficiencia mecánica. Deben estar diseñados para soportar tanto cargas radiales como axiales.

5. Encoder, resolver o sensor Hall:

Dado que los motores PMSM requieren un control preciso de la posición del rotor, generalmente incluyen un encoder óptico o sensor de efecto Hall para detectar la posición angular y permitir un control optimizado de la corriente aplicada en el estator.

6. Sistema de Control Electrónico:

El controlador del motor gestiona la conmutación de corriente en los devanados del estator para garantizar que el motor opere en sincronía con la fuente de alimentación. En aplicaciones avanzadas, se utilizan inversores electrónicos con algoritmos de control vectorial (FOC - Field-Oriented Control) para optimizar la eficiencia y el rendimiento del motor.

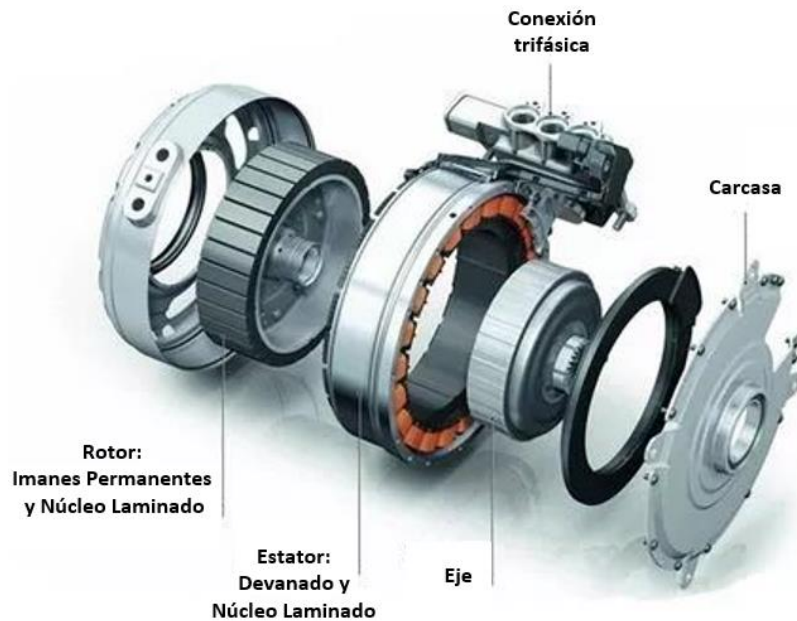


Figura 3: Partes de un PMSM. [REF_ 7]

1.2. Importancia en Actuadores Electromecánicos (EMAs) en aeronáutica

En el contexto del estado del arte para la aplicación de actuadores electromecánicos (EMAs) en aeronáutica, su importancia radica principalmente en el marco del concepto de "More Electric Aircraft" (MEA). Existe una tendencia general en la industria aeroespacial hacia el aumento del uso de equipos alimentados eléctricamente, con el objetivo de redefinir las fuentes y sumideros de energía en el transporte aéreo. MEA puede considerarse una filosofía que busca evolucionar progresivamente hacia un entorno de aeronaves completamente eléctricas "All-Electric Aircraft". (Batlle i Capa, 2020). [REF_ 8](#)

Dentro de esta evolución, los EMAs desempeñan un papel fundamental al reemplazar los actuadores hidráulicos convencionales, especialmente en las superficies de control de vuelo. Los sistemas hidráulicos tradicionales en aeronaves presentan desventajas significativas, como un alto mantenimiento y la necesidad de una infraestructura pesada debido a las numerosas tuberías requeridas para el líquido del actuador. Esto también implica una mayor vulnerabilidad. (Torabzadeh-Tari, 2005). [REF_ 9](#)

La adopción de EMAs, parte de los sistemas "Power-by-Wire" (PBW), ofrece múltiples beneficios que subrayan su creciente importancia en la aeronáutica:

- **Reducción del mantenimiento y de los costos de operación:** Los sistemas hidráulicos convencionales son exigentes en cuanto a mantenimiento, lo que se traduce en altos costos operativos. Los EMAs, al eliminar la necesidad de fluido hidráulico y tuberías extensas, reducen considerablemente las tareas de mantenimiento y los costos asociados. (Batlle i Capa, 2020). [REF_ 8](#)
- **Mayor eficiencia:** Los sistemas eléctricos tienen inherentemente mayores eficiencias que los sistemas hidráulicos convencionales. El aumento de la eficiencia global de la aeronave contribuye a la reducción del consumo de combustible. (Batlle i Capa, 2020). [REF_ 8](#)
- **Reducción del peso:** Los EMAs, al sustituir los sistemas hidráulicos centralizados, contribuyen a la disminución del peso total de la aeronave. El peso es un factor crítico en la operación de aeronaves. (Batlle i Capa, 2020). [REF_ 8](#)
- **Incremento de la tolerancia a fallos y la fiabilidad:** La tecnología PBW y los sistemas EMA están diseñados para ofrecer mayor tolerancia a fallos y fiabilidad en comparación con los sistemas convencionales. Un diseño tolerante a fallos es un requisito básico para los EMAs en aplicaciones MEA. (Torabzadeh-Tari, 2005). [REF_ 9](#)
- **Eliminación de fluidos hidráulicos:** Los fluidos hidráulicos son a menudo contaminantes, inflamables o contaminantes. Los EMAs eliminan la necesidad de estos fluidos. (Berri et al., 2018) [REF_ 10](#)
- **Potencial de regeneración de energía:** Existe la posibilidad de regenerar energía del movimiento de los actuadores de vuelta al sistema de energía. (Torabzadeh-Tari, 2005). [REF_ 9](#)
- **Infraestructura simplificada:** Los sistemas PBW transportan la energía mediante cables en lugar de tuberías hidráulicas, lo que simplifica la infraestructura. (Batlle i Capa, 2020). [REF_ 8](#)
- **Mejora de la maniobrabilidad:** Los sistemas PBW también pueden mejorar la maniobrabilidad de las aeronaves. (Batlle i Capa, 2020). [REF_ 8](#)

Un actuador electromecánico (EMA) típicamente se compone de un motor eléctrico (como un motor síncrono de imanes permanentes - PMSM), un

mecanismo de engranaje, un convertidor de potencia y una unidad de control electrónico. (Quiao et al., 2017). [REF_ 11](#)

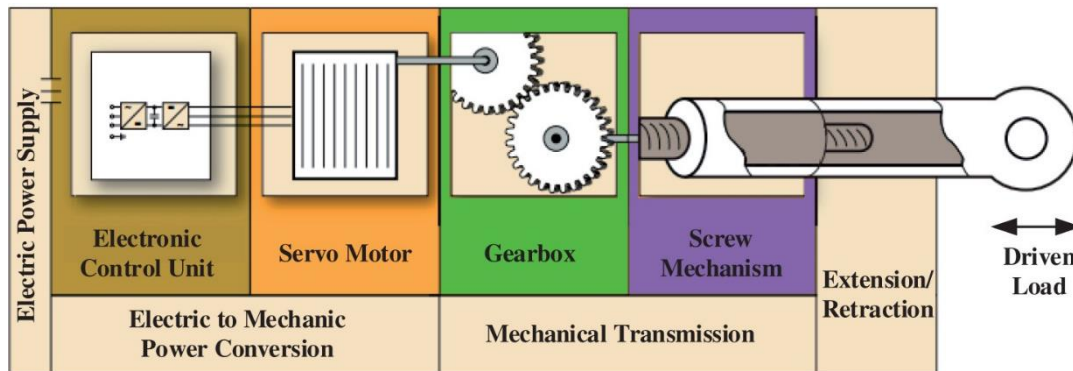


Figura 4: Componentes principales de un actuador electromecánico (EMA). De izquierda a derecha: Unidad de control electrónica, Motor eléctrico, Reductora y Husillo. [\[REF_ 11\]](#)

La aplicación de EMAs en aeronáutica se centra principalmente en los sistemas de actuación de las superficies de control de vuelo. Estas superficies pueden ser primarias (alerones, timón de dirección, timón de profundidad) o secundarias (flaps, slats, spoilers). Aunque históricamente los actuadores hidráulicos o electrohidráulicos han sido la norma, en los últimos años los sistemas electromecánicos han ganado terreno, especialmente en aplicaciones secundarias o como sistemas de respaldo. La tendencia actual apunta hacia su implementación incluso en superficies de control de vuelo primarias, que son críticas para la seguridad y la operación correcta del vuelo. Otros sistemas de la aeronave que también pueden beneficiarse de la electrificación incluyen el tren de aterrizaje, el control vectorial del empuje (TVC) y los sistemas de actuación del motor. (Batlle i Capa, 2020). [REF_ 8](#); (Berri et al., 2018) [REF_ 10](#)

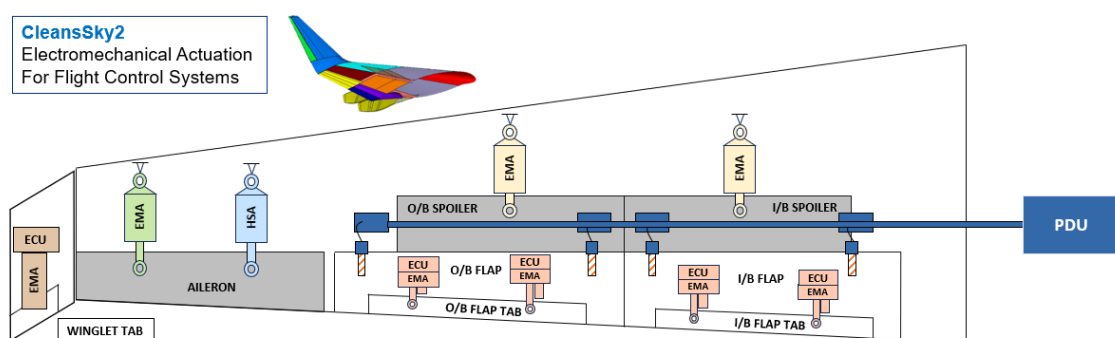


Figura 5: Implementación de actuación electromecánica en aviones de nueva generación dentro del programa CleanSky II. Mandos de vuelo como Flaps, Winglet tab o alerones contienen EMAs como forma de actuación. [\[REF_ 12\]](#)

A pesar de los avances significativos, la implementación generalizada de EMAs en aeronaves, especialmente en superficies de control de vuelo primarias, enfrenta desafíos. La fiabilidad es crucial, particularmente la prevención de fallos como el atasco (jamming) o el embalamiento (runaway), que pueden comprometer el control de la superficie y, por ende, la aeronave. (Batlle i Capa, 2020). [REF_ 8](#)

La gestión térmica y la monitorización de la salud del actuador también son aspectos importantes. Cumplir con las rigurosas normativas de aeronavegabilidad (como las de EASA y FAA) requiere un proceso de certificación exhaustivo. Esto implica la necesidad de una investigación continua, experimentación y desarrollo de modelos matemáticos complejos que permitan simular y predecir el comportamiento de los EMAs en diversas condiciones operativas. (Torabzadeh-Tari, 2005). [REF_ 9](#)

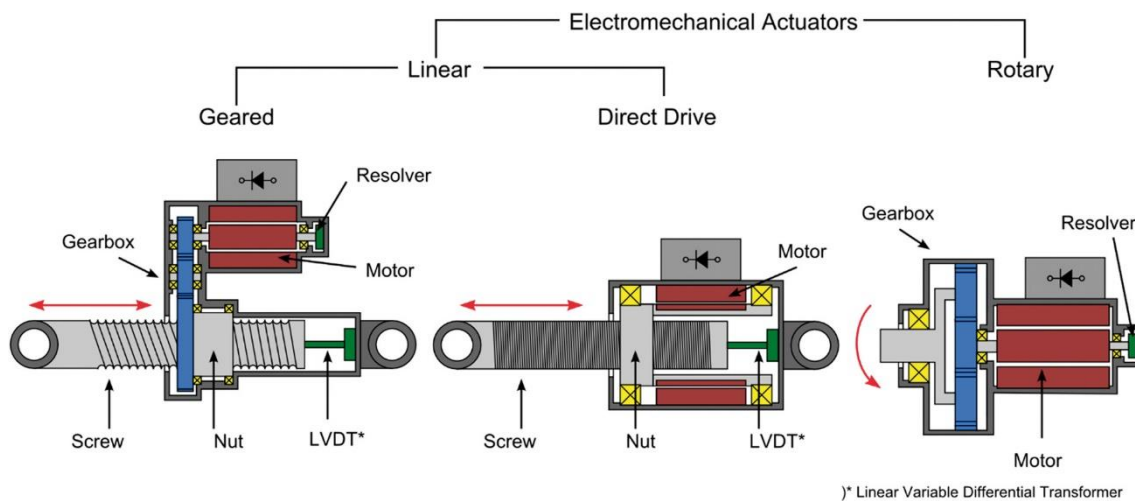


Figura 6: Diferentes topologías de EMA en función del tipo de actuación, lineal o rotativa. [\[REF_ 11\]](#)

En resumen, la importancia de los actuadores electromecánicos en aeronáutica reside en su papel central en la transición hacia aeronaves más eléctricas, ofreciendo ventajas sustanciales en términos de eficiencia, peso, mantenimiento y fiabilidad en comparación con las tecnologías tradicionales. La investigación y el desarrollo en este campo son fundamentales para superar los desafíos técnicos y regulatorios restantes, permitiendo su aplicación segura y efectiva en una gama más amplia de sistemas de aeronaves, incluyendo las superficies de control de vuelo primarias. (Batlle i Capa, 2020). [REF_ 8](#)

1.3. Limitaciones de métodos tradicionales

El diseño óptimo de un motor eléctrico representa un problema no lineal complejo. Tradicionalmente, el análisis del rendimiento de los motores se ha basado principalmente en tres métodos: modelos analíticos (AM), modelos de circuito magnético equivalente (MEC) y análisis de elementos finitos (FEA). Estos enfoques suelen preferirse en etapas específicas del proceso de diseño. A pesar de su utilidad, cada uno presenta limitaciones significativas, especialmente en el contexto de la optimización del diseño. (Cheng et al., 2024) [REF_ 15](#)

El análisis de elementos finitos (FEA) ha surgido como un método de simulación altamente preciso con una notable flexibilidad geométrica. FEA es una herramienta ampliamente adoptada, célebre por su excepcional precisión y flexibilidad geométrica. Sobresale en el diseño de motores complejos y la optimización de campos multifísicos, ofreciendo información tanto del comportamiento electromagnético como mecánico. (Cheng et al., 2024) [REF_ 15](#)

El gran inconveniente de FEA es su costo computacional. Aplicar FEA, especialmente en escenarios que involucran numerosos "individuos" o candidatos de diseño, sigue siendo un proceso computacionalmente intensivo. Cada individuo en un proceso de optimización debe ser resuelto por software FEA para obtener una solución de campo precisa. Esta carga computacional es particularmente desafiante en la optimización de topología debido al vasto número de individuos involucrados. Aunque los algoritmos de optimización inteligentes pueden facilitar la búsqueda del diseño óptimo, la aplicación de FEA para el modelado y la simulación sigue siendo costosa en términos de computación. (Cheng et al., 2024) [REF_ 15](#)

Equilibrar la flexibilidad geométrica, la precisión y el costo computacional es un objetivo difícil de alcanzar, y ninguna de las metodologías tradicionales puede abordar simultáneamente todos estos aspectos de manera efectiva. La gran cantidad de simulaciones individuales requeridas en la optimización tradicional hace que este trabajo sea muy intensivo computacionalmente. (Cheng et al., 2024) [REF_ 15](#)

El diseño preliminar tradicional de motores eléctricos para aeronáutica generalmente depende de un proceso iterativo que combina métodos empíricos, simulaciones por elementos finitos (FEA) y validaciones experimentales. Este enfoque, aunque ha proporcionado resultados

satisfactorios, presenta varias limitaciones significativas: (Qiu et al., 2024). [REF_ 2](#)

- **Dependencia del conocimiento empírico:** Los métodos tradicionales se apoyan en gran medida en la experiencia previa del diseñador, limitando la capacidad para explorar soluciones innovadoras más allá del conocimiento acumulado.
- **Procesos iterativos prolongados:** Las simulaciones y ensayos repetitivos necesarios para ajustar los diseños pueden ser lentos, costosos y propensos a errores acumulativos.
- **Alcance limitado del espacio de diseño:** Debido a restricciones de tiempo y recursos, los métodos convencionales no permiten una exploración exhaustiva del espacio de soluciones posibles, limitando potencialmente la innovación y optimización tecnológica.

Estas limitaciones resaltan la necesidad urgente de metodologías avanzadas, como las basadas en inteligencia artificial, que puedan automatizar, acelerar y optimizar el proceso de diseño preliminar de motores eléctricos. (Qiu et al., 2024). [REF_ 2](#)

1.4. Objetivos del trabajo

El presente trabajo tiene como objetivo principal la implementación y validación de un método inteligente basado en IA para el prediseño automatizado y optimizado de motores eléctricos síncronos de imán permanente (PMSM), específicamente orientado a su aplicación en actuadores electromecánicos aeronáuticos. Concretamente, se pretende:

- **Objetivo 1:**
Desarrollar una metodología data-driven capaz de automatizar el proceso preliminar del diseño mediante técnicas avanzadas de inteligencia artificial.
- **Objetivo 2:**
Reducir significativamente el tiempo y los costes asociados al proceso tradicional de diseño, facilitando la exploración exhaustiva del espacio de diseño.
- **Objetivo 3:**
Proponer modelos predictivos basados en redes neuronales artificiales (ANN) y otras técnicas avanzadas, que permitan obtener predicciones

precisas del comportamiento y rendimiento del motor desde etapas iniciales.

- **Objetivo 4:**

Validar el método mediante la generación y comparación de modelos optimizados frente a diseños convencionales, verificando su eficacia en términos de densidad de potencia, eficiencia energética y peso total del motor.

- **Objetivo 5:**

Establecer una base sólida para futuras investigaciones y desarrollos en plataformas completamente automatizadas que integren modelos multidominio, con el objetivo de extender esta metodología a otros tipos de motores y aplicaciones aeronáuticas.

De esta forma, se espera no solo superar las limitaciones inherentes a los métodos tradicionales, sino también contribuir significativamente a la innovación tecnológica dentro del sector aeronáutico, promoviendo diseños más eficientes, sostenibles y tecnológicamente avanzados.

2. Marco Teórico

El presente capítulo establece las bases conceptuales necesarias para comprender la intersección entre la Inteligencia Artificial y el diseño de motores eléctricos, sirviendo como fundamento para el estudio desarrollado en este trabajo. Se abordan los principios esenciales de la Inteligencia Artificial, las técnicas específicas aplicadas en el contexto del diseño de máquinas eléctricas y, finalmente, la metodología basada en datos que permite integrar estos enfoques.

2.1. Fundamentos de Inteligencia Artificial

La Inteligencia Artificial (IA) se define de diversas maneras, pero en esencia, busca crear sistemas que realicen tareas que, si fueran llevadas a cabo por un ser humano, requerirían inteligencia. Se puede entender también como la parte de la ciencia de la computación que investiga los procesos de razonamiento simbólico y no algorítmico, y la representación de conocimiento para su uso en máquinas inteligentes. En el contexto de la ingeniería y, específicamente, en el diseño de motores eléctricos, la IA se considera una forma diferente de programar computadoras, transitando del procesamiento algorítmico de datos al procesamiento de conocimiento. El término fue oficialmente acuñado en 1956 durante la conferencia de Dartmouth, marcando el inicio formal de esta disciplina. (Vicente & Mateos, 2023) [REF_ 16](#)

A lo largo de su historia, la IA ha explorado diversas aproximaciones metodológicas. Un enfoque significativo ha sido la IA simbólica, que se basa en la creación de representaciones de alto nivel, legibles para el ser humano, de problemas, lógica y conocimiento. Este enfoque contrasta con otros que se centran en aprender patrones directamente de los datos. Independientemente del enfoque, el objetivo subyacente es que las máquinas puedan adoptar comportamientos humanos como el aprendizaje, la toma de decisiones o la resolución de problemas sin necesidad de ser programadas explícitamente para cada tarea. (Vicente & Mateos, 2023) [REF_ 16](#)

Las áreas de trabajo de la IA son variadas y buscan replicar o emular diversas capacidades consideradas inteligentes. Entre ellas se encuentran la representación del conocimiento, la resolución de problemas mediante métodos de búsqueda, la planificación, el tratamiento del lenguaje natural, el razonamiento automático, la creación de sistemas basados en conocimiento,

la percepción (como el reconocimiento de imágenes o patrones), y el autoaprendizaje o aprendizaje automático (Machine Learning). Este último, junto con el aprendizaje profundo (Deep Learning), son ramas fundamentales dentro del amplio campo de la IA, enfocadas en la capacidad de las máquinas para aprender a partir de la experiencia y los datos. La Figura 7 ilustra la relación jerárquica donde el Aprendizaje Automático y el Aprendizaje Profundo se sitúan como subcampos dentro de la Inteligencia Artificial. (Arán, 2022) [REF_ 17](#)

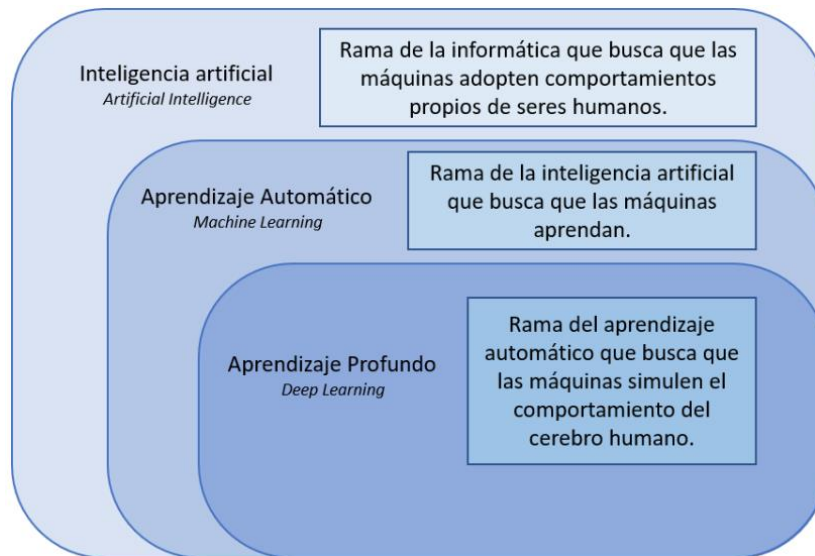


Figura 7: Diagrama de conocimiento sobre Inteligencia Artificial. [[REF_ 17](#)]

Estos fundamentos de IA, apoyados por conceptos matemáticos como la estadística, el álgebra lineal y la optimización permiten a los sistemas de IA analizar datos, reconocer patrones, y generar o evaluar diseños de motores de manera mucho más rápida y eficiente que los métodos tradicionales. (Vicente & Mateos, 2023) [REF_ 16](#)

2.2. Técnicas de IA en el Diseño de Motores Eléctricos

El diseño óptimo de un motor eléctrico es un problema complejo y no lineal. Los métodos tradicionales, como los modelos analíticos, los circuitos magnéticos equivalentes o el análisis por elementos finitos (FEA), presentan limitaciones significativas, especialmente en tareas de optimización que requieren la evaluación de un gran número de candidatos de diseño. El FEA, aunque preciso y flexible, es computacionalmente muy costoso, haciendo que los procesos de optimización tradicionales sean muy intensivos en cálculo.

Esta dificultad es la que ha impulsado la exploración de nuevos enfoques. (Cheng et al., 2024) [REF_ 15](#)

La IA emerge como un pilar de apoyo robusto a lo largo del ciclo de vida de los motores eléctricos, desde el diseño hasta el control y el mantenimiento. Un cambio de paradigma clave es la adopción de metodologías basadas en datos. Aquí es donde los modelos sustitutos (surrogate models), construidos utilizando técnicas de IA, juegan un papel fundamental. Estos modelos basados en datos emplean algoritmos para identificar la relación entre las variables de entrada (parámetros de diseño) y las variables de salida (rendimiento, densidad de potencia, etc.), permitiendo una evaluación rápida y reduciendo el costo computacional en comparación con simulaciones directas como FEA. Un ejemplo de su aplicación es la creación de bases de datos de "expertos" en IA para guiar el diseño preliminar rápido. (Qiu et al., 2024). [REF_ 2](#)

Las técnicas de IA relevantes en este campo incluyen:

- **Machine Learning (ML):** Es un componente central de estos enfoques basados en datos. Permite a los sistemas aprender y mejorar a partir de la experiencia sin ser programados explícitamente. (Cheng et al., 2024) [REF_ 15](#)
- **Deep Learning (DL):** Una rama del ML que utiliza Redes Neuronales (NN) con múltiples capas. Las Redes Neuronales Convolucionales (CNN), por ejemplo, son efectivas para procesar entradas basadas en imágenes (como secciones transversales de motores) y predecir respuestas complejas como el par de cogging. Conceptos fundamentales para entrenar estas redes incluyen funciones de activación, retropropagación (backpropagation) y optimización basada en gradiente descendente. (Cheng et al., 2024) [REF_ 15](#)
- **Aprendizaje Supervisado:** Los modelos aprenden a mapear entradas a salidas deseadas basándose en un conjunto de datos etiquetado útil para tareas como regresión (predecir un valor continuo) o clasificación (categorizar un diseño). (Qiu et al., 2024). [REF_ 2](#)
- **Aprendizaje No Supervisado:** Busca encontrar patrones o estructuras ocultas en datos sin etiquetas aplicable en clustering o reducción de dimensionalidad. (Qiu et al., 2024). [REF_ 2](#)
- **Aprendizaje por Refuerzo:** Implica agentes que aprenden a tomar secuencias de decisiones interactuando con un entorno para maximizar

una recompensa. Se ha explorado para el diseño automático. (Qiu et al., 2024). [REF_ 2](#)

- **Lógica Difusa:** Permite el razonamiento aproximado y el manejo de la incertidumbre, siendo útil en sistemas de control o diagnóstico. (Qiu et al., 2024). [REF_ 2](#)
- **Métodos de Optimización Inteligente:** A menudo utilizados en conjunto con modelos sustitutos, estos algoritmos exploran el espacio de diseño para encontrar soluciones que optimicen uno o varios objetivos (ej. maximizar eficiencia, minimizar volumen). Incluyen algoritmos como los Algoritmos Genéticos (GA), la Optimización por Enjambre de Partículas (PSO), y otros algoritmos metaheurísticos. Estos métodos son aplicados para la optimización multiobjetivo y la aceleración del diseño. (Qiu et al., 2024). [REF_ 2](#)

La Figura 8 ilustra la integración de las metodologías de IA en el ciclo de vida de los motores eléctricos, mostrando cómo estas técnicas fundamentales son aplicadas en áreas como el diseño, el control y el mantenimiento. (Qiu et al., 2024). [REF_ 2](#)

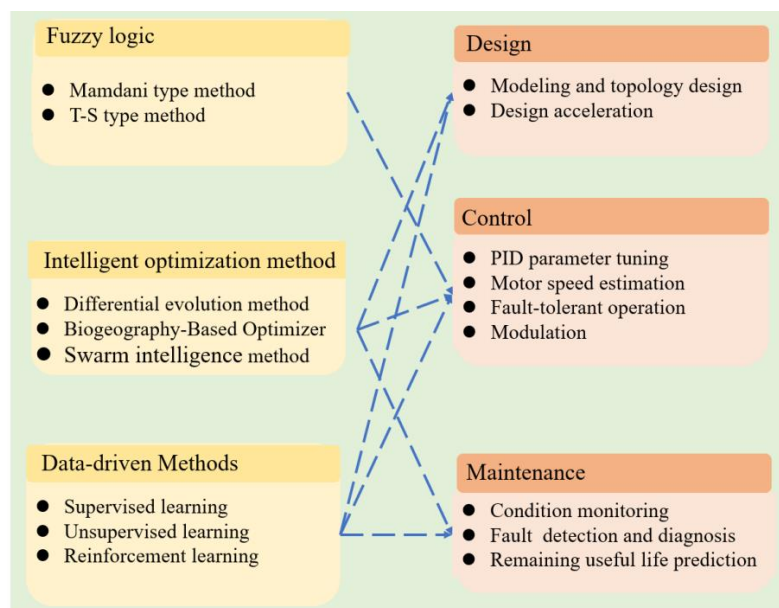


Figura 8: Aplicación de la Inteligencia Artificial en el ciclo de vida de un motor eléctrico. [[REF_ 2](#)]

La integración de estas técnicas de IA permite abordar diversas tareas en el proceso de diseño de motores eléctricos: (Qiu et al., 2024). [REF_ 2](#)

1. **Modelado y Diseño de Topología:** La IA, especialmente el DL, se aplica en el modelado complejo y la optimización de la topología del motor.

2. **Aceleración del Diseño:** Los modelos sustitutos basados en ML/DL reducen drásticamente el tiempo necesario para evaluar iteraciones de diseño.
3. **Diseño Preliminar Automático:** Se han desarrollado marcos basados en datos e IA para generar diseños preliminares directamente a partir de especificaciones, guiados por "bases de datos expertas" creadas previamente con IA y optimización.
4. **Predicción del Rendimiento:** Las redes neuronales y otros modelos de ML pueden predecir métricas clave de rendimiento, como el par, la eficiencia o las pérdidas, a partir de los parámetros geométricos y de materiales.

En resumen, la IA, a través de sus diversas técnicas como ML y DL, y la aplicación de modelos sustitutos basados en datos, ofrece una solución prometedora para superar las limitaciones computacionales de los métodos tradicionales en el diseño y optimización de motores eléctricos, permitiendo procesos más rápidos y eficientes basados en el análisis de datos y la automatización. (Qiu et al., 2024). [REF_2](#)

2.3. Metodología Data-Driven

La metodología data-driven (basada en datos) representa una aproximación contemporánea y eficaz en el diseño de máquinas eléctricas, que se distingue de los métodos tradicionales. Este enfoque se fundamenta en la utilización intensiva de información obtenida a partir de diversas fuentes de datos para guiar, optimizar y validar el proceso de diseño. (Wang et al., 2024) [REF_4](#)

El corazón de la metodología data-driven reside en la adquisición y procesamiento de conjuntos de datos que encapsulan el comportamiento y las características de diseño de máquinas eléctricas. Una fuente primordial de estos datos son los resultados detallados de simulaciones basadas en el método de elementos finitos (FEM). Estas simulaciones generan información rica y precisa sobre el rendimiento electromagnético, térmico y mecánico de configuraciones de diseño específicas bajo diversas condiciones operativas. Almacenar y organizar los resultados de múltiples simulaciones, variando los parámetros de diseño, crea un conjunto de datos experimental digital invaluable para el proceso data-driven. (Wang et al., 2024) [REF_4](#)

Adicionalmente a los datos generados por simulación, la metodología data-driven puede nutrirse de la experiencia acumulada y el conocimiento experto codificado en bases de datos de diseños de máquinas eléctricas existentes. Estos conjuntos de datos pueden incluir una amplia gama de parámetros de diseño (como dimensiones del estator y rotor, número de ranuras, características de los materiales magnéticos y conductores, parámetros del bobinado) y las métricas de rendimiento asociadas obtenidas de pruebas o cálculos. (Wang et al., 2024) [REF_ 4](#)

El propósito fundamental de reunir estos datos es permitir que los algoritmos de inteligencia artificial, particularmente aquellos dentro del campo del aprendizaje automático (Machine Learning), descubran patrones, relaciones y dependencias no lineales entre los parámetros de entrada del diseño y las características de rendimiento de salida. Al "aprender" de estos ejemplos pasados, las máquinas pueden desarrollar modelos predictivos capaces de estimar el rendimiento de un nuevo diseño sin necesidad de recurrir a simulaciones completas o cálculos analíticos detallados para cada iteración. Esto se alinea con la idea fundamental del aprendizaje automático: permitir que una máquina aprenda automáticamente de los datos y mejore su rendimiento a partir de sus experiencias. (Arán, 2022) [REF_ 17](#)

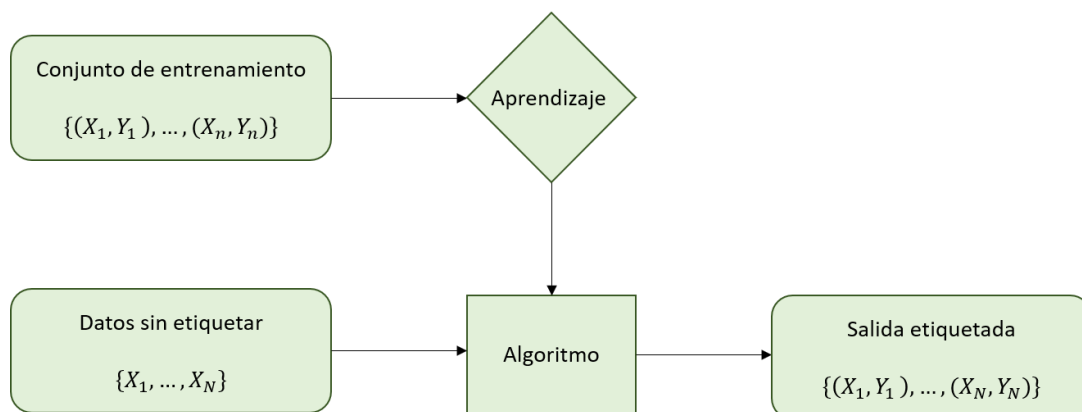


Figura 9: Proceso de aprendizaje automático mediante el uso de datos. [\[REF_ 17\]](#)

Estos modelos predictivos, a menudo implementados como modelos sustitutos o metamodelos, actúan como representaciones computacionales ligeras del comportamiento físico de la máquina. Una vez entrenados con los datos disponibles, estos modelos sustitutos pueden proporcionar evaluaciones de rendimiento rápidas y precisas para un gran número de candidatos de diseño. Esta capacidad de evaluación rápida es esencial para explorar

eficientemente el vasto espacio de diseño y encontrar configuraciones óptimas que satisfagan los requisitos de rendimiento y restricciones, un proceso que sería computacionalmente prohibitivo con los métodos tradicionales basados en simulaciones intensivas. (Wang et al., 2024) [REF_4](#)

En esencia, la metodología data-driven en el diseño de motores eléctricos transforma grandes cantidades de datos de simulación y diseños previos en conocimiento accionable. Este conocimiento es utilizado por algoritmos de aprendizaje automático para crear herramientas predictivas y modelos sustitutos que aceleran significativamente el proceso de diseño preliminar y la optimización, facilitando la identificación de soluciones de diseño superiores. (Wang et al., 2024) [REF_4](#)

3. Datos y Generación de la Base de Conocimiento

Este apartado detalla los datos esenciales que sustentan la metodología data-driven aplicada al prediseño de motores eléctricos aeronáuticos, describiendo cómo se generan y el tratamiento básico que requieren.

La integración de la Inteligencia Artificial (IA) para optimizar el diseño preliminar de motores eléctricos se basa en un cambio de paradigma hacia enfoques basados en datos. Para que estos sistemas de IA operen eficazmente, es indispensable contar con conjuntos de datos extensos, robustos y de alta calidad. Esta metodología implica el uso intensivo de información recopilada de diversas fuentes para guiar, optimizar y validar el proceso de diseño. (Qiu et al., 2024). [REF_2](#)

3.1. Definición de Variables de Diseño y Especificaciones

El fundamento de la metodología data-driven reside en la adquisición y procesamiento de datos que capturen el comportamiento y las características de diseño de las máquinas eléctricas. El proceso de diseño de un motor síncrono de imanes permanentes (PMSM) para aplicaciones aeronáuticas parte de una especificación técnica. Esta especificación define los parámetros fijos del sistema \bar{S} y los límites, condiciones de contorno, \bar{B} . A partir de aquí, se definen las variables de diseño, que se clasifican en geométricas \bar{X} y conducidas \bar{M} . La relación entre el rendimiento del motor (representado por \bar{S} o indicadores de rendimiento \bar{P}) y estas variables de diseño (\bar{X}, \bar{M}) es una función no lineal. (Wang et al., 2024) [REF_4](#)

$$S = F(X, M)$$

En resumen $\bar{S}, \bar{B}, \bar{X}, \bar{M}, \bar{P}$ son todas las matrices que describen el sistema que se definirán individualmente a continuación. También veremos su relación con el proceso de modelado y selección de los mejores candidatos de motor.

3.1.1. Variables de salida, \bar{P}

La especificación \bar{S} se descompone en indicadores de rendimiento o **variables de salida** $\bar{P} = \{P_{out}, \eta, W, \rho_p, T_{nom}, n_{nom}\}$, donde:

Tipo de parámetro	Descripción del parámetro	Símbolo	Unidades	Rango
P1	Potencia nominal	P_{out}	W	[150, 500]
P2	Eficiencia global	η	%	[90, 99]
P3	Peso total del motor	W	kg	< 1
P4	Densidad de potencia	ρ_p	W/kg	> 500
P5	Par nominal	T_{nom}	Nm	[0.15, 0.25]
P6	Velocidad nominal	n_{nom}	rpm	[3500, 4000]

Tabla 1: Descripción de las condiciones de los indicadores de rendimiento o variables de salida, obtenidos a partir de una especificación aeronáutica.

Estas variables de salida serán los objetivos a alcanzar por nuestra simulación y los con los que compararemos a los distintos candidatos de motores eléctricos diseñados por el modelo de Inteligencia Artificial.

3.1.2. Condiciones de contorno, \bar{B}

La adecuada definición y acotamiento de estas variables y especificaciones garantiza que la base de conocimiento cubra de forma exhaustiva el espacio de diseño pertinente al dominio aeronáutico, facilitando la exploración masiva en etapas posteriores.

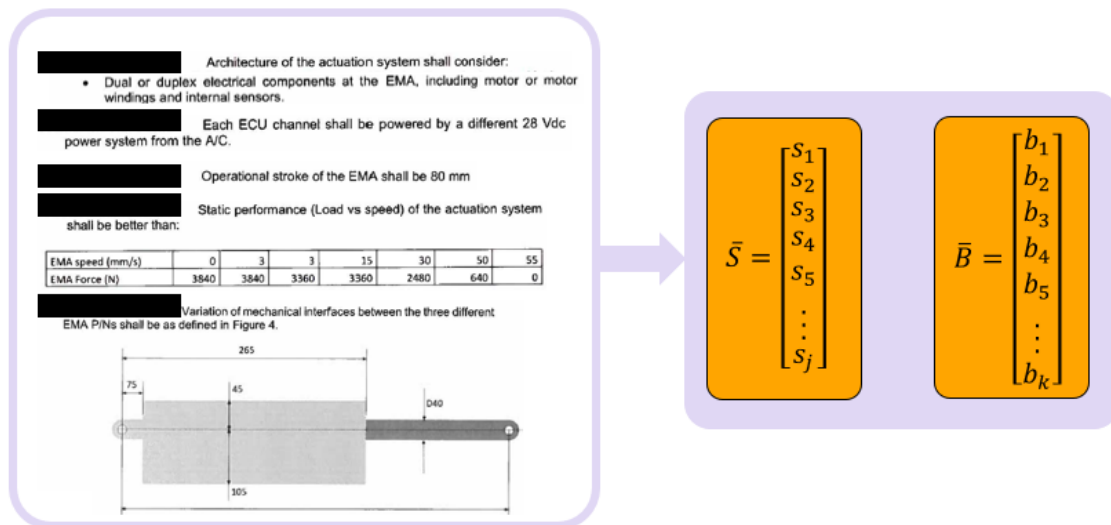


Figura 10: Obtención de la matriz \bar{S} , parámetros obtenidos directamente de la especificación. Obtención de la matriz \bar{B} , parámetros "boundaries" o condiciones de contorno del sistema. Se obtienen a partir de \bar{S} y condiciones derivadas de la especificación que son necesarias para un correcto funcionamiento del sistema. Elaboración propia.

Como se muestra en la Figura 10 partir de la especificación \bar{S} , podemos obtener las condiciones de contorno del sistema. Cada una de ellas está definida en la Tabla 2. Estas condiciones representadas por \bar{B} , pueden ser:

- **Variables geométricas clave:** $\{OSD_{max}, L_{max}, Dsh_{min}, g, GFF_{max}, etc\}$
- **Variables topológicas:** $\{m, Ns, Np, etc\}$
- **Variables electromagnéticas:** $\{V_{BUS}, I_{max}, n_{max}, f_{max}, etc\}$

Donde:

Tipo de parámetro	Descripción del parámetro	Símbolo	Unidades	Valor/Ecuación
B1	Voltaje del BUS	Vbus	Vdc	28
B2	Intensidad máxima del sistema	Imax	Arms	26
B3	Densidad de corriente máxima	Jmax	A/mm ²	12,3
B4	Velocidad máxima permitida	n_max	rpm	10000
B5	frecuencia máxima	f_max	Hz	500
B6	Número de fases	m		3
B7	Número de ranuras	N_slots		12
B8	Número de polos	N_polos		10
B9	Diámetro externo del stator_maximo	OSD_max	mm	60
B10	Diámetro interno del eje mínimo	Dsh_min	mm	8
B11	Airgap	g	mm	0,5
B12	Amplitud en grados del imán	Cmag	°	150
B13	Altura del diente del stator	hso	mm	1
B14	ancho de apertura del slot	bso	mm	1,2
B15	Ángulo del pie del diente del slot	Vs	mm	20
B16	Factor de llenado del slot máximo	GFFmax		55
B17	Factor de llenado del slot Mínimo	GFFmin		20
B18	Diámetro del hilo	dw	mm	0,5105
B19	Masa máxima del motor	Wmax	kg	1
B20	altura del entrediente del slot mínima	he_min	mm	3,5

Tabla 2: Descripción de las condiciones de contorno del sistema a partir de una especificación aeronáutica. Incluye limitaciones en las variables geométricas clave, variables topológicas y variables electromagnéticas.

Estos parámetros garantizan que todos los diseños probados cumplan con las condiciones eléctricas, térmicas y mecánicas requeridas por la normativa aeronáutica y las especificaciones del proyecto.

3.1.3. Variables geométricas, \bar{X}

El conjunto de variables geométricas $\bar{X} = \{x_1, x_2, x_3, \dots, x_n\}$ describe la morfología básica de la máquina eléctrica. A continuación, podemos ver sobre un esquema de motor PMSM qué son cada una de las variables que usaremos.

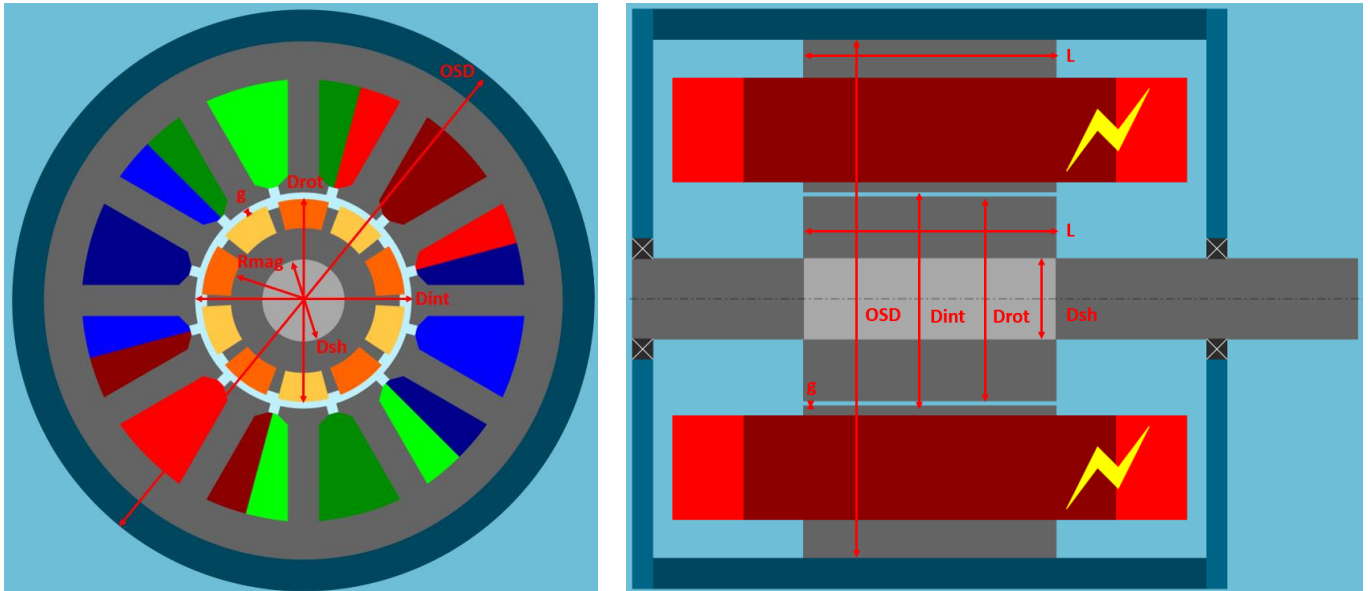


Figura 11: Esquema de un motor PMSM en sus dos secciones principales. Se representan los parámetros que se describen en las distintas matrices. Elaboración propia mediante el programa comercial Flux Motor®.

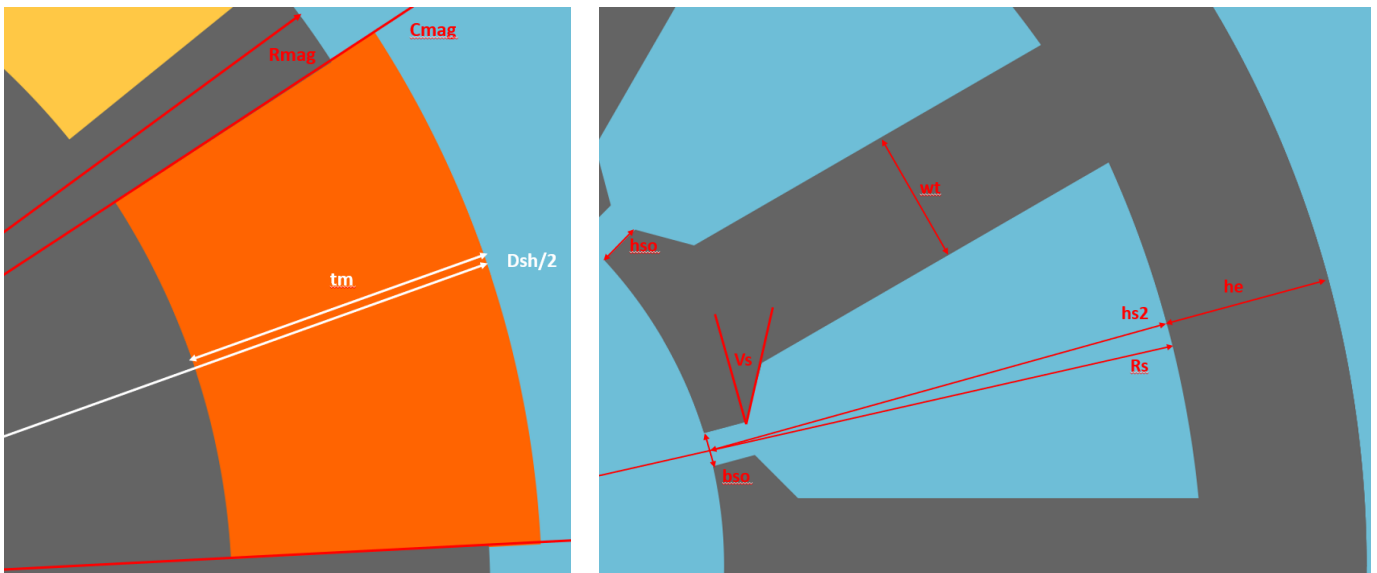


Figura 12: Esquema del sector circular del rotor y el estator. Se representan los parámetros de diseño de los imanes y del laminado del estator que se describen en las distintas matrices. Elaboración propia mediante el programa comercial Flux Motor®.

La selección de estas variables se basa en su influencia directa sobre la densidad de flujo, la rigidez mecánica y el espacio para el devanado. En general, la elección de variables depende de la topología y del conocimiento del profesional especialista. Solo se definen estas variables una vez, no es necesario hacerlo con cada especificación, lo que hace el diseño del modelo de IA mucho más escalable. Lo que sí que podrán variar serán los rangos de diseño. Para el diseño preliminar de un PMSM aeronáutico, se definen ocho variables primarias:

Tipo de parámetro	Descripción del parámetro	Símbolo	Unidades	Valor/Ecuación
X1	Diámetro externo del estator	OSD	mm	[45, 60]
X2	Diámetro interior del estator	Dint	mm	[21, 2, 45]
X3	Longitud del estator	L	mm	[10, 40]
X4	Altura del imán	tm	mm	[2, 3, 5]
X5	Altura del hueco del slot	hs2	mm	[5, 15]
X6	Ancho del Slot	wt	mm	[2, 5]
X7	Número de vueltas	Nt		[5, 30]
X8	Número de cables en la mano	Nh		[3, 9]

Tabla 3: Descripción de las variables geométricas para el diseño de motor.

Estos rangos se establecen en función del baseline machine utilizada como referencia, una máquina real probada experimentalmente, y permiten abarcar un espacio de diseño suficientemente amplio para identificar soluciones óptimas

3.1.4. Variables conducidas, \bar{M}

Las variables conducidas $\bar{M} = \{m1, m2, m3, \dots, m_m\}$ son ecuaciones que dependen de las variables anteriores. Aunque en este caso solo hagan referencia a cálculos geométricos, también pueden usarse cálculos físicos que sean de interés para el modelo de IA.

Tipo de parámetro	Descripción del parámetro	Símbolo	Unidades	Valor/Ecuación
M1	Diámetro externo del rotor	Drot	mm	$D_{int} - 2g$
M2	Diámetro del eje	Dsh	mm	$D_{int} - 2g - 2tm - D_{int}/3.5$
M3	Altura del entrediente del slot	he	mm	$OSD/2 - (D_{int} + 2hs2)/2$
M4	Radio del hueco del imán	Rmag	mm	$(D_{rot}/2) - 0,25 \cdot tm$
M5	Radio de apertura del stack	Rs	mm	$(D_{int}/2) + hs2$
M6	Superficie de conductor en la ranura	CS	mm ²	$Nt \cdot Nh \cdot 2 \cdot \pi \cdot (dw/2)^2$

M7	Superficie del Slot aproximado	SS	mm ²	$(\pi \cdot R_s \cdot R_s - \pi \cdot (D_{int}/2)^2)/N_s - wt \cdot hs2$
M8	Factor de llenado del conductor	GFF	%	$(CS/SS) \cdot 100$

Tabla 4: Descripción de las variables geométricas para el diseño de motor.

Estas funciones garantizan que las condiciones de manufactura y los ajustes geomagnéticos se mantengan dentro de parámetros realistas, evitando diseños inviables y mejorando la convergencia de las simulaciones FEA.

3.2. Generación de Datos mediante Simulación FEA

La generación de datos de alta fidelidad a partir de simulaciones de análisis de elementos finitos (FEA) constituye el pilar central de la metodología data-driven propuesta. Dado el alto coste computacional asociado al FEA, resulta imprescindible diseñar un flujo de trabajo automatizado, robusto y eficiente que permita explorar de manera sistemática el espacio de diseño definido en la sección previa. A continuación, se puede observar el proceso de construcción de las matrices mediante análisis FEA.

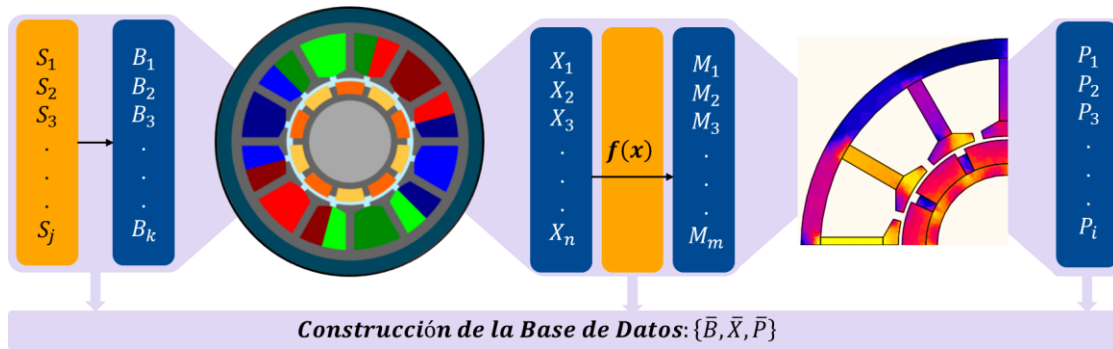


Figura 13: Proceso de obtención de las matrices que conformarán la base de datos de entrenamiento y validación. Se obtienen las variables de salida mediante simulación FEA. Elaboración propia, inspirado en (Wang et al., 2024) [REF_4](#). Uso de imágenes extraídas del programa comercial Flux Motor®.

Para superar la limitación del tiempo computacional mediante FEA y explorar sistemáticamente el espacio de diseño definido por las variables (\bar{X}, \bar{M}) , se emplea un algoritmo de barrido paramétrico. Este algoritmo automatiza el proceso de variación de las variables de diseño de entrada, mediante scripts (en Python), y la ejecución de las simulaciones FEA correspondientes para cada combinación. Se utilizan herramientas comerciales de simulación FEA, como HyperStudy® y Flux® de Altair, para evaluar los parámetros de rendimiento \bar{P} del motor para cada diseño generado por el barrido paramétrico.

Vamos a sintetizar todo este proceso en una serie de pasos que ayudarán a la comprensión de la obtención de la base de datos de entrenamiento:

PASO 1: Generación del motor baseline mediante la herramienta comercial Flux Motor®.

PASO 2: Simulación FEA, del test “Performance Mapping” de la herramienta Flux Motor®. Este text es una simulación electromagnética del motor diseñado y se devuelven variables de interés como el par o la velocidad nominales del motor.

PASO 3: Mediante un linker se exporta toda esta información a HyperStudy®.

PASO 4: Es necesario configurar las herramientas de generación del espacio muestral, así como el solver a usar. Se aplican las relaciones de \bar{M} y las condiciones de contorno de \bar{B} .

PASO 5: Se genera una base de datos de tamaño previamente definido barriendo todo el espacio de diseño mediante algoritmos que seleccionan aleatoriamente valores de los parámetros de diseño siempre dentro de los rangos definidos.

PASO 6: Se seleccionan todos los motores que sean físicamente construibles y los etiquetamos como “Validos”. Esta será nuestra Base de Datos de entrenamiento, y validación. Consideramos estos motores obtenidos mediante simulación FEA como “Reales”.

x1:OSD	x2:Dint	x3:cl	x4:tm	x5:hs2	x6:wt	x7:NI	x8:Nh	m1:Drot	m2:Dsh	m3:che	m4:Imag	m5:Rs	m6:Gff	p1:W	p4:Gff	p5:BSP	p6:BSP	p7:BSP	p8:BSP	p9:BSP	p10:MSP
56,87893763	25,93700204	13,96120952	3,361204727	11,96228495	2,439674026	25	6	24,93700204	10,80402057	3,508682848	11,62819984	24,93078597	95,07857569	51,1252229	87,94738087	0,832889831	2138,09505	234,8146837	74,94670304	15,0991042	5390,9625
58,45400942	21,42394654	38,70528288	2,266803025	10,6353806	2,074551442	15	5	20,42394654	9,769212903	7,879650847	9,645272511	21,34735386	45,59847198	0,918258225	51,01650556	0,885106449	3415,071208	294,2306517	81,82313618	12,58894956	3770,742
56,37381589	32,8914542	10,58488087	2,339747268	7,331313062	4,501579061	14	8	31,8914542	17,81440132	4,409867783	15,36079028	23,77704016	103,5312851	0,46852724	109,7258921	0,679040547	6891,664345	477,956119	87,8221021	20,13477573	9383,6700
54,10305071	22,01387635	22,34227866	2,675007213	6,615435737	3,942007075	5	3	21,01387635	9,374182968	9,429151443	9,838186373	17,62237391	26,07246067	0,494488648	29,13561988	0,126783124	18875,54314	233,679373	91,0554743	7,550785174	19806,403
51,49035901	26,69044665	34,89126358	2,776343081	7,550254221	3,35238564	5	9	25,69044665	12,51191859	4,84970196	12,15113755	20,89547754	43,39186293	0,685863495	50,8552193	0,723745543	8813,116972	666,9911702	92,67897567	22,655243	9910,8301
52,86225432	29,1336714	29,7637481	3,239311276	7,093636609	2,16429898	11	8	28,1336714	13,31114274	4,77065489	15,25700788	21,66047231	69,24095543	0,721728417	84,08815272	1,081436555	5384,715849	572,1678445	88,39419686	20,14084117	5672,7685
56,02912213	31,72485661	32,83938072	3,477796781	7,334358696	2,950543679	20	3	30,72485661	14,7050183	4,817774066	14,49297911	23,196787	46,17029606	0,765447132	59,24104784	0,999424277	1762,917782	202,9853915	80,65918803	7,552715003	1932,1586
50,8523984	25,29235969	29,93390375	2,649398235	7,511686456	3,442394944	5	5	24,29235969	11,76717474	5,248332897	11,48383029	20,1778663	26,32832197	0,548512255	29,99837013	0,315148348	11419,15218	381,3916145	91,39669487	12,58672534	12246,92
55,85029951	23,7375413	26,07529577	3,220873015	9,228777886	3,501741316	10	3	22,7375413	9,514040611	6,827601219	10,5696024	21,09754853	25,89522932	0,573739379	28,08861679	0,314909955	6768,833058	217,6481574	89,14858072	7,552212496	7645,504
54,36546697	23,72652707	23,44788305	3,09582012	5,092143945	3,208229788	16	6	22,72652707	9,328353313	10,227326	10,53588803	16,9540748	177,6206186	0,689848722	126,2036697	0,806368852	2913,842811	337,5824494	83,31297983	15,10192942	4764,4811
59,89418831	25,21070835	26,30335305	2,161891215	6,482993814	4,670316418	29	6	24,21070835	12,68386639	10,85874617	11,56488133	19,08834799	302,3272445	0,967365715	123,6162695	1,197414436	-1229,837087	200,7047312	72,31826595	15,09630961	2701,18659
56,34407941	27,27091641	15,63251367	2,306139494	5,239035847	3,431562166	8	7	26,27091641	13,86694702	9,297545653	12,55892333	18,7449405	85,97534318	0,488515721	96,74696113	0,427093022	11154,94325	507,7545166	92,53394364	17,62017629	13327,326
59,47565723	26,97175206	32,6752239	2,130080378	8,810643627	2,943810448	8	4	25,97175206	14,00537643	7,441308954	12,45335594	22,29651966	23,09920554	0,758719562	25,91520616	0,468482818	5914,294867	299,6611562	98,96324581	10,07128422	6634,16259
56,82648865	22,3085179	19,52746154	2,793725695	8,730449825	4,147115273	29	7	21,3085179	9,347204252	8,528535553	9,955827526	19,88470877	238,7592314	0,80205112	131,5737415	1,053066546	41,4818393	188,9451421	72,12116774	17,61489371	5306,0643
58,52672689	27,8411643	30,51402558	3,065691483	7,765252357	3,283206209	21	4	26,8411643	12,75516296	7,577528938	12,65415928	21,68583451	73,19014005	0,840593739	80,99255867	1,115497844	2004,17761	235,9479383	78,85099496	10,06824621	2429,3104
58,19025467	30,44800289	35,71256023	3,080100234	5,857465973	2,202684107	28	8	29,44800289	14,58837303	8,013659918	13,95397639	21,08146742	213,9699346	1,22329759	125,5589919	1,720179871	-209,444725	341,3618048	75,98190777	20,13785546	221,87002
50,5596643	30,06454651	24,11968226	2,961559747	5,344626636	4,462782911	10	9	29,06454651	14,55155658	4,902932262	13,79188332	20,37698989	143,1141618	0,634153717	123,6920608	1,038835887	4463,441641	591,4077036	89,46456289	22,6514895	5759,5802
51,48971688	24,33505096	16,29025888	3,165117176	9,078992244	4,714191537	25	4	23,33505096	10,05194491	4,98340715	10,87624619	21,24651773	111,565887	0,497390295	96,05876723	0,617164891	1895,48974	76,9191926	10,06338345	4393,5907	
56,8873981	34,11915455	16,69739698	3,284742075	6,673416376	4,40961633	14	7	33,11915455	16,80134053	4,710705399	13,73299365	22,55047366	0,464910657	91,06685346	0,580425699	9018,112066	500,5938503	86,69911034	17,62360993	9871,5344	

Figura 14: Ejemplo de la BD generada a partir de 600 motores simulados mediante FEA. Las columnas son las variables de las matrices X , M y P . Elaboración propia.

Se ha repetido el proceso para generar varias bases de datos variando el número de motores “válidos”. En algunos se ha implementado un algoritmo de optimización para recorrer el espacio de diseño optimizando resultados

respecto a alguna de las variables de salida. Las bases de datos de entrenamiento generadas han sido:

- **BD de 200 motores:**
 - Distribución aplicada: Uniforme y Optimizada.
 - Tiempo medio de cómputo por motor: 17 min
 - Paralelización: 5 motores a la vez
 - Tiempo total de cómputo: 11.3 horas
- **BD de 400 motores:**
 - Distribución aplicada: Uniforme y Optimizada.
 - Tiempo medio de cómputo por motor: 17 min
 - Paralelización: 5 motores a la vez
 - Tiempo total de cómputo: 22.6 horas
- **BD de 600 motores:**
 - Distribución aplicada: Uniforme.
 - Tiempo medio de cómputo por motor: 17 min
 - Paralelización: 10 motores a la vez
 - Tiempo total de cómputo: 17 horas
- **BD de 1000 motores:**
 - Distribución aplicada: Uniforme.
 - Tiempo medio de cómputo por motor: 17 min
 - Paralelización: 10 motores a la vez
 - Tiempo total de cómputo: 28.3 horas
- **BD de 5000 motores:**
 - Distribución aplicada: Uniforme.
 - Tiempo medio de cómputo por motor: 17 min
 - Paralelización: 10 motores a la vez
 - Tiempo total de cómputo: 5 días y 21.7 horas

Estos valores dependerán significativamente de la capacidad de cómputo del servidor en el que se ejecute. Pero como todas las simulaciones se han realizado en el mismo servidor, se puede hacer análisis de escala. El tiempo de cómputo es similar entre la BD de motores uniformemente distribuidos en el espacio de diseño o usando el algoritmo de optimización.

3.3. Exploración y Análisis de Datos (EDA)

NOTA: A partir de este momento se considerará la BD de 5000 motores como el baseline, todo lo explicado a continuación aplica por igual a todas las bases de datos.

Una vez generada la base de datos, el tratamiento básico de los datos y, de manera fundamental, la calidad del dato son aspectos críticos para el éxito de la aplicación de IA. La eficacia de los sistemas de IA para el diseño, control y mantenimiento de motores depende en gran medida de la recopilación, gestión y análisis de grandes volúmenes de datos. Aunque las fuentes no detallan explícitamente todos los pasos de preprocesamiento como la limpieza o el manejo de valores atípicos, enfatizan la necesidad fundamental de implementar estrategias de Gobernanza de Datos (Data Governance).

La Gobernanza de Datos es esencial para garantizar la calidad, integridad, seguridad y disponibilidad de la información. Una gobernanza deficiente, que puede resultar de errores en las mediciones, datos incompletos o inconsistencias (por ejemplo, entre versiones de software), puede llevar a que los modelos de IA generen resultados inexactos. Esto, a su vez, afectaría la eficiencia y confiabilidad de los motores y aumentaría los costos.

El tratamiento básico, guiado por los principios de la Gobernanza de Datos, implica asegurar:

- La calidad de datos mediante estándares para limpiar, filtrar y normalizar.
- La consistencia y trazabilidad de los datos.
- La integridad de datos históricos.
- La interoperabilidad, integrando datos de diversas fuentes (FEA, experimentos, si se consideraran).

Las bases de datos generadas y tratadas, obtenidas mediante el barrido paramétrico y las simulaciones FEA, se convierte en el pilar fundamental para el entrenamiento y validación de los modelos sustitutos (surrogate models). Estos modelos, basados en técnicas de machine learning, aprenden a aproximar la compleja relación no lineal entre las variables de diseño de entrada y el rendimiento del motor utilizando los datos generados. La Exploración y Análisis de Datos (EDA) es una etapa importante en este proceso antes del entrenamiento.

El Análisis Exploratorio de Datos (EDA) implementado en esta investigación se desarrolló mediante un notebook en Python, empleando las librerías pandas, numpy, matplotlib y seaborn, con el fin de garantizar la calidad, consistencia y riqueza informativa del dataset generado en la sección previa. A continuación, se describen en detalle cada una de las fases del EDA.

Se realizaron las siguientes acciones:

1. Carga de datos y exploración inicial:

- Inspección de la estructura. Verificar tipos de datos y presencia de valores nulos.
- Primeros registros. Se comprobaron ejemplos de filas y formatos, confirmando la correcta lectura de los decimales y la integración de las columnas esperadas.

2. Visualización de distribuciones:

- Análisis estadístico. Se obtuvo para cada variable un resumen estadístico con valores como la media, varianza, máximo/mínimo y percentiles.
- Visualización de las distribuciones de cada variable. Se hizo para cada una de las variables de las matrices \bar{X} , \bar{M} , \bar{P} .

Distribuciones del grupo "x"

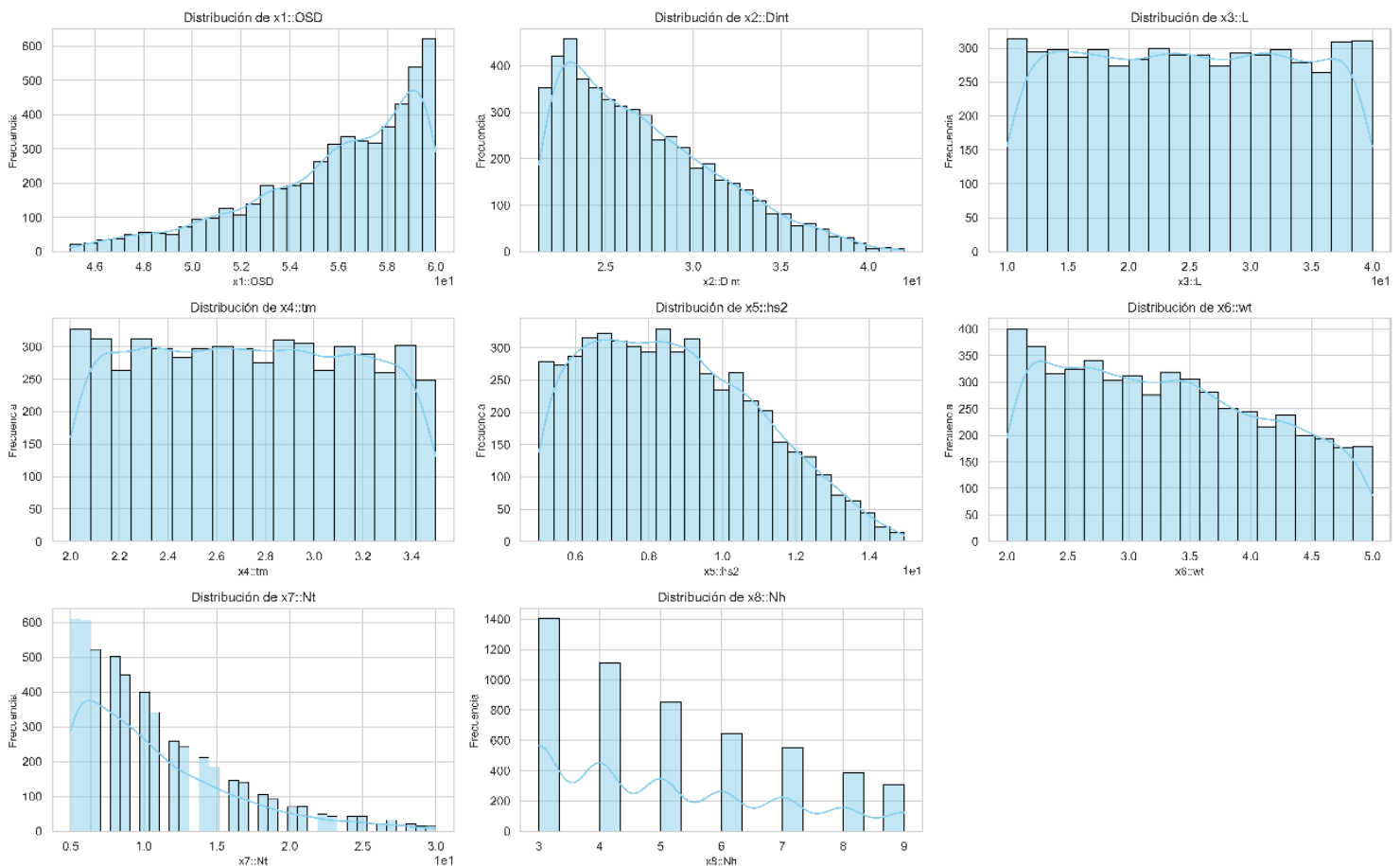


Figura 15: Distribuciones para cada variable de diseño geométrica para la BD de 5000 motores.
Elaboración propia.

3. Filtros, correcciones y optimización de tipo de datos:

- Se han eliminado los motores con valores faltantes o duplicados porque son catalogados como “No válidos”.
- Conversión de tipo de archivo de 64 bits a 32 bits reduciendo el uso de memoria total del dataframe.

4. Conjunto de datos final y estadísticas de post-procesado:

- Las medias y percentiles se mantuvieron dentro de rangos esperados, con ligeros ajustes por la eliminación de casos extremos.
- Se calculó la matriz de correlación de Pearson para evaluar relaciones lineales.

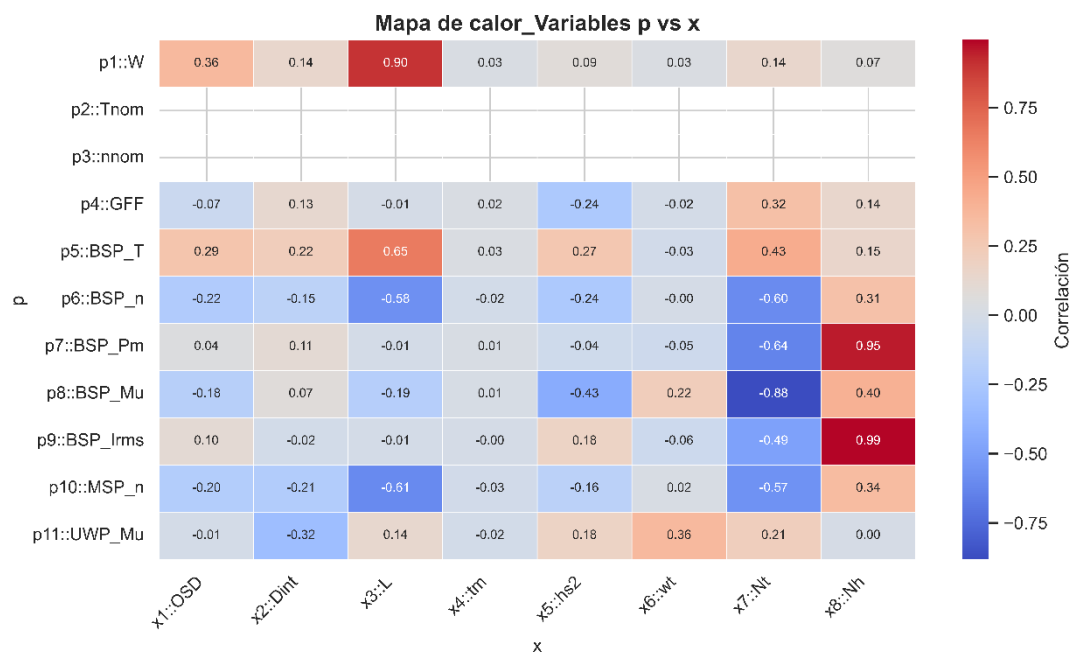


Figura 16: Matriz de correlación de Pearson entre las variables geométricas y las variables de salida. Hay huecos porque Tnom y Nnom eran valores fijos y se eliminaron para no ocasionar errores de código. Elaboración propia.

Lo interesante de esta correlación es que supo identificar relaciones lineales entre variables geométricas y variables de salida que están linealmente relacionadas en sus ecuaciones físicas como, por ejemplo:

$$T \propto (OSD)^2 \cdot L$$

Y vemos que hay una correlación fuerte entre “p5::BSP_T” y “x3::L”. Siendo cada uno de ellos: “p5::BSP_T” el par nominal y la longitud total del motor.

4. Modelo de Inteligencia Artificial Aplicado

El objetivo de este apartado es presentar en detalle la construcción, validación y optimización de los modelos de inteligencia artificial (IA) empleados como metamodelos para aproximar el comportamiento de los motores eléctricos a partir de datos generados por simulación FEA. Estos metamodelos permiten evaluar de forma rápida y precisa el rendimiento del motor en etapas de prediseño, reduciendo drásticamente el coste computacional asociado al análisis mediante elementos finitos. La metodología integra dos fases principales:

1. **Desarrollo de modelos sustitutos (surrogate models) y modelos predictivos.**
2. **Optimización de los hiperparámetros de cada modelo.**
3. **Selección del mejor candidato mediante algoritmos metaheurísticos.**

En el contexto de un flujo de diseño **data-driven**, los metamodelos actúan como funciones objetivo de bajo coste computacional que emulan la relación no lineal

$$\bar{P} = F(\bar{X}, \bar{M})$$

Entre:

- \bar{X} : Variables geométricas.
- \bar{M} : Variables conducidas (conjunto de ecuaciones dependientes).
- \bar{P} : Indicadores de rendimiento o variables de salida.

El proceso seguido para la construcción del modelo fue el siguiente:

1. **Preprocesamiento:** escalado de cada variable de entrada con StandardScaler para homogeneizar rangos y facilitar la convergencia de los algoritmos.
2. **Entrenamiento:** ajuste de distintos algoritmos de regresión supervisada sobre el conjunto de datos FEA, comparando desempeño mediante validación cruzada de 5 pliegues.
3. **Selección de metamodelos:** elección del modelo óptimo para cada salida en función del coeficiente de prognosis (*COP*) y/o del coeficiente de determinación (R^2).

Esta aproximación permite evaluar **miles de configuraciones** en segundos, frente a los minutos o horas que requeriría cada simulación FEA real.

4.1. Modelos subrogados y modelos predictores

Modelos subrogados

Los modelos subrogados (o surrogate models) son aproximaciones de bajo coste computacional que emulan el comportamiento de simuladores de alta fidelidad, como las simulaciones de elementos finitos (FEA), mediante funciones matemáticas o algoritmos de aprendizaje automático. Estos metamodelos se construyen a partir de un conjunto representativo de datos obtenidos de las simulaciones originales y permiten predecir indicadores de rendimiento (potencia, par, eficiencia, peso, etc.) con gran rapidez, sacrificando una mínima fracción de la precisión para acelerar procesos de exploración y optimización del espacio de diseño.

Modelos predictivos

Los modelos predictivos son algoritmos de inteligencia artificial entrenados para inferir la relación entre variables de entrada (por ejemplo, dimensiones geométricas y parámetros electromagnéticos) y salidas de interés (como métricas de rendimiento) a partir de datos históricos o simulados. A diferencia de los metamodelos subrogados, su objetivo principal es maximizar la precisión de predicción en nuevos casos, empleando técnicas que van desde la regresión lineal y los ensamblados de árboles hasta redes neuronales profundas, mediante procedimientos de validación cruzada y optimización de hiperparámetros.

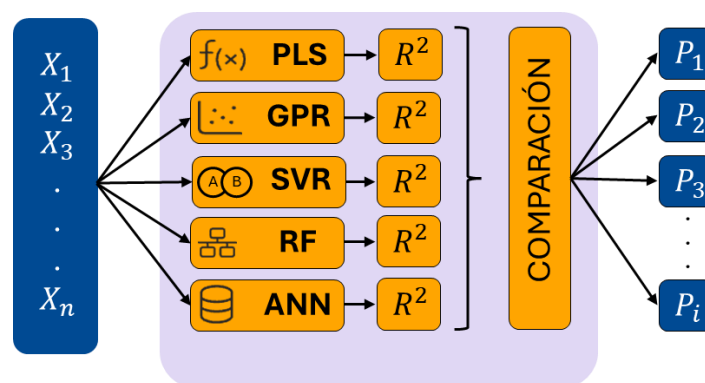


Figura 17: Esquema de operación del modelo de Inteligencia Artificial. Obtiene una matriz de datos de entrada, mediante distintas técnicas de predicción maximiza el coeficiente de determinación y selecciona la técnica que tenga ese valor más alto para cada variable de salida. Elaboración propia, inspirado en (Wang et al., 2024) [REF_ 4](#).

El flujo de construcción de estos metamodelos consta de las siguientes etapas:

1. Preprocesamiento de datos

El proceso se inicia con la carga del archivo CSV que contiene las muestras generadas. Mediante la librería pandas se identifican cada una de las matrices de diseño. Para homogeneizar los rangos y mejorar la convergencia de los algoritmos, la variable objetivo multisalida (y_{train}) se escala con StandardScaler, ajustando la media a cero y la desviación estándar a uno; el mismo escalador se aplica al conjunto de test (y_{test}) sin recalcular parámetros.

2. Definición de pipelines

Con el fin de asegurar un flujo de trabajo reproducible y modular, se define un pipeline de preprocesamiento en scikit-learn compuesto únicamente por un StandardScaler para las variables de entrada \bar{X} . Este bloque actúa como paso previo a todos los modelos y se reutiliza dentro de cada pipeline específico de regresión.

Sobre este pipeline de datos se construyen siete pipelines de modelado, uno para cada técnica evaluada:

- PLSRegression.
- LinearRegression.
- GaussianProcessRegressor (Kriging).
- Support Vector Regression (SVR).
- RandomForestRegressor..
- ANN.
- ANN-Keras.

3. Validación cruzada y métrica de selección

Antes de abordar la búsqueda fina de hiperparámetros, se realiza una evaluación preliminar de los siete modelos mediante validación cruzada de 5 pliegues. Para cada pipeline, se emplea la validación cruzada sobre el conjunto de entrenamiento escalado, obteniendo predicciones en cada partición y preservando la independencia de los datos de validación.

A partir de las predicciones se calculan:

- **MSE** (Mean Squared Error) por cada columna de salida.
- R^2 por columna de salida.
- **COP** coeficiente de prognosis.

4. Entrenamiento de hiperparámetros

Para comparar la capacidad predictiva de los modelos, se adopta como **métrica principal el coeficiente de determinación R^2** , equivalente al coeficiente de prognosis (CoP) obtenidos en validación cruzada y calculado como el cuadrado del coeficiente de correlación de Pearson entre valores reales y predichos. Este indicador es especialmente adecuado en flujos de diseño preliminar, ya que pondera tanto precisión global como fidelidad en la curva de respuesta.

Con base en este protocolo de evaluación, se procede en la siguiente sección a la optimización de hiperparámetros mediante GridSearchCV y BayesSearchCV, fijando los rangos de búsqueda específicos para cada técnica de manera que se maximice el R^2 sin comprometer excesivamente el tiempo de cómputo.

5. Selección del mejor metamodelo

Los datos de entrada alimentan los distintos modelos/técnicas de Machine Learning, estas previamente optimizadas, se entrenan para maximizar el coeficiente de determinación para cada variable de salida, se compara el mejor metamodelo y finalmente se selecciona para cada caso, generando así la matriz de variables de salida predicha.

4.2. Algoritmos regresores y predictivos usados

En la implementación anterior, se construyó una batería de siete modelos de regresión supervisada, abarcando desde métodos lineales clásicos hasta técnicas no lineales avanzadas y redes neuronales, con el fin de comparar capacidades predictivas y seleccionar el metamodelo más adecuado para cada variable de salida. A continuación, se describen de forma sistemática cada uno de los algoritmos, su configuración de hiperparámetros y su desempeño inicial en validación cruzada.

4.2.1. Regresión por Mínimos Cuadrados Parciales (PLS)

Definición

La regresión por mínimos cuadrados parciales (PLS) es una técnica que combina la reducción de dimensionalidad de análisis de componentes principales con la regresión múltiple, proyectando simultáneamente las variables de entrada X y las salidas P en un espacio latente que maximiza su covarianza. Esta característica la hace especialmente útil cuando existe multicolinealidad entre las variables de diseño y se desea construir un modelo predictivo robusto frente a datos correlacionados.

Librería

Se empleó la implementación de scikit-learn, importando `PLSRegression`.

Hiperparámetros y valores seleccionados

La optimización de hiperparámetros se realizó con `GridSearchCV` sobre los parámetros del estimador `model__n_components`, `model__max_iter`, `model__scale` y `model__tol`. Como mejores parámetros se obtuvo:

- `n_components`: 7
- `max_iter`: 500
- `scale`: False
- `tol`: 10^{-6}

Estos valores condujeron a un R^2 medio en validación cruzada de **0.8227** tras **4.92 s** de cómputo.

4.2.2. Regresión Lineal Múltiple (LR)

Definición

La regresión lineal múltiple es un método estadístico que modela la relación entre un conjunto de variables independientes y una variable dependiente asumiendo una combinación lineal de los primeros para aproximar el segundo. En este contexto, se utiliza como referencia (“baseline”) para evaluar la mejora que aportan técnicas más complejas: su simplicidad y bajo coste computacional permiten establecer un punto de comparación claro sobre el que medir la efectividad de métodos no lineales y ensamblados.

Librería

Se empleó la implementación de scikit-learn, importando LinearRegression.

Hiperparámetros y valores seleccionados

La optimización de hiperparámetros se realizó con GridSearchCV sobre los parámetros del estimador fit_intercept, positive, y copy_X. Como mejores parámetros se obtuvo:

- fit_intercept=True (se ajusta el término independiente)
- positive=False (permitir coeficientes negativos)
- copy_X=True (hacer copia interna de la matriz de diseño)

Con esta configuración, la regresión lineal múltiple alcanzó un R^2 medio de **0.8232** en validación cruzada, entrenándose en apenas **0.18 s**

4.2.3. Kriging (GPR)

Definición

La regresión por procesos gaussianos, comúnmente conocida como Kriging en el ámbito de metamodelado, se basa en la suposición de que los datos se generan según un proceso gaussiano definido por una función de media y una función de covarianza (kernel). Este modelo es capaz de capturar relaciones complejas y no lineales, proveer una estimación de la incertidumbre de la predicción y adaptarse a la forma de los datos sin requerir una especificación previa de la función de base.

Librería

Se empleó la implementación de scikit-learn, importando GaussianProcessRegressor.

Hiperparámetros y valores seleccionados

La optimización de hiperparámetros se llevó a cabo mediante BayesSearchCV, explorando:

- estimator__estimator__kernel__k1__length_scale (longitud de escala del RBF)
- estimator__estimator__kernel__k2__noise_level (nivel de ruido del WhiteKernel)
- estimator__estimator__alpha (varianza adicional)
- estimator__estimator__normalize_y (normalización de la salida)

Los valores finales seleccionados fueron:

- kernel__k1__length_scale: 5.12
- kernel__k2__noise_level: 0.001
- alpha: 10^{-8}
- normalize_y: True

Con esta configuración, el modelo GPR alcanzó un R^2 de **0.9810**, requiriendo aproximadamente **7 120 s** de optimización en el cluster.

4.2.4. Support Vector Regression (SVR)

Definición

La Support Vector Regression (SVR) es una extensión de las máquinas de vectores de soporte (SVM) para problemas de regresión. SVR busca ajustar una función que se desvíe de los puntos de entrenamiento en no más de un margen ε , penalizando desviaciones mayores a través de un parámetro de regularización C . Gracias a sus kernels (RBF, lineal, polinómico), SVR puede capturar relaciones complejas y no lineales entre las variables de diseño y los indicadores de rendimiento, manteniendo al mismo tiempo la parsimonia del modelo y evitando el sobreajuste.

Librería

Se empleó la implementación de scikit-learn, importando SVR.

Hiperparámetros y valores seleccionados

Para la optimización de parámetros se empleó GridSearchCV sobre los siguientes parámetros:

- `model__estimator__C`: [0.1, 1, 10, 100]
- `model__estimator__epsilon`: [0.001, 0.01, 0.1, 0.5, 1.0]
- `model__estimator__kernel`: ['rbf', 'linear', 'poly']
- `model__estimator__gamma`: ['scale', 'auto']

El mejor conjunto de hiperparámetros resultó ser:

- $C = 10$
- $\epsilon = 0.001$
- $\text{kernel} = \text{'rbf'}$
- $\text{gamma} = \text{'scale'}$

Con esta configuración el SVR alcanzó un R^2 de **0.9837** en validación cruzada, tras **4 709.85 s** de optimización.

4.2.5. Random Forest (RF)

Definición

El Random Forest Regressor es un ensamblado de múltiples árboles de decisión que mejora la precisión y robustez de la regresión al promediar las predicciones de cada árbol, reduciendo la varianza sin incrementar significativamente el sesgo. Cada árbol se entrena con una muestra aleatoria del conjunto de datos (bootstrap) y considera únicamente un subconjunto aleatorio de características en cada división, lo que aumenta la diversidad y evita el sobreajuste.

Librería

Se empleó la implementación de scikit-learn, importando RandomForestRegressor.

Hiperparámetros y valores seleccionados

Para la optimización de parámetros se empleó GridSearchCV sobre los siguientes parámetros:

- model__n_estimators: [50, 100, 200, 300]
- model__max_depth: [None, 10, 20, 30]
- model__max_features: ['auto', 'sqrt', 'log2']
- model__min_samples_split: [2, 5, 10]
- model__min_samples_leaf: [1, 2, 4]

El mejor conjunto de hiperparámetros resultó ser:

- n_estimators = 300
- max_depth = None
- max_features = 'log2'
- min_samples_split = 2
- min_samples_leaf = 1

Con estos valores, el Random Forest alcanzó un R^2 de **0.8772** en validación cruzada, tras un tiempo de optimización de **312 s**.

4.2.6. Redes Neuronales Artificiales (MLP)

Definición

El perceptrón multicapa (MLP) es un tipo de red neuronal de alimentación hacia adelante compuesto por una capa de entrada, una o más capas ocultas y una capa de salida. Cada neurona aplica una función de activación (generalmente no lineal) a la combinación lineal de sus entradas, lo que permite a la red aprender relaciones complejas y no lineales entre las variables de diseño y los indicadores de rendimiento del motor, ajustando los pesos mediante retropropagación del error.

Librería

Se empleó la implementación de scikit-learn, importando MLPRegressor.

Hiperparámetros y valores seleccionados

Para la optimización de parámetros se empleó GridSearchCV sobre los siguientes parámetros:

- hidden_layer_sizes: [(100, 50), (100, 100), (150, 50)]
- activation: ['relu', 'tanh', 'logistic']
- solver: ['adam', 'lbfgs', 'sgd']
- alpha (tasa de regularización L2): [10^{-4} , 10^{-3} , 10^{-2}]
- learning_rate_init: [10^{-4} , 10^{-3}]
- max_iter: [500, 1000]

El mejor conjunto de hiperparámetros resultó ser:

- hidden_layer_sizes = (100, 100)
- activation = 'logistic'
- solver = 'lbfgs'
- alpha = 10^{-4}
- learning_rate_init = 10^{-4}
- max_iter = 1000

Con estos valores, el MLPRegressor alcanzó un R^2 de **0.9866** en validación cruzada, tras un tiempo de optimización de **5316 s**.

4.2.7. Redes Neuronales Artificiales (ANN-Keras)

Definición

La implementación de redes neuronales con Keras permite construir arquitecturas personalizadas de múltiples capas densas (dense layers) para capturar relaciones altamente no lineales entre las variables de diseño de un PMSM y sus métricas de rendimiento. Gracias a su flexibilidad, estas redes pueden ajustar el número y tamaño de las capas ocultas, la función de activación y la estrategia de optimización, obteniendo modelos con gran capacidad de representación que, tras el entrenamiento, predicen con alta precisión los resultados de simulaciones FEA.

Librería

Se utilizó el wrapper KerasRegressor de la librería Scikeras, aplicado sobre una función ANN_K_model creada con TensorFlow Keras (tensorflow.keras.models.Sequential, tensorflow.keras.layers.Dense)

Hiperparámetros y valores seleccionados

Para la optimización de parámetros se empleó GridSearchCV sobre los siguientes parámetros:

- model__hidden_layer_size: número de neuronas en la primera capa oculta – valores probados [50, 100, 200]
- model__hidden_layer_size_2: neuronas en la segunda capa oculta – valores probados [3, 10, 20]
- model__epochs: épocas de entrenamiento – valores probados [100, 300, 500]
- model__batch_size: tamaño de *batch* – valores probados [16, 32]

El mejor conjunto de hiperparámetros resultó ser:

- hidden_layer_size = 100
- hidden_layer_size_2 = 10
- epochs = 500
- batch_size = 16

Con estos valores, el ANN-Keras alcanzó un R^2 de **0.9836** en validación cruzada, tras un tiempo de optimización de **1873 s**.

4.3. Optimización del Modelo mediante Metaheurísticas

En la fase de generación de la base de datos de motores generados mediante metamodelos, los modelos subrogados y predictivos entrenados permiten evaluar de forma extremadamente eficiente el desempeño de un gran número de configuraciones de diseño. Sin embargo, la exploración exhaustiva del espacio de diseño multidimensional resulta inviable incluso con metamodelos, debido al elevado coste que implicaría barrer todas las combinaciones posibles de variables geométricas y electromagnéticas. Para abordar este reto, se recurre a algoritmos metaheurísticos que combinan estrategias de búsqueda global y local, capaces de escapar de óptimos locales y aproximarse de forma efectiva a la(s) solución(es) óptima(s) bajo un conjunto de restricciones y objetivos predefinidos.

Debido a la falta de tiempo en este trabajo no se llegaron a implementar los modelos metaheurísticos que se presentan a continuación. El espacio de diseño de la base de datos de motores generada **usa un algoritmo de distribución homogénea**. El número de motores, y por lo tanto la densidad de la red en el espacio de diseño, es de **10000 motores**. Este número genera una red lo suficientemente densa para abarcar soluciones satisfactorias con la especificación inicial. Sin embargo, la selección de los mejores candidatos se estableció mediante filtros y una **frontera de Pareto**.

Se presentan tres familias representativas de metaheurísticas que podrán incorporarse en trabajos futuros:

1. Algoritmos Genéticos (GA)

Inspirados en la selección natural, operan sobre una población de individuos (vectores de variables de diseño) mediante operadores de cruce, mutación y selección basada en aptitud (fitness), calculada a partir de la predicción del metamodelo más preciso para la métrica de interés.

2. Optimización por Enjambre de Partículas (PSO)

Basados en el movimiento colectivo de partículas, cada una ajusta su posición en el espacio de diseño según su mejor experiencia individual y la mejor posición global, actualizando velocidad e inercia para equilibrar exploración y explotación del espacio.

3. Optimización por Colonia de Hormigas (ACO)

Modelan el comportamiento de forrajeo de las hormigas, donde cada solución deja “feromonas” que guían a la colonia hacia regiones

prometedoras del espacio de búsqueda; la evaporación y el refuerzo adaptativo de la feromona facilitan la convergencia hacia óptimos robustos.

5. Procedimiento del Diseño Inteligente

El procedimiento de diseño inteligente integra de manera secuencial y automatizada las fases de generación y limpieza de datos, su análisis exploratorio, el entrenamiento de metamodelos de alta fidelidad y la optimización basada en metaheurísticas.



Figura 18: Esquema del procedimiento seguido para el diseño inteligente de la base de datos de motores. Elaboración propia, inspirado en (Wang et al., 2024) [REF_4](#). Uso de imágenes extraídas del programa comercial Flux Motor®.

En primer lugar, se parte de una base de datos preprocesada de muestras FEA, que tras un filtrado de valores atípicos y faltantes se reduce a un conjunto limpio y equilibrado de diseños representativos. A continuación, mediante un notebook de EDA se obtienen estadísticas descriptivas, distribuciones y mapas de correlación que identifican las variables de entrada con mayor impacto en cada indicador de rendimiento, orientando la configuración de los regresores en la etapa posterior.

En la segunda etapa, el notebook de Machine Learning desarrolla y valida una batería de modelos subrogados —desde técnicas lineales como PLS y regresión múltiple, hasta métodos avanzados como Gaussian Process Regression, SVR y redes neuronales— mediante pipelines unificados que incluyen escalado, partición de datos y validación cruzada. La selección de hiperparámetros se realiza con GridSearchCV y BayesSearchCV, optimizando el coeficiente de determinación como métrica principal. Los metamodelos con mejor desempeño se ensamblan en un UnifiedDescaledRegressor capaz de predecir simultáneamente todos los indicadores de rendimiento con precisión equivalente a FEA, pero en fracciones de segundo.

Finalmente, el notebook maestro coordina la automatización total del prediseño: invoca de forma ordenada los notebooks de EDA, ML y MOP, gestiona la transferencia de artefactos (datos limpios, hiperparámetros, modelo unificado) y en desarrollos futuros se despliega algoritmos metaheurísticos (GA, PSO, ACO) que exploran de manera inteligente el espacio de diseño, manteniendo y refinando en cada iteración la frontera de Pareto de soluciones óptimas. El resultado es un diseño inteligente de base de datos enriquecida con centenares de miles de configuraciones prediseñadas y validadas, lista para alimentar fases posteriores de detalle y prototipado en el desarrollo de motores síncronos de imanes permanentes aeronáuticos.

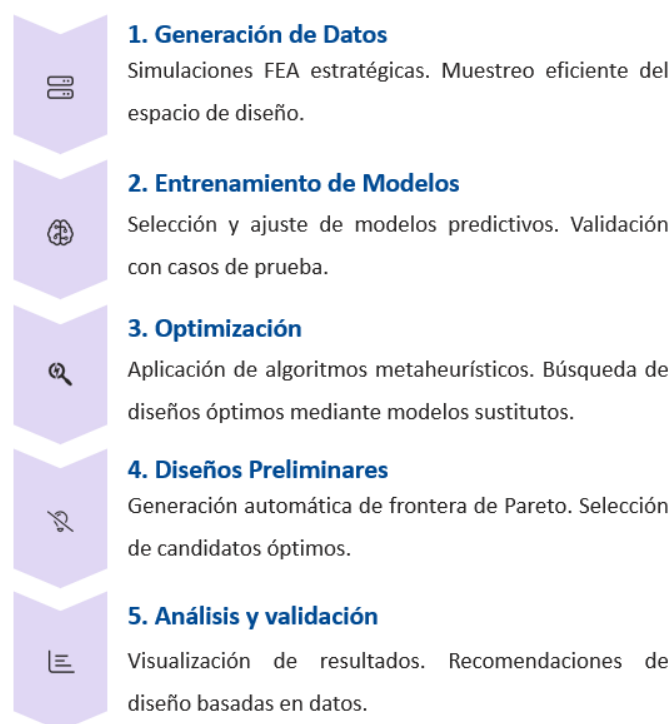


Figura 19: Resumen visual del procedimiento de diseño seguido. Elaboración propia.

5.1. Implementación del Proceso Data-Driven

El flujo completo de diseño inteligente se organiza en cuatro notebooks que procesan de manera secuencial los datos, entrenan los modelos y ensamblan el metamodelo unificado. Cada notebook automatiza una etapa clave del método Data-Driven, desde la limpieza y visualización inicial de datos (EDA), pasando por la construcción y validación de metamodelos (ML), hasta la configuración final del sistema y su orquestación en un único artefacto listo para optimización. A continuación, se describen en detalle los subapartados correspondientes a cada notebook.

5.1.1. Notebook 1: Análisis Exploratorio de Datos

El notebook EDA_V1.ipynb tiene como propósito garantizar la integridad y la representatividad del dataset generado por simulaciones FEA. Su objetivo es identificar y corregir anomalías (valores faltantes, duplicados o fuera de rango), describir estadísticamente las variables de diseño y rendimiento, y revelar relaciones preliminares entre entradas y salidas que guíen la selección de variables para el modelado posterior. Este Notebook se puede consultar en el enlace del ANEXO I: Repositorio.

El proceso de diseño es:

1. **Carga y limpieza.**
2. **Estadística descriptiva.**
3. **Visualización.**
4. **Correlación de variables.**

5.1.2. Notebook 2: Desarrollo y Entrenamiento de Metamodelos

El notebook ML_V7.ipynb implementa y evalúa una batería de siete regresores supervisados que sirven como metamodelos de la función $\bar{P} = F(\bar{X}, \bar{M})$. Su meta es comparar métodos lineales y no lineales, seleccionar los más prometedores y obtener una primera estimación del parámetro de coeficiente de determinación para cada técnica. Este Notebook se puede consultar en el enlace del ANEXO I: Repositorio.

El proceso de diseño es:

5. **Partición de datos.**
6. **Preprocesamiento común.**
7. **Definición de pipelines de regresión.**
8. **Evaluación inicial.**

5.1.3. Notebook 3: Afinamiento de Hiperparámetros y Modelo Unificado

El notebook MOP_V13.ipynb se centra en incorporar los hiperparámetros optimizados —obtenidos mediante GridSearchCV y BayesSearchCV— a los pipelines definidos, y en ensamblar un UnifiedDescaledRegressor capaz de devolver simultáneamente todas las predicciones desescaladas en una única llamada. Además, persiste el modelo unificado en disco para su uso en etapas de optimización. Este Notebook se puede consultar en el enlace del ANEXO I: Repositorio.

El proceso de diseño es:

9. Carga de hiperparámetros.
10. Evaluación de los pipelines.
11. Construcción del modelo unificado.
12. Almacenamiento del modelo unificado.

5.1.4. Notebook 4: Orquestación del Flujo Completo

El notebook DBG_V5.ipynb actúa como script maestro que orquesta la ejecución secuencial de los notebooks de EDA, ML y MOP, asegurando la correcta transferencia de artefactos intermedios (datos limpios, hiperparámetros, modelo unificado) y generando finalmente del diseño inteligente de la base de datos lista para optimización multiobjetivo. Este Notebook se puede consultar en el enlace del ANEXO I: Repositorio.

El proceso de diseño es:

13. Llamada secuencial de los elementos producidos en los anteriores Notebooks.
14. Gestión de los ficheros e integración.
15. Generación de la base de datos de 10000 motores.
16. Selección del mejor candidato

6. Resultados y Análisis

En el primer apartado se describirá la calidad y estructura del dataset final obtenido tras la limpieza y el filtrado de valores atípicos y faltantes de las simulaciones FEA. Se presentarán visualizaciones univariadas (histogramas y KDE) y mapas de calor de correlación que justifican la selección de las variables más influyentes para el modelado. Este examen inicial permitirá contextualizar la representatividad del espacio de diseño y la coherencia de los límites impuestos por las especificaciones aeronáuticas.

A continuación, se evaluará el desempeño de los metamodelos entrenados: primero, mediante validación cruzada previa a la optimización de hiperparámetros para comparar la capacidad explicativa de técnicas lineales (PLS, RL) frente a métodos no lineales (GPR, SVR, RF, ANN), y luego tras el ajuste fino de parámetros con GridSearchCV y BayesSearchCV. Se mostrarán métricas como R^2 , MSE, RMSE y el Coeficiente de Prognosis (CoP) antes y después, destacando la superioridad de las redes neuronales y los procesos gaussianos en términos de fidelidad predictiva. Asimismo, se incluirá la validación final en el conjunto de test, confirmando la robustez y ausencia de sobreajuste de los modelos definitivos.

Finalmente, el apartado se centrará en la generación del diseño inteligente de la BD de motores y el análisis de la frontera de Pareto obtenido al barrer el espacio de diseño con el metamodelo unificado. Se detallará cuántos diseños fueron evaluados, los criterios de filtrado geométrico y de rendimiento aplicados. Se presentarán comparaciones entre diseños convencionales y los propuestos por el método, incluyendo mejoras en densidad de potencia y potencia nominal, y un análisis de residuales e incertidumbre (especialmente para GPR), que aportará un nivel cuantitativo de confianza en las predicciones del sistema automatizado.

Como referencia para este apartado, se tomará como ejemplo la BD de 5000 motores diseñados mediante FEA. Pero el procedimiento y análisis de resultados es muy comparable para el resto de BDs. Al final se realizará una comparativa entre las distintas BDs y se mostrarán las implicaciones de escalarlas en tamaño y número de parámetros.

6.1. Estructura del dataset final

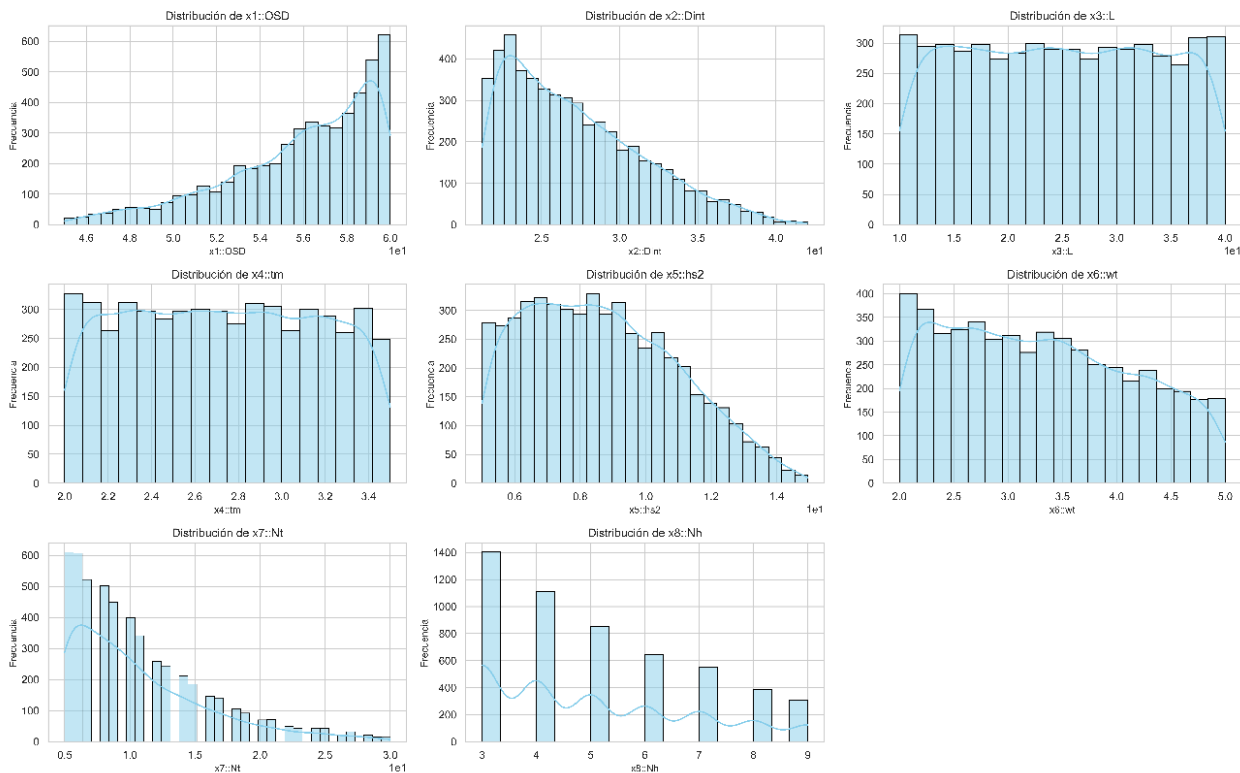
Tras la fase inicial de limpieza descrita en el apartado 3.3 Exploración y Análisis de Datos (EDA), la base de datos preprocesada contaba con 5 242 registros y 25 columnas (variables geométricas x , parámetros derivados m e indicadores de rendimiento p). A continuación, se presentan todas las columnas analizadas:

Columna	Matriz	Descripción del parámetro
x1::OSD	\bar{X}	Diámetro exterior del motor
x2::Dint		Diámetro interior del estator
x3::L		Longitud de las partes activas del motor.
x4::tm		Espesor del imán
x5::hs2		Altura de la ranura del slot
x6::wt		Ancho del diente del slot
x7::Nt		Número de vueltas
x8::Nh		Número de hilos en la mano
m1::Drot	\bar{M}	Diámetro externo del rotor
m2::Dsh		Diámetro externo del eje
m3::he		Altura del entrediente del slot
m4::Rmag		Radio del hueco del imán
m5::Rs		Altura del diente del stator
m6::GFF		Factor de llenado del conductor
p1::W	\bar{P}	Peso total del motor
p2::Tnom		Par nominal
p3::nnom		Velocidad nominal
p4::GFF		Factor de llenado de conductor (Gross Fill Facor)
p5::BSP_T		Par motor en el Base Speed Point
p6::BSP_n		Velocidad del motor en el Base Speed Point
p7::BSP_Pm		Potencia Mecánica en el Base Speed Point
p8::BSP_Mu		Rendimiento motor en el Base Speed Point
p9::BSP_Irms		Corriente rms en el Base Speed Point
p10::MSP_n		Velocidad máxima en el Maximum Speed Point
p11::UWP_Mu		Rendimiento motor en el User Working Point

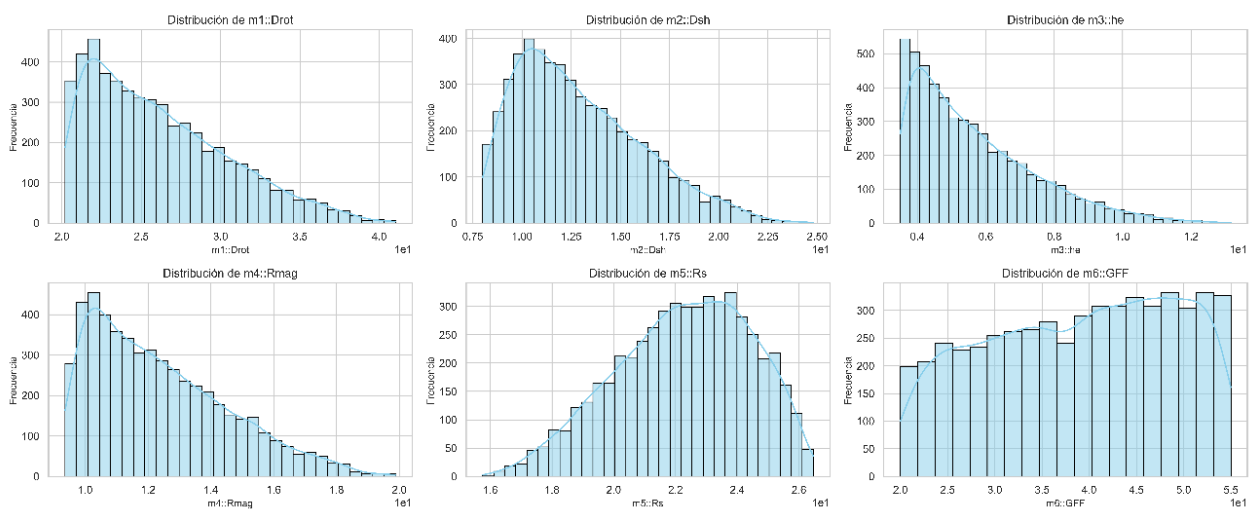
Tabla 5: Tabla descriptiva de las variables geométricas, conducidas y de salida.

La eliminación de 1481 muestras con valores faltantes en indicadores críticos y de registros no convergentes condujo a un conjunto limpio de 3 761 muestras válidas EDA_V1. Para optimizar la eficiencia de procesamiento, las columnas numéricas se convirtieron a tipos float32 e int32, reduciendo el uso de memoria en aproximadamente un 50 % sin degradar la precisión numérica. Podemos ver sus distribuciones en las siguientes figuras:

Distribuciones del grupo "x"



Distribuciones del grupo "m"



Distribuciones del grupo "p"

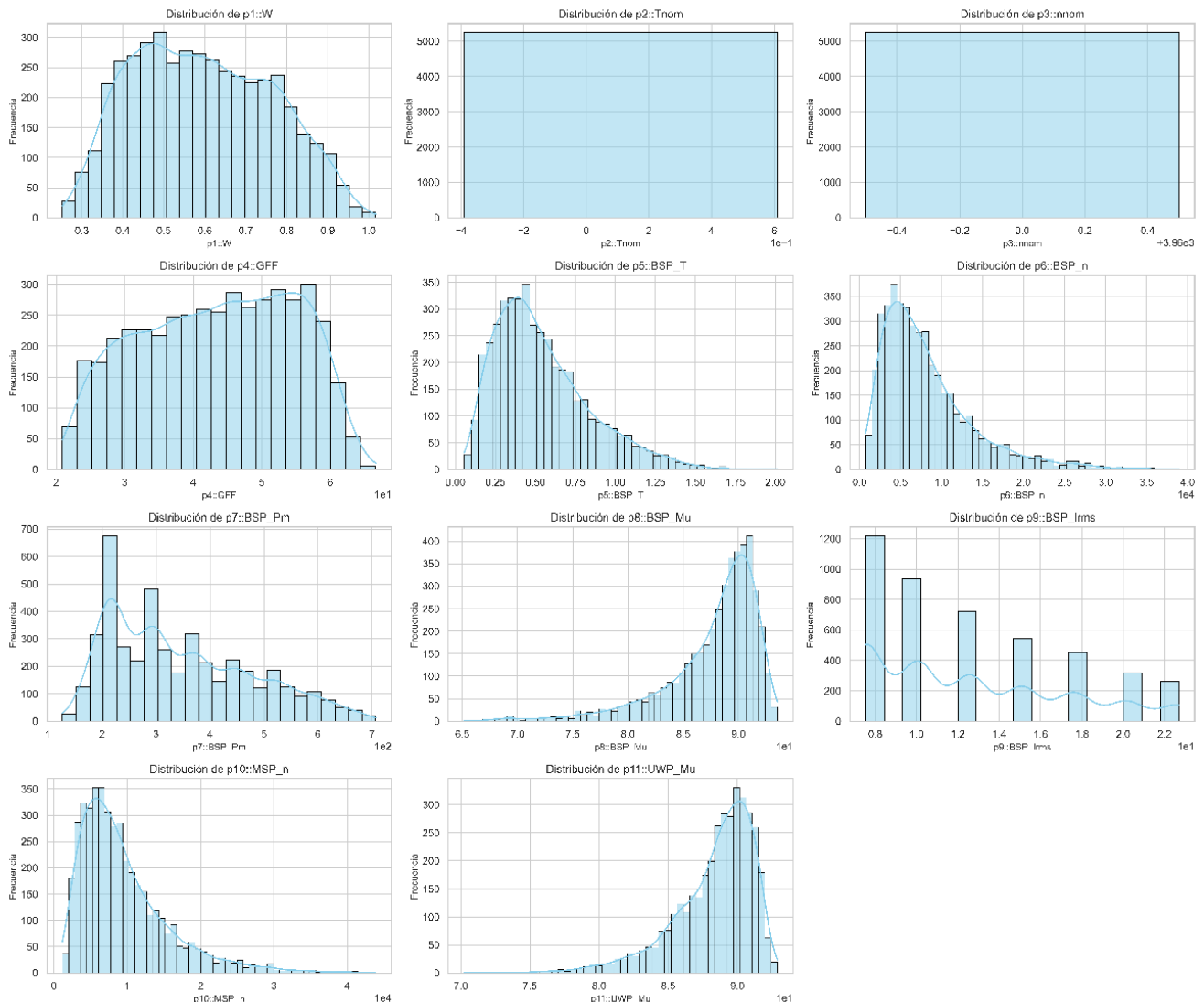


Figura 20: Distribución de cada una de las variables para la BD de 5000 motores generado por FEA. Se representa un histograma para cada variable y una línea que representa el KDE (Kernel Density Estimation) para cada una de ellas. Elaboración propia.

Tras analizar el KDE de estas variables se dedujo que había el comportamiento de algunas de las variables, como $p2::Tnom$ y $p3::nnom$, que al presentar valores constantes su impacto en los modelos era irrelevante, por lo que se omitieron mediante un filtro en el EDA.

A continuación, se analizaron las correlaciones lineales mediante correlaciones de Pearson:

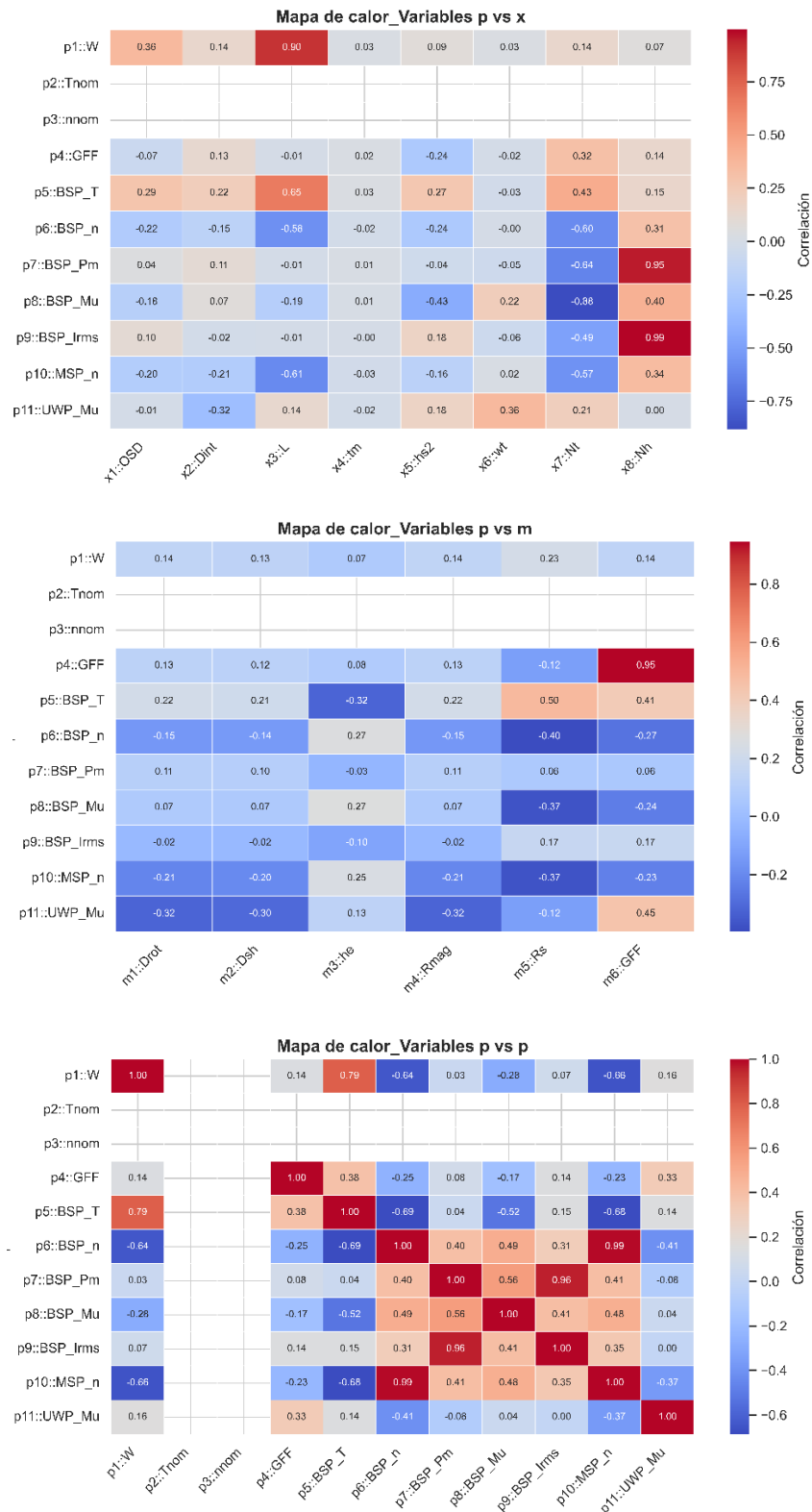


Figura 21: Mapas de calor mediante Correlación de Pearson para cada una de las variables.
Elaboración propia.

Además de poder observar dependencias lineales descritas en los modelos físicos que describen el motor eléctrico, también se pueden observar las variables que más influyen en la topología, abriendo así una vía de estudio sobre este tipo de influencia paramétrica.

6.2. Comparativa del Desempeño del Modelo

Con los datos obtenidos se entrenaron los modelos descritos, siguiendo el proceso desglosado en el apartado 4. Modelo de Inteligencia Artificial Aplicado. Que son 7: PLS, LR, GPR, SVR, RF, ANN y ANN-K.

El análisis de rendimiento del modelo se basó en maximizar el coeficiente de determinación R^2 y reducir al mínimo el error cuadrático medio MSE . Los valores resultantes de cada modelo para cada una de las variables de salida fueron los siguientes:

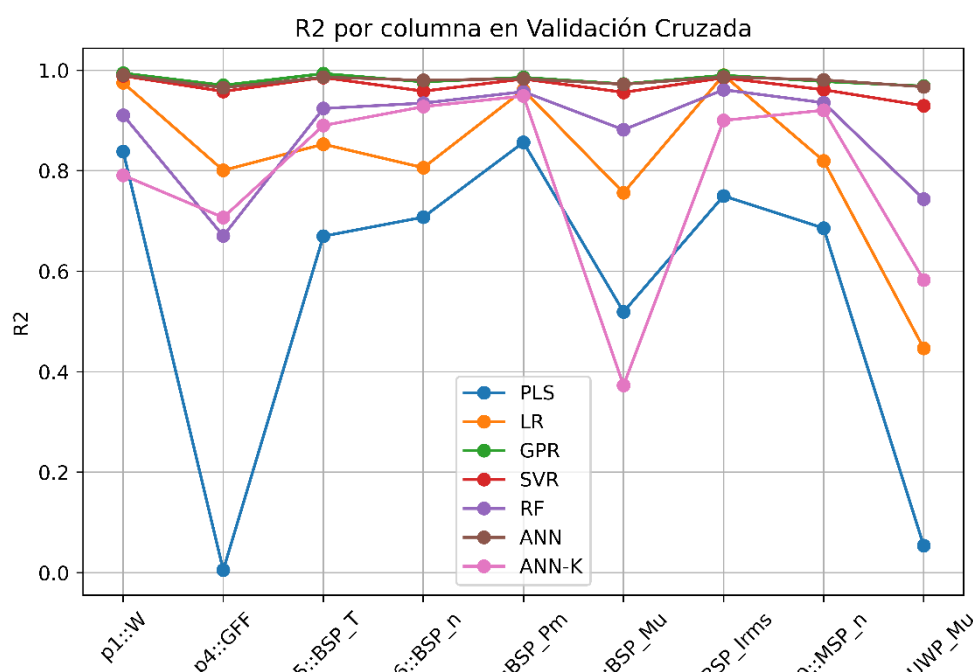


Figura 22: Valor de R^2 para cada una de las variables de salida obtenido de cada uno de los modelos entrenados con la BD de 5000 motores, antes de la optimización de hiperparámetros. Elaboración propia.

En esta figura podemos ver que los modelos lineales como PLS y LR arrojan peores resultados o no saben capturar la relación entre las variables de entrada y salida como es el caso de PLS para la variable de salida p4::GFF o P8::BSP_Mu. Por otro lado, se observa que los modelos multicapa como ANN, el modelo Krigin y SVR, capturan muy bien las relaciones no lineales de las ecuaciones que describen esta topología.

De la misma manera el error cuadrático medio:

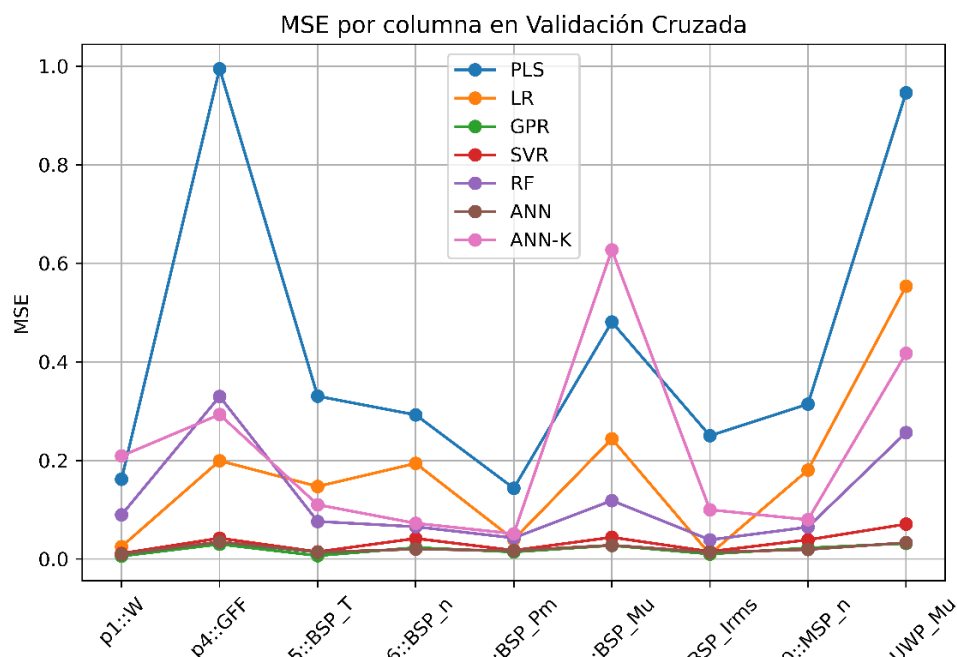


Figura 23: Valor de MSE para cada una de las variables de salida obtenido de cada uno de los modelos entrenados con la BD de 5000 motores, antes de la optimización de hiperparámetros. Elaboración propia.

La forma de la gráfica es la inversa a la anterior, como cabría esperar, en general se puede observar que los modelos con un error cuadrático medio menor son los mismos que en el caso anterior.

A modo de resumen se ha realizado una media de la puntuación de los modelos para cada una de las variables de salida y se presenta en la siguiente figura:

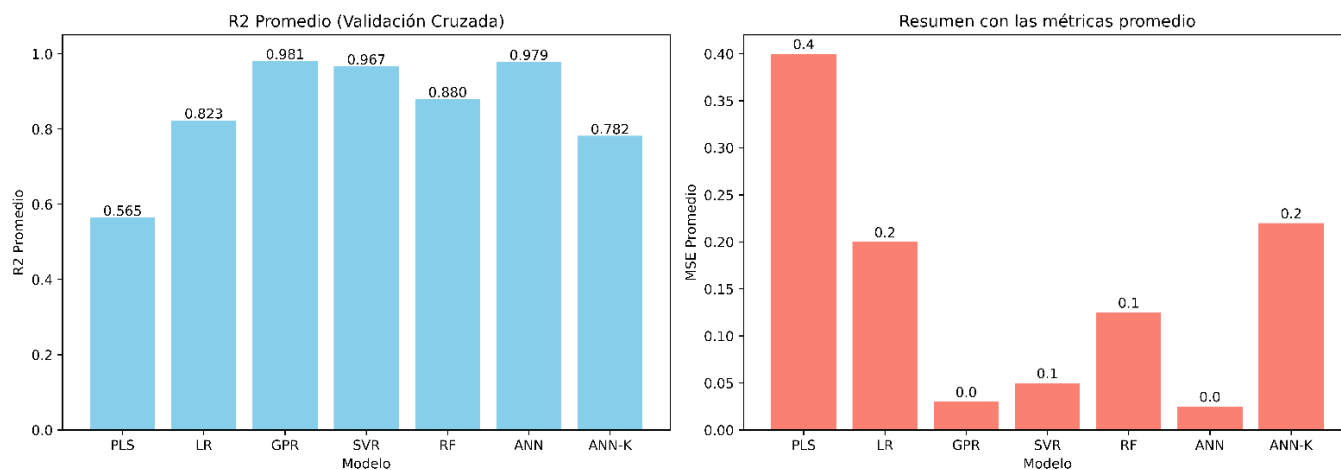


Figura 24: Resumen del valor promedio de cada uno de los modelos para el parámetro de rendimiento R^2 y MSE, antes de la optimización de hiperparámetros. Elaboración propia.

El modelo que mejor resultados presenta es el modelo **Kriging** que asume una relación gaussiana en los datos de entrada, **GPR**. Con los siguientes resultados:

$$\text{Valor de } R^2: 0.981$$

$$\text{valor de MSE: 0.019}$$

A pesar, de que los resultados de algunos de los modelos son bastante aceptables, se realizó un proceso de análisis y optimización de hiperparámetros con el fin de afinar estos modelos ganadores y observar cómo cambiaban aquellos modelos que no presentaban buenos resultados como PLS y RF.

Los hiperparámetros optimizados están descritos en el apartado 4.2. Algoritmos regresores y predictivos usados. Los resultados con el coeficiente de determinación fueron:

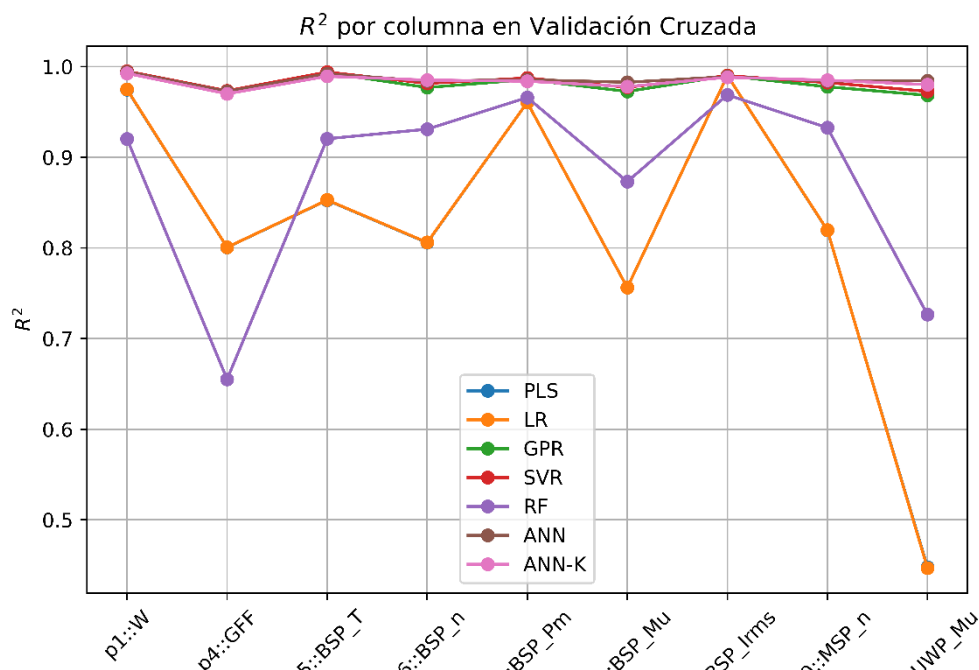


Figura 25: Valor de R^2 para cada una de las variables de salida obtenido de cada uno de los modelos entrenados con la BD de 5000 motores, después de la optimización de hiperparámetros. Elaboración propia.

Tras aplicar el entrenamiento a los modelos con los hiperparámetros óptimos se pudo observar lo siguiente:

- PLS tiende a LR con un número de componentes elevado.
- Los valores óptimos para RF y GPR, eran los valores por defecto del modelo y no sufrieron ninguna mejora.

- Los modelos multicapa fueron los más beneficiados con el ajuste de hiperparámetros ya que son los que mejores resultados presentan en la mayoría de variables de salida.
- Que el modelo general use el mejor modelo para variable de salida parece tomar sentido ya que vemos que no todos los modelos capturan igual de bien las relaciones físicas de la topología para cada variable de salida. Vemos que en este caso los mejores modelos son SVR y ANN-K.

El error cuadrático medio:

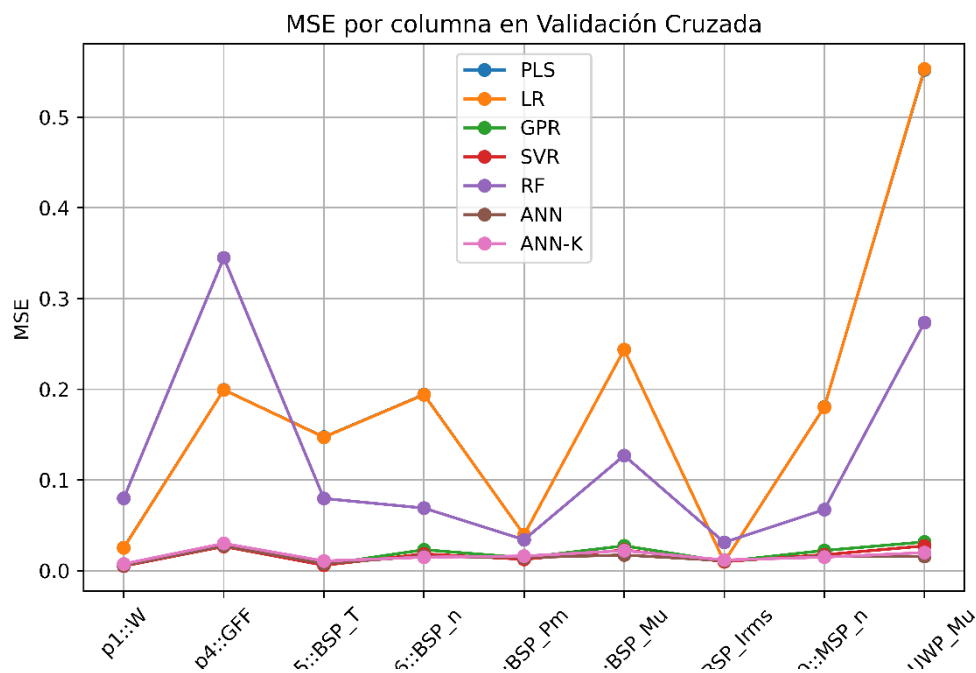


Figura 26: Valor de MSE para cada una de las variables de salida obtenido de cada uno de los modelos entrenados con la BD de 5000 motores, antes de la optimización de hiperparámetros. Elaboración propia.

Se observa una reducción general de los errores al aplicar la optimización de hiperparámetros, a excepción de los casos mencionados anteriormente.

El resumen de los valores promedio después de aplicar los hiperparámetros optimizados es:

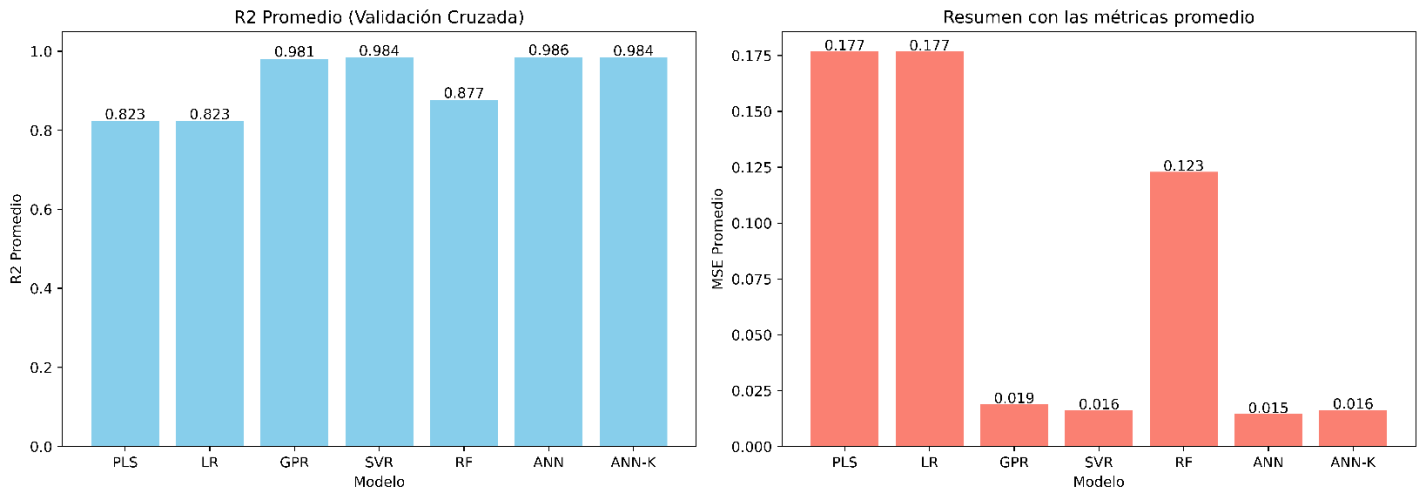


Figura 27: Resumen del valor promedio de cada uno de los modelos para el parámetro de rendimiento R^2 y MSE, después de la optimización de hiperparámetros. Elaboración propia.

Después de la optimización de hiperparámetros el modelo que mejor resultados presenta es el modelo perceptrón multicapa, ANN. Con los siguientes resultados:

Valor de R^2 : 0.986
valor de MSE: 0.015

6.3. Análisis de Sensibilidad y Correlaciones

Tras haber afinado los modelos, se desarrolló el algoritmo comparativo, encargado de seleccionar al mejor modelo para cada variable de salida. Es una comparación simple en base al coeficiente de determinación R^2 . Los resultados de esa comparativa fueron:

variable p1 :: $W \rightarrow SVR \rightarrow R^2 = 0.995$
 variable p4 :: $GFF \rightarrow SVR \rightarrow R^2 = 0.973$
 variable p5 :: $BSP_T \rightarrow SVR \rightarrow R^2 = 0.994$
 variable p6 :: $BSP_n \rightarrow ANN_K \rightarrow R^2 = 0.985$
 variable p7 :: $BSP_Pm \rightarrow SVR \rightarrow R^2 = 0.988$
 variable p8 :: $BSP_Mu \rightarrow ANN \rightarrow R^2 = 0.983$
 variable p9 :: $BSP_Irms \rightarrow LR \rightarrow R^2 = 0.990$
 variable p10 :: $MSP_n \rightarrow ANN_K \rightarrow R^2 = 0.985$
 variable p11 :: $UWP_Mu \rightarrow ANN \rightarrow R^2 = 0.984$

El modelo LR que, de forma general, no da buenos resultados, ha sido seleccionado por el algoritmo comparador por sus buenos resultados en

P9::BSP_Irms, reforzando la utilidad de usar distintos modelos y no un modelo general.

Este modelo híbrido se almacenó como un modelo unificado y desescalado. Es necesario desescalar el modelo porque no todas las variables se comportan adecuadamente con un escalador. Antes de eso, se analizaron otras métricas de correlación como *COP* y *RMSE*. Todas las métricas y evaluación entre los valores predichos por el modelo y los “reales” (datos de entrenamiento) se representan de forma conjunta a continuación:

Comparación Real vs Predicción (valores desescalados)

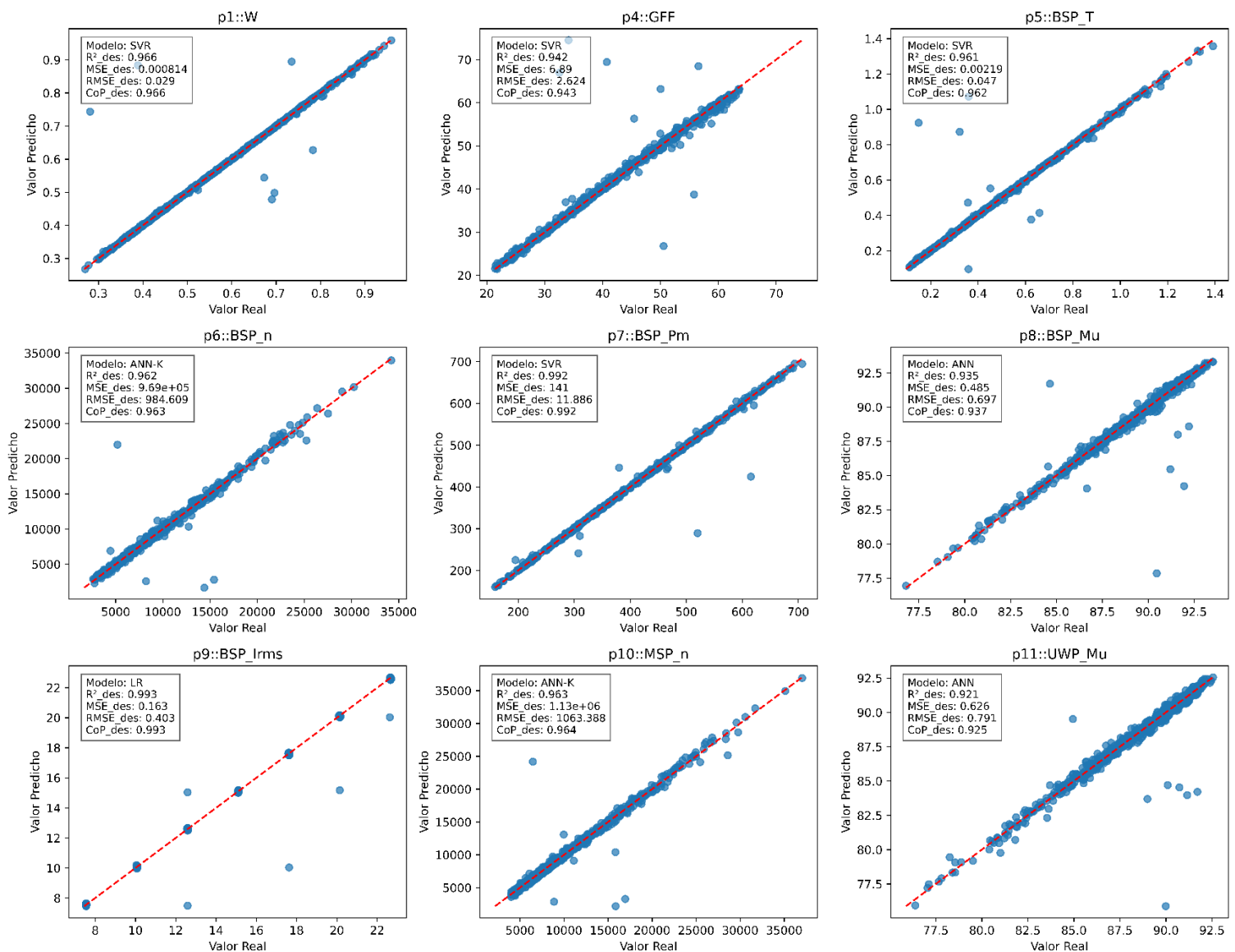


Figura 28: Correlación del modelo desescalado y análisis de las métricas R^2 , *MSE*, *RMSE*, *COP*.
Elaboración propia.

En esta figura podemos observar cómo los valores de MSE y *RMSE* no concuerdan con los valores anteriormente presentados disparándose en algunos casos como *p6::BSP_n*. Esto no es un error sino un efecto de desescalar las variables, y la influencia del escalador en el modelo.

Por otro lado, visualmente, se puede observar correlaciones muy buenas entre los resultados predichos y los reales. A pesar de que los valores del coeficiente de determinación también han variado un poco, los resultados se ajustan muy bien a la recta de regresión.

6.4. Generación de Diseño Óptimo mediante IA

Con el modelo afinado por su optimización de hiperparámetros, entrenado y preparado para elegir mediante el algoritmo comparador el mejor modelo para cada variable de salida, el siguiente paso es generar la base de datos de 10000 motores mediante un diseño inteligente.

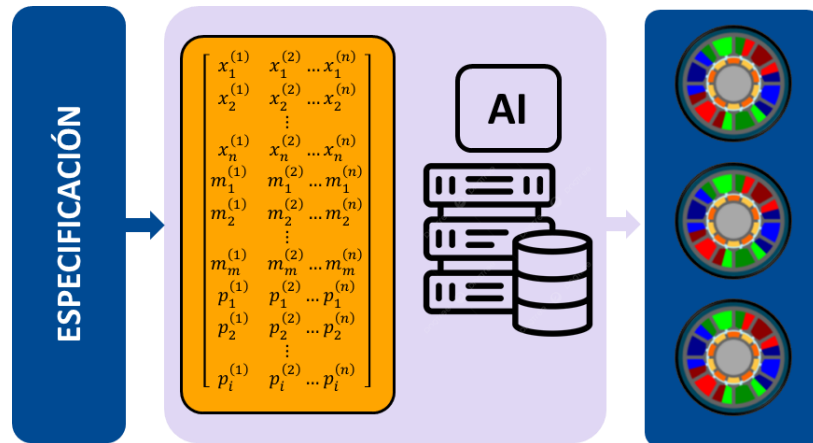


Figura 29: Esquema de alto nivel del proceso de elaboración de la base de datos de motores eléctricos mediante un diseño inteligente. Elaboración propia, inspirado en (Wang et al., 2024) [REF_ 4](#). Uso de imágenes extraídas del programa comercial Flux Motor®.

El diseño de una base de datos robusta y fabricable depende del know-how de la persona y los objetivos que se pretenden alcanzar con esa base de datos. Esto se refleja en la matriz de condiciones de contorno \bar{B} , que definirá los límites del espacio de diseño para la creación de la base de datos, así como los rangos de las variables de diseño de la matriz \bar{X} . Por ejemplo, con las condiciones de contorno, *B1::Voltaje de BUS = 28 [V]*, y la corriente máxima permitida, *B2::Intensidad maxima del sistema = 26 [Arms]* podría definirse la potencia máxima del espacio de diseño. Además de este ejemplo, diseños orientados a fabricación forman parte de una correcta elección de estos parámetros de diseño.

El algoritmo usado para la generación de la base de datos es el de un método aleatorio homogéneamente distribuido. Esto permite una exploración rápida de resultados. Sin embargo, el uso de otros métodos metaheurísticos como Algoritmos genéticos (GA), Optimización por Enjambre de Partículas (PSO) u Optimización por Colonia de Hormigas (ACO) ayudarían a generar una base de datos más optimizada a los objetivos marcados, o los requisitos de la especificación particular. Por falta de tiempo no se ha podido hacer este desarrollo, pero se plantea como desarrollo futuro.

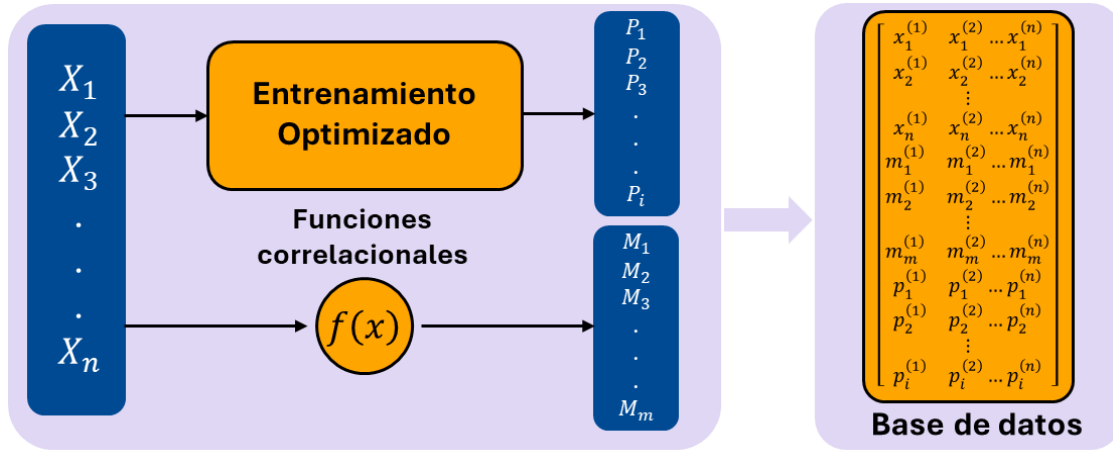


Figura 30: Proceso de generación de la base de datos de motores. Las variables de entrada se generan mediante el modelo de Inteligencia Artificial y algoritmos metaheurísticos. Elaboración propia, inspirado en (Wang et al., 2024) [REF_ 4](#).

Una vez generada la base de datos de motores es necesario hacer un filtrado de aquellos motores que cumplan con los requisitos de la especificación. Para ello se añaden filtros a las variables de salida, algunos de los filtros usados son:

- $p1::W \leq 1 [kg]$
- $p4::GFF \geq 1 \ \& \ p4::GFF \leq 60$
- $p5::BSP_T \geq 0.5 [Nm]$
- $p6::BSP_n \geq 3000 [rpm]$
- $p8::BSP_Mu \geq 85 [\%] \ \& \ p8::BSP_Mu \leq 99 [\%]$
- $p10::MSP_n \geq 4000 [rpm]$

Estos filtros ayudan a definir los puntos de operación del motor y es muy importante hacer una correcta selección de los mismos. Los motores que cumplen con estos filtros son considerados como “validos” y el resto se descartan y se consideran como “no validos”. Luego mediante una frontera de Pareto seleccionamos los parámetros de rendimiento que queremos

maximizar y/o minimizar para cumplir con la especificación, por ejemplo, en este caso:

- Minimizar: $p1::W$
- Maximizar: $p8::BSP_Mu$
- Maximizar: Densidad de potencia, obtenida como:

$$\rho_p = \frac{\text{Potencia Mecánica}}{\text{Peso total}} = \frac{p7::BSP_Pm}{p1::W}$$

Con esto la frontera de Pareto generada fue:

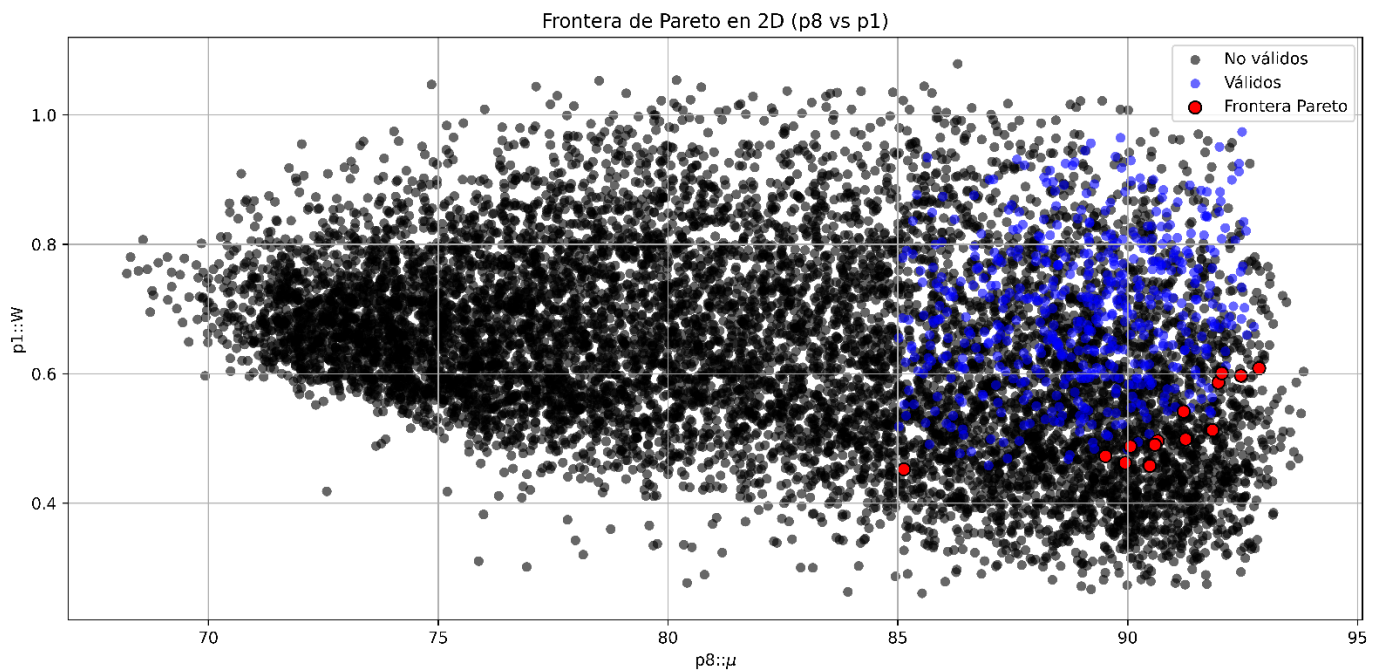


Figura 31: Frontera de Pareto. Visualización 2D del espacio de diseño generado por las variables de salida $p1::W$ y $p8::BSP_Mu$. Elaboración propia.

Esta figura representa lo anteriormente mencionado, puntos azules son los motores “válidos”, puntos negros son los “no válidos” y los puntos rojos representan los motores más óptimos dentro del espacio de diseño para la especificación inicial.

6.5. Validación del modelo y comparativa.

Entre los mejores candidatos se seleccionó aquel que cumplía con todos los requerimientos y que tenía un buen balance entre peso y rendimiento. EL candidato seleccionado fue el N° 2962 de los 10000 motores generados. Su geometría y mapa de rendimiento se muestran a continuación:

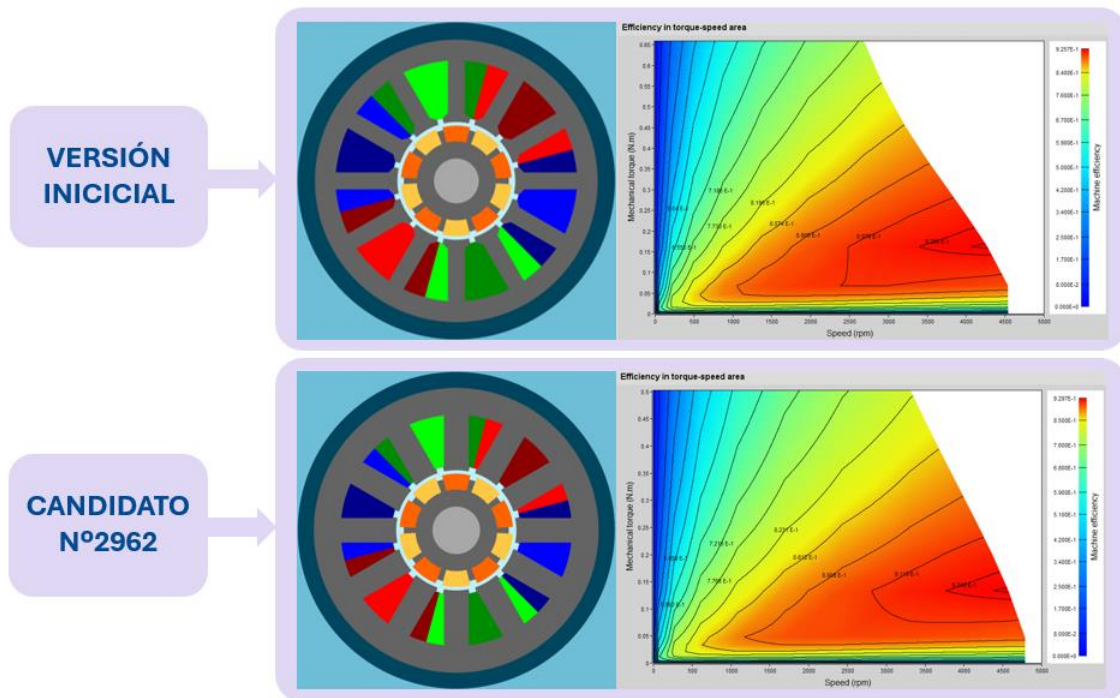


Figura 32: Representación de la geometría y del mapa de rendimiento del motor inicialmente desarrollado para esta especificación y el generado por el modelo de Inteligencia Artificial. Elaboración propia. Uso de imágenes extraídas del programa comercial Flux Motor®.

Si comparamos visualmente el diseño por un experto en el campo y el generado por el modelo de IA, vemos grandes similitudes en la toma de decisiones sobre el diseño. Es decir, vemos que el modelo opta por un tamaño de rotor pequeño además de imanes y eje bien proporcionados. También hay otras diferencias como un grueso slot mayor al de la versión inicial. A pesar de esto el motor es completamente fabricable. Cabe destacar que la curva de rendimiento presenta morfologías similares, siendo la del candidato N°2962 más suave en su decaimiento de la velocidad máxima.

Para validar al candidato se introdujeron sus parámetros de diseño en el programa comercial Flux Motor®, y se realizó un test de eficiencia como el que se mostraba en la imagen. La finalidad de esto es ver que efectivamente el motor generado es “válido” en términos de la especificación inicial y por otro lado queríamos comparar frente a los datos generados por simulación FEA las variables de salida del motor.

Los resultados se presentan en la siguiente tabla comparativa entre el motor generado por IA, el motor FEA y el motor inicialmente diseñado por un experto, que no tiene por qué tener parámetros de diseño similares, pero sí cumple con la especificación.

Parámetro	Inicial	FEA N° 2962	IA N° 2962
Diámetro exterior[mm]	51,69	8,5% ↔ 56,502	56,502
Diámetro interior [mm]	21,32	9,6% ↔ 23,589	23,589
Longitud [mm]	25,14	14,8% ↔ 21,908	21,908
Espesor iman [mm]	3,04	8,1% ↔ 3,31	3,31
Altura del slot [mm]	11,26	1,8% ↔ 11,06	11,06
Número de vueltas	20	5,3% ↔ 19	19
Número de conductores en la mano	4	33,3% ↔ 3	3
Potencia BSP [W]	52,751	37,2% ↔ 38,445	-
Par BSP [T]	0,659	31,0% ↔ 0,503	0,8% ↔ 0,499
Velocidad BSP [rpm]	2684	18,9% ↔ 3310	13,9% ↔ 3844
Densidad de potencia BSP [W/Kg]	412	9,3% ↔ 377	-
Velocidad MSP [rpm]	4538	5,2% ↔ 4783	7,72% ↔ 5183
Eficiencia nominal [%]	91,7	0,69% ↔ 92,34	0,31% ↔ 92,63
Peso total del motor [Kg]	0,566	1,91% ↔ 0,577	0,52% ↔ 0,574

Figura 33: Resumen comparativo entre el motor inicial, el motor generado por el modelo de IA y el simulado mediante FEA. Entre columnas se colocan valores de los errores relativos. Elaboración propia.

Los errores relativos que presenta el motor N°2962 en su versión generada por el modelo de IA y el FEA son muy bajos en algunas variables de salida y más elevados en otras. Sin embargo, en las variables objetivo como el peso o el rendimiento, vemos que la precisión del modelo es realmente buena, siendo menor al 1% en muchos casos.

En general, desde el punto de vista de eficiencia computacional y tiempo requerido de diseño, el diseño propuesto por el modelo de IA presenta resultados excelentes. Para una mayor precisión en las variables de salida es absolutamente recomendable hacer una simulación FEA del candidato. A pesar de eso, el tiempo requerido para generar la BD de motores y filtrar al candidato es de aproximadamente 3 minutos, la simulación FEA ha tardado

17 minutos y el diseño completo de la versión inicial realizado por un experto tardó aproximadamente 5 días. Si comparamos el tiempo y el computo requerido para el caso manual frente al caso del diseño inteligente está claro que estos nuevos procesos son muy beneficiosos en las fases iniciales de desarrollo y prediseño de motor.

6.6. Otras Bases de Datos

Como parte del estudio comparativo y para profundizar en el óptimo desarrollo del modelo de inteligencia artificial, que se ha propuesto a lo largo de este trabajo, se analizaron los resultados de varias bases de datos generadas mediante distribuciones uniformes o distribuciones optimizadas, esto se describe en el apartado 3.2. Generación de Datos mediante Simulación FEA.

El resultado fue el siguiente:

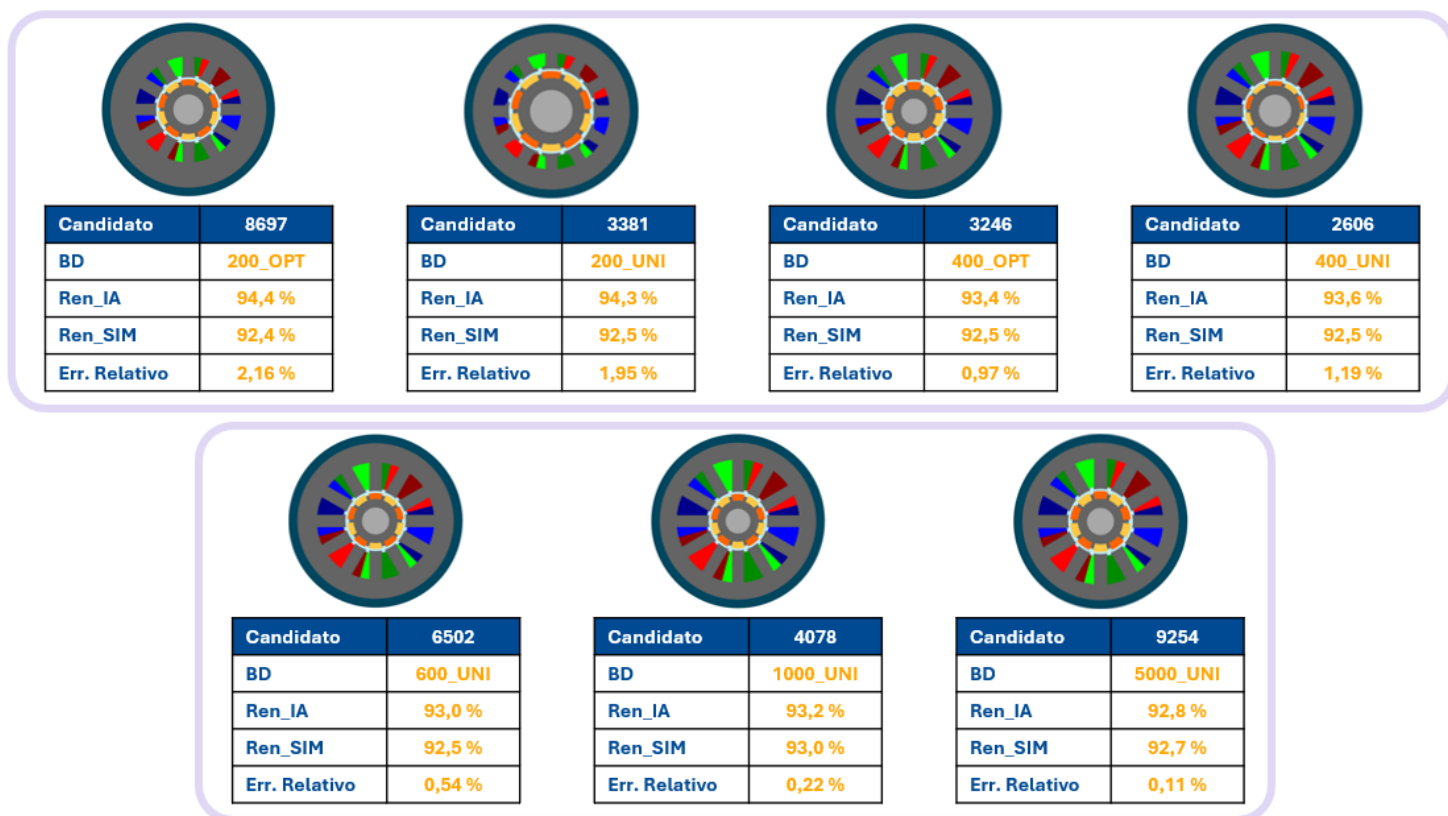


Figura 34: Candidatos para cada base de datos con el rendimiento como parámetro objetivo. Se compara el rendimiento del modelo IA y de la simulación FEA de cada candidato. Elaboración propia.

Los tiempos de cómputos son muy sensibles a la cantidad de variables de diseño que se usan (aumento del tiempo medio por simulación FEA) y al

tamaño de la base de datos objetivo (Aumento en la paralelización de las simulaciones FEA). Para hacer un balance entre tiempos de cómputo y precisión del modelo para esa cantidad de datos de entrenamiento, se analizaron todas las bases de datos mencionadas siguiendo el procedimiento descrito a lo largo de todo este trabajo.

Al comparar los errores relativos de los distintos candidatos, se puede observar que hay una disminución progresiva del error relativo a medida que aumenta el tamaño de la BD de entrenamiento, tal y como es esperable por los modelos de inteligencia artificial. Podemos considerar que un tiempo de cómputo de 5 días, para el caso de la BD de 5000 motores FEA, es demasiado, sobre todo si se quieren explorar otras geometrías u otros espacios de diseño. Por otro lado, errores relativos del 2,16% en el rendimiento no es deseable. El punto dulce lo encontramos en tamaños de BD de 600 motores, ya que tiene un tiempo de cómputo de 1 día aproximadamente y presenta errores relativos aceptables.

Al final dependerá del objetivo que se quiera alcanzar, por ejemplo, si el objetivo es hacer un barrido de topologías, usar bases de datos pequeñas puede ayudar a una exploración eficaz. Por otro lado, si se tiene muy bien definida la topología y los tamaños usuales de diseño (como sucede en aeronáutica), tener mayor precisión a costa de más tiempo de cómputo puede ser deseable.

7. Conclusiones

La investigación presentada ha desarrollado y validado un método integral de diseño inteligente para Motores Síncronos de Imán Permanente (PMSM) en aplicaciones aeronáuticas, que combina simulaciones FEA, análisis exploratorio de datos, metamodelos de IA y algoritmos metaheurísticos.

El flujo de trabajo completo, orquestado mediante notebooks especializados, ha permitido optimizar su almacenamiento y calidad, y construir metamodelos con coeficiente de determinación superior al 0,98 en las principales métricas de rendimiento, comparable a la fidelidad de FEA tradicional, pero con costes computacionales reducidos en órdenes de magnitud. El metamodelo unificado, facilita la evaluación simultánea de todos los indicadores críticos (potencia nominal, par, eficiencia, peso, densidad de potencia, etc.) en milisegundos, habilitando la exploración de decenas de miles de configuraciones en cuestión de minutos frente a los días que exigirían simulaciones convencionales.

El proceso de diseño inteligente se ha demostrado eficaz en varios niveles:

1. **Preprocesamiento y EDA:** la conversión a tipos y la eliminación de valores faltantes redujeron el tiempo de cálculo y garantizaron un dataset robusto, mientras que los mapas de correlación de Pearson guiaron la selección de variables de entrada más influyentes, mejorando la convergencia y generalización de los modelos
2. **Modelado predictivo:** la batería de siete algoritmos, desde PLS, LR, GPR, SVR, RF, ANN y ANN-K, mostró que las técnicas captaron las relaciones físicas no lineales inherentes a esta topología de motor eléctrico alcanzando $R^2 > 0,98$, tras optimización de hiperparámetros, sin sobreajuste y estabilidad en las predicciones.
3. **Diseño inteligente de la BD de 10000 motores:** aunque su implementación completa se reserva para trabajos futuros, la generación de BD inteligente con 10000 diseños uniformemente distribuidos y su filtrado mediante criterios geométricos y de rendimiento —complementados con frontera de Pareto para múltiples objetivos— demostró ser eficaz refinando la búsqueda de soluciones óptimas de forma eficiente en costes temporales y computacionales.

Con esto se consideran alcanzados todos los objetivos propuestos inicialmente, así como las hipótesis de partida que han motivado el desarrollo de esta herramienta en el prediseño de motores eléctricos.

La conclusión de este trabajo y las ideas principales se reafirman en los siguientes puntos:

- **Tiempo de diseño.**
Frente a tiempos de diseño de días o semanas por una persona experta en diseño de motores eléctricos, el diseño mediante modelos de Inteligencia Artificial puede obtener resultados comparables en cuestión de minutos.
- **Precisión respecto a simulación FEA.**
Los motores generados por esta herramienta tienen errores relativos muy bajos en comparación a las simulaciones. Esto reduce drásticamente el coste computacional y reduce el número de iteraciones necesarias para lograr obtener un motor que cumpla con la especificación objetivo.
- **Flexibilidad de diseño y escalabilidad.**
Modificando las condiciones de contorno o las variables de diseño podemos obtener distintos espacios de diseño para un mismo modelo de inteligencia artificial. Además de eso esta herramienta se puede adaptar fácilmente a otras topologías, ya que solo hay que solo depende de la base de datos de entrenamiento.

7.1. Beneficios de la IA en el Prediseño de Motores

El empleo de inteligencia artificial en el prediseño de PMSM aporta beneficios concretos y cuantificables respecto a los métodos tradicionales basados exclusivamente en simulaciones FEA y experiencia empírica. En primer lugar, la eficiencia computacional aumenta drásticamente: la evaluación de un diseño mediante el metamodelo unificado se realiza en milisegundos, permitiendo procesar cientos de configuraciones en el tiempo que demandaría una sola simulación FEA (≈ 17 min). Este ahorro de tiempo se traduce en un mayor número de iteraciones de diseño preliminar, facilitando la exploración de espacios de diseño de alta dimensión que de otro modo serían inaccesibles.

En segundo término, la exploración exhaustiva del espacio de diseño es posible gracias a la rapidez de los metamodelos. Mientras que las

simulaciones FEA limitan el número de muestras a unas pocas decenas o centenas, la metodología data-driven y metaheurística aquí desarrollada ha generado bases de datos de hasta 10 000 diseños uniformes y, potencialmente, decenas de miles utilizando algoritmos inteligentes. Esta densidad de muestreo garantiza la identificación de configuraciones no intuitivas que maximizan la densidad de potencia y la eficiencia energética dentro de restricciones geométricas y eléctricas especificadas.

Un tercer beneficio es la precisión comparable a FEA. Con R^2 promedio de 0,986 para ANN, la desviación media de las predicciones frente a simulaciones FEA reales es inferior al 1 % en variables críticas como peso y eficiencia. Esta alta fidelidad garantiza que las decisiones de diseño basadas en metamodelos sean fiables, reduciendo la necesidad de simulaciones de validación y permitiendo un ciclo de desarrollo más ágil.

Adicionalmente, la IA permite optimización multiobjetivo mediante frontera de Pareto dinámica, facilitando la toma de decisiones cuando existen objetivos contrapuestos (maximizar densidad de potencia vs. minimizar peso vs. maximizar eficiencia). El uso de metaheurísticas vinculadas a metamodelos posibilita encontrar soluciones de compromiso óptimas que satisfacen múltiples criterios simultáneamente, algo muy costoso de lograr con métodos FEA directos.

Por último, la robustez y adaptabilidad del enfoque data-driven es destacable. Al integrarse pipelines modulares y escalables (StandardScaler, PCA opcional, diversos regresores y wrappers), el sistema puede adaptarse a nuevas topologías de motor, rangos de especificación o variables adicionales (térmicas, mecánicas). Esto habilita la evolución continua del método conforme aumenten las necesidades de diseño y disponibilidad de datos, asegurando su vigencia a largo plazo.

7.2. Implicaciones para la Aeronáutica

La aplicación de estas técnicas de IA en el prediseño de motores PMSM tiene profundas implicaciones para el sector aeronáutico, donde la eficiencia, el peso y la fiabilidad son críticos. En el contexto del paradigma More Electric Aircraft, la adopción de metamodelos de IA acelera la transición hacia sistemas Power-by-Wire, reduciendo la dependencia de actuadores hidráulicos y permitiendo diseños eléctricos más ligeros y eficientes. La reducción de peso derivada de optimizar geometrías y materiales se traduce directamente en menores consumos de combustible y emisiones, contribuyendo a los objetivos de sostenibilidad de la industria.

Desde el punto de vista de certificación y mantenimiento, la capacidad de identificar de forma anticipada configuraciones robustas frente a variaciones de parámetros y condiciones de contorno fortalece el proceso de homologación ante organismos como EASA y FAA. La IA proporciona un análisis sistemático de sensibilidad e incertidumbre que cuantifica la confiabilidad de las predicciones en zonas poco muestreadas, ofreciendo evidencia cuantitativa para respaldar decisiones de diseño y planes de mantenimiento predictivo.

La reducción de costes de desarrollo es otra dimensión crítica: al acortar los ciclos de prediseño de semanas o meses a horas o días, las empresas pueden iterar más rápidamente, explorar configuraciones avanzadas y responder con mayor agilidad a cambios de requisitos de misión o de normativas ambientales. Esto mejora la competitividad de fabricantes de EMAs y motores aeronáuticos, permitiendo la personalización de diseños para plataformas específicas (drones militares, aeronaves comerciales, satélites) sin incurrir en sobrecostes desmesurados.

El potencial de extensión multidominio también es relevante: aunque este trabajo se ha centrado en variables electromagnéticas y geométricas, la metodología puede ampliarse a dominios térmico, vibracional o aerodinámico, integrando modelos de IA adicionales y datos experimentales. Esto daría lugar a simulaciones co-simuladas con metamodelos integrados que consideren interacciones complejas, ofreciendo una visión holística del comportamiento de la máquina en su entorno real de operación.

Finalmente, el éxito de este enfoque en PMSM aeronáuticos allana el camino para su aplicación a otros tipos de máquinas eléctricas (motores de inducción, máquinas de flujo axial, compresores magnéticos) y a sistemas híbridos (motor + convertidor + control), promoviendo un ecosistema de diseño automático y colaborativo basado en IA. De esta forma, la industria aeronáutica puede beneficiarse de ciclos de innovación acelerados, diseños más eficientes y herramientas de decisión fundamentadas en datos, respondiendo a los retos emergentes de electrificación, sostenibilidad y seguridad operacional.

8. Trabajos Futuros

El presente trabajo ha demostrado la eficacia de un enfoque data-driven para el prediseño de motores síncronos de imán permanente (PMSM) en aplicaciones aeronáuticas, combinando generación masiva de datos FEA, metamodelos de IA de alta fidelidad y algoritmos metaheurísticos para optimización multiobjetivo. A pesar de los excelentes resultados alcanzados, con coeficientes de determinación superiores a 0,98 y aceleraciones de dos órdenes de magnitud en tiempos de evaluación, existen múltiples líneas de investigación que podrían potenciar aún más este marco integrado. A continuación, se describen de manera exhaustiva tres grandes bloques de trabajos futuros:

8.1. Extensión a Otros Tipos de Motores

Aunque el presente estudio se ha centrado en PMSM, el método data-driven propuesto es genérico y puede adaptarse a otras topologías de máquinas eléctricas, tales como motores de inducción, motores de reluctancia conmutada (SRM), máquinas de flujo axial e incluso generadores de corriente continua. Cada una de estas familias presenta retos particulares:

- a) variabilidad en las distribuciones de flujo magnético,
- b) dependencia más acusada de parámetros de resistencia y reactancia,
- c) diferentes mecanismos de pérdidas (por ejemplo, pérdidas rotóricas en motores de inducción) que requerirían nuevos criterios de evaluación y variables de diseño.

Para cada tipología sería necesario redefinir las variables geométricas \bar{X} y las variables conducidas \bar{M} más influyentes, así como las especificaciones \bar{S} adecuadas (por ejemplo, resistencia del devanado, corrientes de rotor, ángulo de conmutación en SRM). El proceso de barrido paramétrico FEA debería ajustarse para incluir mallas y condiciones de contorno específicas — por ejemplo, modelado 3D de motores de inducción para capturar el efecto enderezador— garantizando así la representatividad de los datos generados.

La transición a otras máquinas exigirá reevaluar la selección de metamodelos y la configuración de hiperparámetros, dado que la complejidad de las relaciones $\bar{P} = F(\bar{X}, \bar{M})$ puede variar. Las técnicas de Kriging y ANN han mostrado gran versatilidad, pero podría explorarse la eficacia de métodos más específicos como XGBoost, LightGBM o Redes de Convolución 2D para

captar patrones en topologías cilíndricas o axiales, así como modelos de Transfer Learning para reutilizar conocimiento de una tipología a otra, reduciendo el coste de generación de nuevos datos.

Este trabajo futuro permitirá demostrar la generalidad del método y su aplicabilidad a la familia completa de máquinas eléctricas relevantes para la aeronáutica y otros sectores, ampliando el alcance de la AI Expert Database y su utilidad en distintas fases de desarrollo de sistemas electromecánicos.

8.2. Integración de Modelos Multidominio

La operación de un motor aeronáutico no solo depende de su comportamiento electromagnético, sino que está condicionada por fenómenos térmicos, mecánicos (vibraciones y fatiga), y aerodinámicos (flujo térmico). La metodología actual se circunscribe al dominio electromagnético, asumiendo límites térmicos (densidades de corriente) como contornos estáticos. Sin embargo, una visión holística requiere la co-simulación o integración de metamodelos en todos estos dominios, para anticipar fenómenos de interacción y garantizar la viabilidad operativa en vuelo.

Desarrollo de metamodelos multidominio:

1. **Térmico:** Entrenar un surrogate model que aproxime la distribución de temperatura del núcleo y devanados, a partir de datos FEA térmicos o CFD, para predecir hotspots y degradación de aislantes en función de las variables de diseño y las pérdidas electromagnéticas calculadas.
2. **Mecánico:** Integrar modelos de vibración modal y tensiones estructurales, entrenados a partir de simulaciones FEA mecánicas, que permitan estimar frecuencias naturales y amplitudes de vibración para cada diseño.
3. **Aerodinámico:** Generar datos CFD simplificados del flujo de aire y pérdidas por ventilación, entrenando modelos de IA que predigan la eficacia del sistema de refrigeración integrado.

Estos metamodelos se pueden encapsular en pipelines anidados, donde la predicción electromagnética alimenta al modelo térmico, que a su vez aporta datos al modelo mecánico y aerodinámico, conformando un flujo en cascada de metamodelos. La ejecución de cada etapa seguiría siendo ultrarrápida, manteniendo tiempos de evaluación de milisegundos.

La optimización multiobjetivo evoluciona hacia un problema multifidelidad y multidominio, donde los objetivos incluirían eficiencia

electromagnética, máxima temperatura, nivel de vibración y eficiencias de refrigeración. Se requerirán algoritmos metaheurísticos extensibles, por ejemplo, NSGA-III, capaces de manejar más de tres objetivos, apoyados en los metamodelos multidominio para evaluar rápidamente la aptitud de cada candidato. La frontera de Pareto resultante ofrecerá soluciones de compromiso en todos los dominios, facilitando decisiones de diseño que equilibren rendimiento, durabilidad y peso.

Desafíos y soluciones:

- **Dimensionalidad elevada:** La combinación de variables de múltiples dominios incrementa la dimensión del espacio de diseño. Se sugiere la incorporación de técnicas de Active Subspace o Autoencoders para reducir dimensionalidad priorizando combinaciones de variables con mayor impacto global.
- **Consistencia de datos:** Garantizar que los datos generados en distintos dominios correspondan a las mismas condiciones de contorno requiere un automatic pipeline de preprocesado y postprocesado, orquestado con herramientas como papermill o snakemake.
- **Verificación de fidelidad:** Cada metamodelo multidominio deberá validarse en test sets independientes y, de ser posible, con pruebas experimentales de prototipos o bancos de ensayo, cerrando el ciclo de validación.

La integración multidominio conformará un gemelo digital (digital twin) ultraligero, capaz de simular el comportamiento completo de la máquina en tiempo real y de anticipar fenómenos de fallo mecánico o térmico, alineándose con las tendencias emergentes en digitalización aeronáutica.

8.3. Desarrollo de Plataformas Automatizadas Completas

El notebook maestro DBG_V5 ya ofrece una orquestación básica de notebooks EDA, ML y MOP, pero el objetivo a mediano plazo es evolucionar este flujo hacia una plataforma de prediseño completamente automatizada, con interfaces gráficas de usuario (GUI), conectividad con sistemas CAD/CAE y capacidad de ejecución en la nube o en clusters HPC de manera transparente para el usuario final.

Componentes de la plataforma:

1. **Frontend y GUI:** Un dashboard web (por ejemplo, con React y Dash) donde el diseñador introduzca las especificaciones \bar{S} , visualice estadísticas del diseño inteligente de la BD, ajuste criterios de optimización y explore la frontera de Pareto de forma interactiva.
2. **Backend escalable:** Microservicios en Python FastAPI o Flask contendrán:
 - Gestión de datos (ingesta, limpieza y almacenamiento en bases NoSQL o SQL).
 - Módulos de entrenamiento y predicción de metamodelos (cargando joblib o contenedores Docker con TensorFlow).
 - Ejecución de metaheurísticas distribuida con Ray o Dask, habilitando la simulación de poblaciones de decenas de miles de diseños simultáneamente.
3. **Integración CAD/CAE:** Conexión directa con herramientas como Altair Flux o ANSYS Maxwell mediante APIs o scripts, para generar automáticamente modelos CAD parametrizados y, si se requiere, lanzar simulaciones de validación en batch.
4. **Módulo de digital twin:** Un servicio autónomo que monitorice en tiempo real parámetros de un prototipo físico (temperaturas, vibraciones, corrientes) y, mediante los metamodelos multidominio, ajuste la operación o realice predicciones de vida útil mediante algoritmos de RUL y Machine Learning en streaming.

El flujo completo desde la definición de especificaciones hasta la entrega de geometrías y parámetros de control debe materializarse en un proceso que:

- Genere y consolide datos de entrenamiento de forma continua (por ejemplo, incorporando resultados reales de test bench).
- Reentrene metamodelos de forma programada o basada en eventos (por ejemplo, tras acumular 1000 nuevas muestras).
- Permita el despliegue continuo (CI/CD) de la plataforma con pruebas automatizadas de validación de precisión.

- Ofrezca APIs RESTful para integrar el servicio en ecosistemas de diseño colaborativo, pipelines DevOps o incluso herramientas de Gemelo Digital a nivel de avión completo.

Beneficios y retos:

- **Reducción de barreras de entrada:** Los diseñadores podrán explotar el sistema sin conocimientos profundos de IA, simplemente definiendo especificaciones de rendimiento.
- **Escalabilidad:** La ejecución en la nube permite ajustar recursos según demanda, acelerando el prediseño en proyectos de gran envergadura.
- **Seguridad y trazabilidad:** Con registro de versiones de datos, modelos y resultados, se facilita la certificación ante organismos aeronáuticos.
- **Desafíos de validación:** Mantener la precisión de los metamodelos a lo largo de actualizaciones requiere un riguroso gobierno de datos y planes de test automáticos que detecten desviaciones de comportamiento frente a FEA o datos reales.

8.4. Ecosistema de diseño inteligente

La consecución de estas líneas de investigación: extensión a nuevas tipologías de motores, desarrollo de metamodelos multidominio y creación de plataformas de prediseño automatizado de extremo a extremo. Consolidará un **ecosistema de diseño inteligente** capaz de atender las exigencias de eficiencia, peso y fiabilidad del sector aeronáutico y más allá. La exploración de técnicas de **Transfer Learning**, **Aprendizaje Activo** y **Optimización Robusta** bajo incertidumbre, así como la implementación de **gemelos digitales** en tiempo real y la integración de **modelos co-simulados** en múltiples dominios, marcarán las futuras fronteras del prediseño automatizado de máquinas eléctricas. Con ello, se avanzará hacia un ciclo de desarrollo acelerado, iterativo y completamente fundamentado en datos, cumpliendo con los retos de la electrificación, la sostenibilidad y la seguridad operacional.

9. Bibliografía

- REF_ 1* Contreras Novoa, H. A. (2022). *Diseño y escalamiento de motor de imanes permanentes interior para aplicaciones en tracción de vehículos eléctricos*.
- REF_ 2* Qiu, W., Zhao, X., Tyrrell, A., Perinpanayagam, S., Niu, S., & Wen, G. (2024). *Application of Artificial Intelligence-Based Technique in Electric Motors: A Review*. IEEE Transactions on Power Electronics, 39(10), 13543-13568.
- REF_ 3* Sayeb, E., et al. (2021). *Review of Electric Machines in More/Hybrid/Turbo Electric Aircraft*. IEEE Transactions on Transportation Electrification, PP (99), 1-1.
- REF_ 4* Wang, Y., Yang, T., Huang, H., Zou, T., Li, J., Chen, N., & Zhang, Z. (2024). *Data Driven Automatic Electrical Machine Preliminary Design with Artificial Intelligence Expert Guidance*. (En prensa).
- REF_ 5* Imagen WEB Obtenida en:
<https://www.directindustry.es/fabricante-industrial/motor-brushless-93497.html>
- REF_ 6* Yang, S.-H., Pyo, H.-J., Jung, D.-H., & Kim, W.-H. (2023). A Study on Optimal Design Process of Dual Rotor Axial-Flux Permanent Magnet Synchronous Motors. *Machines*, 11(4), 445. DOI:<https://doi.org/10.3390/machines11040445>
- REF_ 7* Imagen WEB Obtenida en:
https://www.sohu.com/a/204712961_755644
Propietario del concepto: Audi
- REF_ 8* *Batlle i Capa, G. (2020). Study of application of electromechanical actuators to deploy control surfaces according to More Electric Aircraft concept [Bachelor's thesis, Universitat Politècnica de Catalunya]. UPC UpCommons.*
- REF_ 9* *Torabzadeh-Tari, M. (2005). Analysis of electro-mechanical actuator systems in more electric aircraft applications [Licentiate thesis, Royal Institute of Technology]*
- REF_ 10* Berri, P. C., Dalla Vedova, M. D. L., Maggiore, P., & Scanavino, M. (2018, diciembre). Permanent magnet synchronous motor (PMSM) for aerospace servomechanisms: Proposal of a lumped model for prognostics. En *Proceedings of the 2nd European Conference on Electrical Engineering & Computer Science (EECS 2018)* (pp. 471–477). IEEE. DOI:<https://doi.org/10.1109/EECS.2018.00093>
- REF_ 11* Qiao G, Liu G, Shi Z, Wang Y, Ma S, Lim TC. A review of electromechanical actuators for More/All Electric aircraft

systems. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science. 2017;232(22):4128-4151. DOI:10.1177/0954406217749869

- REF_ 12* Imagen WEB Obtenida en:
<https://cesa.aero/cleansky-2/>
- REF_ 13* Kinoti, E., Mosetlhe, T. C., & Yusuff, A. A. (2024). Multi-Criteria Analysis of Electric Vehicle Motor Technologies: A Review. *World Electric Vehicle Journal*, 15 (12), 541.
DOI:<https://doi.org/10.3390/wevj15120541>
- REF_ 14* Senevirathna, N. (2020). *Design of a Permanent Magnet Synchronous Motor for an Electric Traction Application* [Tesis de Máster, Universidad de Oviedo]. Repositorio Institucional de la Universidad de Oviedo. <http://hdl.handle.net/10651/57063>
- REF_ 15* Cheng, Mengyu, Zhao, Xing, Dhimish, Mahmoud et al. (2 more authors) (2024) *A Review of Data-driven Surrogate Models for Design Optimization of Electric Motors*. IEEE Transactions on Transportation Electrification. ISSN 2332-7782
DOI:<https://doi.org/10.1109/TTE.2024.3366417>
- REF_ 16* Vicente Cestero, E., & Mateos Caballero, A. (2023). *Inteligencia artificial: fundamentos matemáticos, algorítmicos y metodológicos* (1.^a ed.). Editorial Arquitectura Viva S.L.
- REF_ 17* Arán Godés, P. (2022). *Métodos de aprendizaje automático y aplicaciones* [Trabajo de fin de máster, Universidad de Zaragoza]. Zaguán



ANEXOS

ANEXOS

ANEXO I: Repositorio

Enlace al repositorio de GITHUB:

- https://github.com/Ergobe/TFM_GonzaloVargasTendero.git

