UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingenieros Industriales

Reconfiguración y supervisión automática de sistemas ciberfísicos basada en gemelos digitales e Inteligencia Artificial. Aplicación a una línea piloto de Industria 4.0

TESIS DOCTORAL

Presentada para optar al título de Doctor por:

Yarens Joaquín Cruz Hernández
M. Sc.

Madrid, 2024

UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingenieros Industriales

Doctorado en Automática y Robótica

Reconfiguración y supervisión automática de sistemas ciberfísicos basada en gemelos digitales e Inteligencia Artificial. Aplicación a una línea piloto de Industria 4.0

TESIS DOCTORAL

Presentada para optar al título de Doctor por:

Yarens Joaquín Cruz Hernández
M. Sc.

Bajo la dirección de:
Dr. Rodolfo Elías Haber Guerra (Director)
Dr. Fernando Castaño Romero (Codirector)

Madrid, 2024

Título: Reconfiguración y supervisión automática de sistemas ciberfísicos basada en gemelos digitales e Inteligencia Artificial. Aplicación a una línea piloto de Industria 4.0

Autor: Yarens Joaquín Cruz Hernández

Programa de Doctorado: Automática y Robótica

Dirección de tesis:

Dr. Rodolfo Elías Haber Guerra, Investigador Científico, Centro de Automática y Robótica (CSIC-UPM), (Director)

Dr. Fernando Castaño Romero, Investigador Permanente, Centro de Automática y Robótica (CSIC-UPM), (Codirector)

Revisores externos:

Tribunal de tesis:

Fecha de defensa:

Esta tesis ha sido parcialmente financiada por el proyecto europeo H2020 "*platform-enabled KITs of arTificial intelligence FOR an easy uptake by SMEs*" y por los proyectos nacionales "Auto-Reconfiguración en sistemas ciberfísicos industriales basada en gemelos digitales e IA. Métodos y aplicación en una línea piloto de Industria 4.0" y "Generador de recomendaciones inteligentes centradas en el ser humano en fabricación Cero Defectos para una reducción de residuos"

"Saber mucho no es lo mismo que ser inteligente. La inteligencia no es solo información, sino también juicio, la manera en que se recoge y maneja la información."

Carl Sagan

Agradecimientos

Esta tesis marca el fin de una etapa académica y profesional muy importante para mí, pero, sobre todo, una etapa de continuo crecimiento personal. Son muchas las personas que han contribuido directa o indirectamente a que haya podido llegar a este punto, por lo que quiero expresarles mis más sinceros agradecimientos.

Primeramente, a mis directores por haberme dado la oportunidad de formar parte de un grupo de investigación extraordinario y por haber sabido guiarme de manera certera durante todo el desarrollo de este trabajo. Por una parte, a Rodolfo Haber por confiar en mí y siempre exigirme el máximo. Su experiencia y conocimientos han sido una fuente inagotable de aprendizaje. Por otra parte, a Fernando Castaño por la dedicación y el acompañamiento en toda esta etapa. Sus explicaciones han facilitado enormemente el camino hacia la culminación de esta tesis y sus consejos han sido de gran valor en la búsqueda de soluciones.

También quiero agradecer a Albe y Arianna por todos estos años de amistad y especialmente por la acogida y el apoyo desde mi llegada a España. A Gerardo por los ánimos y el entusiasmo en este arduo proceso. A Antonio por el apoyo y la ayuda constante. Al resto de compañeros del grupo GAMHE por sus aportes y comentarios enriquecedores. Muchas gracias a todos por los debates y por los puntos de vista alternativos.

Agradezco asimismo a todo el personal del CAR, en especial a los integrantes de los distintos grupos de investigación con los que he compartido ideas y opiniones.

No quiero pasar por alto a mis antiguos compañeros del CEFAS, especialmente a Marcelino y Quiza, que siempre han estado al tanto de mi progreso y a quienes he podido pedir consejo cada vez que lo he necesitado.

Quiero agradecer al resto de mis amigos, quienes a pesar de la distancia siempre me han alentado para culminar satisfactoriamente esta etapa.

Agradezco de manera especial a todos los miembros de mi familia por siempre creer en mí, brindarme su cariño y comprensión.

Finalmente, a Liliana por ser la mejor compañera de vida. Este logro no hubiera sido posible sin ti. A Ariana, por ser la motivación e inspiración en cada proyecto que he emprendido desde que llegaste a mi vida.

Abstract

The recent convergence of a group of technological advances including the Internet of Things, data analytics, robotics and advanced simulation environments, in addition to the exponential growth experienced by Artificial Intelligence in the last few years, has led to the emergence of a new paradigm in manufacturing: Industry 4.0. This paradigm is characterized by the digitalization of production processes and revolves around the concept of cyber-physical systems, in which real-world processes are integrated with those that take place in the virtual space. This type of system is enabling the creation of more efficient, flexible and adaptive industrial environments.

However, the rapid pace at which all these transformations are occurring has made it difficult for some stakeholders to assimilate the new manufacturing paradigm due to the lack of a defined roadmap for its adoption. Although certain initiatives have defined reference architectures, most of them present the limitation of focusing only on theoretical aspects without delving into practical issues. The main objective of this thesis is the design and development of methods for automatic reconfiguration and supervision of cyber-physical systems by means of digital twins and Artificial Intelligence. The scientific-technical challenge lies in the fact that these methods must be materialized in Artificial Intelligence modules, embedded in computational platforms, ready to be applied to production processes in order to ease the digitalization of manufacturing companies.

The first module is based on an automated machine learning methodology that allows to generate optimal parametrizations of production processes through a data-driven approach. In this methodology, starting from a set of historical data that must include variables of the production process, as well as the Key Performance Indicators to be optimized, a series of stages are carried out to pre-process the data; select the most relevant variables for each Key Performance Indicator; determine between a series of algorithms with optimized hyperparameters and pre-established modelling strategies which is the most suitable one for modelling each Key Performance Indicator; and finally, use the selected models to solve a multi-objective optimization problem, obtaining as a result the optimal parametrizations of the production process. The capabilities of this module make it suitable to operate as the intelligence layer of a digital twin. This was demonstrated by integrating it into a digital twin of a pilot line. As results, models with values for the coefficient of determination higher than 0.98

were obtained and optimal Pareto sets were generated according to different constraints. Additionally, the digital twin proved to be able to co-evolve with the production process when it was deliberately exposed to different types of disturbances that changed its behavior.

The second module has the objective of maintaining the productivity of manufacturing equipment using a Fuzzy Logic strategy, specifically a fuzzy inference system. This module can be applied to heterogeneous equipment and has great relevance since it allows in a simple way to avoid fluctuations in its working speed, preventing the under-utilization or over-utilization of the equipment once it has been configured for a desired regime. The module is capable of operate within a certain range of fluctuations, whereas, if this range is exceeded, it is assumed that an abnormal situation has occurred and, therefore, the production should be stopped to avoid undesirable consequences. This module was also tested in a pilot line, demonstrating its effectiveness.

The development of this thesis has sought to contribute to the democratization of Artificial Intelligence in the industrial sector by providing different stakeholders with tools that allow building Artificial Intelligence-based solutions in a simple and affordable way.

Resumen

La reciente convergencia de un grupo de avances tecnológicos como el Internet de las Cosas, la analítica de datos, la robótica y los entornos avanzados de simulación, unido al rápido crecimiento de la Inteligencia Artificial, han propiciado el surgimiento de un nuevo paradigma de fabricación: la Industria 4.0. Este se caracteriza por la digitalización de los procesos productivos y gira en torno al concepto de sistema ciberfísico, en el cual se integran los procesos del mundo real con los del espacio virtual. Estos sistemas permiten la creación de entornos industriales más eficientes, flexibles y adaptables.

Sin embargo, el ritmo vertiginoso al que están ocurriendo estas transformaciones dificulta la asimilación del nuevo paradigma de fabricación a ciertos actores ya que no existe una ruta prestablecida para su adopción. Si bien algunas iniciativas han definido arquitecturas de referencia, estas presentan la limitación de enfocarse solamente en aspectos teóricos sin profundizar en cuestiones prácticas. Esta tesis tiene como objetivo principal el diseño y desarrollo de métodos de reconfiguración y supervisión automática de sistemas ciberfísicos mediante gemelos digitales e Inteligencia Artificial. El reto científico-técnico radica en que estos métodos deben materializarse en módulos de Inteligencia Artificial, embebidos en plataformas computacionales, listos para ser aplicados a procesos productivos con el fin de facilitar la digitalización en las empresas de fabricación.

El primer módulo se basa en una metodología de aprendizaje automático automatizado para generar parametrizaciones óptimas de los procesos productivos. A partir de datos históricos que deben incluir variables del proceso productivo y los Indicadores Clave de Desempeño que se desean optimizar, se llevan a cabo una serie de etapas para preprocesar los datos; seleccionar las variables con mayor influencia para cada Indicador Clave de Desempeño; determinar de entre una serie de algoritmos y estrategias de modelado cuál es el más adecuado para modelar cada Indicador Clave de Desempeño y; finalmente, utilizar los modelos seleccionados para plantear un problema de optimización multiobjetivo cuya solución son las parametrizaciones óptimas del proceso productivo. Las principales ventajas de este método son que, a pesar de su complejidad, su utilización es sencilla y puede ser aplicado a un amplio rango de procesos productivos. Las capacidades de este módulo le permiten actuar como capa de inteligencia de gemelos digitales, lo cual fue demostrado en una línea piloto. Como resultados se obtuvieron modelos con valores del coeficiente de

determinación superiores a 0.98 y se generaron conjuntos de Pareto óptimos para diferentes restricciones. Además, el gemelo digital demostró ser capaz de coevolucionar con el proceso productivo al ser sometido deliberadamente a varias perturbaciones que alteraban su comportamiento.

El segundo módulo tiene el objetivo de mantener la productividad de los equipos de fabricación utilizando una estrategia basada en la Lógica Borrosa, específicamente en los sistemas de inferencia borrosa. Este módulo se puede aplicar a equipos de carácter heterogéneo y tiene gran relevancia ya que permite de una manera sencilla evitar fluctuaciones en su ritmo de trabajo, previniendo la subutilización o sobreutilización de los equipos una vez que han sido configurados para un régimen de trabajo. El módulo es capaz de operar dentro de un determinado rango de fluctuaciones, mientras que, si se sobrepasa este rango, se asume que ha ocurrido una situación anómala y se detiene la producción para evitar consecuencias no deseadas. La efectividad de este módulo también fue probada en una línea piloto.

Con el desarrollo de esta tesis se busca contribuir a la democratización de la Inteligencia Artificial en el sector industrial al facilitar a los diferentes actores herramientas para construir soluciones basadas en Inteligencia Artificial de manera simple y asequible.

Tabla de Contenido

Agradecimientos v

Abstract vii

Resumen ix

Lista de Figuras xiv

Lista de Tablas xix

Abreviaturas y Acrónimos xx

1. Introducción 1

1.1. Sistemas ciberfísicos e Industria 4.0. Prolegómenos 1

1.2. Motivación y contexto 3

1.3. Objetivos de la investigación 8

1.4. Esquema de la tesis 10

2. Estado del arte. Una aproximación científico-técnica 13

2.1. Introducción 13

2.2. Sistemas ciberfísicos industriales 13

2.2.1. Arquitecturas de referencia 16

2.2.2. Limitaciones de las arquitecturas de referencia 20

2.3. Gemelos digitales en la fabricación 22

2.3.1. Niveles de diseño de gemelos digitales en la fabricación 24

2.4. Inteligencia Artificial en los procesos de fabricación 25

2.4.1. Aplicaciones de la Inteligencia Artificial en la fabricación 27

2.4.2. Aprendizaje automático en la fabricación 28

2.5. Gemelos digitales e Inteligencia Artificial en la reconfiguración y supervisión. Contexto actual 34

2.6. Conclusiones 38

3. Arquitectura 39

3.1. Introducción 39

3.2. Descripción de la arquitectura KITT4SME 39

3.3. Métodos de reconfiguración y supervisión inteligente y automática en KITT4SME. Propuesta 42

3.4. Conclusiones 44

4. Metodología para la reparametrización automática y óptima 47

4.1. Introducción 47
4.2. Metodología 47
4.2.1. Preprocesamiento de datos 51
4.2.2. Análisis sistémico y de variables 53
4.2.3. Diseño y procedimiento para el modelado automático 56
4.2.4. Criterio de selección de modelos 71
4.2.5. Optimización del proceso productivo 73
4.3. Gemelo digital 74
4.3.1. Supervisión 76
4.4. Procedimiento de implementación 80
4.5. Conclusiones 83
5. Metodología para la reconfiguración automática e inteligente **85**
5.1. Introducción 85
5.2. Método propuesto 85
5.2.1. Conjuntos borrosos de las variables de entrada y borrosificación 87
5.2.2. Conjuntos borrosos de la variable de salida 90
5.2.3. Inferencia 90
5.2.4. Desborrosificación y salidas del sistema 94
5.3. Procedimiento de implementación 96
5.4. Conclusiones 97
6. Resultados y Discusión **99**
6.1. Introducción 99
6.2. Descripción de la línea piloto 99
6.3. *Insight Generator*: Aplicación y resultados 103
6.3.1. Acondicionamiento de los datos 104
6.3.2. Selección de variables 104
6.3.3. Optimización y selección de modelos 105
6.3.4. Optimización del proceso 124
6.4. Supervisión automática del sistema ciberfísico. Resultados 129
6.5. Interfaz gráfica del gemelo digital 132
6.6. *Reco*: Aplicación y resultados 134
6.7. Conclusiones 142
7. Conclusiones **145**
7.1. Contribuciones 145
7.2. Desafíos 148
7.3. Futuras líneas de investigación y desarrollo 148

7.4. Promoción y difusión de los resultados 150

Atribuciones y Reconocimientos 153

Referencias 155

Anexos 169

Anexo A. Estructura de los ficheros de datos y configuración 170

Anexo B. Pruebas en problema ZDT4 171

Lista de Figuras

Figura 1.1: Línea de evolución de las revoluciones industriales 2
Figura 1.2: Tecnologías habilitadoras de la Industria 4.0 3
Figura 1.3: Estructura del parque empresarial en España 6
Figura 2.1: Sistemas ciberfísicos industriales 14
Figura 2.2: Modelo RAMI 4.0 17
Figura 2.3: Modelo del ecosistema *Smart Manufacturing* 18
Figura 2.4: Modelo IMSA 20
Figura 2.5: Rutas hacia la transformación digital 21
Figura 2.6: Representación básica de un gemelo digital 23
Figura 2.7: Representación de un gemelo digital de cuatro elementos 23
Figura 2.8: Evaluación del nivel de madurez en la adopción de la IA 26
Figura 2.9: Pasos típicos para desarrollar una solución de aprendizaje automático 30
Figura 2.10: Nivel 0 de MLOps (proceso manual) 31
Figura 2.11: Nivel 1 de MLOps (automatización del flujo de trabajo de aprendizaje automático) 32
Figura 2.12: Ejemplo de Nivel 2 de MLOps (automatización del flujo de trabajo de integración continua / despliegue continuo) 33
Figura 2.13: Marco de desarrollo de gemelos digitales utilizando datos de producción 35
Figura 2.14: Método de detección temprana de fallas 36
Figura 3.1: Arquitectura propuesta por KITT4SME 40
Figura 3.2: Representación conceptual del sistema propuesto 45
Figura 4.1: Metodología para la reparametrización óptima de los procesos productivos 49
Figura 4.2: Flujo de trabajo de la metodología 50
Figura 4.3: Neurona artificial 57
Figura 4.4: Perceptrón multicapa 58

Figura 4.5: Puntos que exceden el margen de error permitido 60
Figura 4.6: Aplicación de un *kernel* para transformar un problema no lineal a uno lineal 61
Figura 4.7: Predicción con XGBoost 63
Figura 4.8: Determinación de los *k*-vecinos más cercanos 64
Figura 4.9: Ejemplo simplificado de la determinación del valor más prometedor de un hiperparámetro 67
Figura 4.10: Ejemplo de mejora en la generalización al combinar las salidas de dos modelos 69
Figura 4.11: Predicción del conjunto mediante la combinación de las salidas 70
Figura 4.12: Predicción del conjunto mediante un metamodelo 72
Figura 4.13: Diagrama de bloques de NSGA-II 75
Figura 4.14: Tipos de deriva de concepto 77
Figura 4.15: Ejemplo de subdivisiones de W analizadas por ADWIN al recibir un nuevo dato para $nmin = 5$ 79
Figura 4.16: Herramientas y bibliotecas utilizadas en la implementación de las propuestas relacionadas con el método de reparametrización óptima y el gemelo digital 81
Figura 5.1: Funciones de pertenencia para los conjuntos borrosos de la variable ΔTa 88
Figura 5.2: Funciones de pertenencia para los conjuntos borrosos de la variable ΔTp 89
Figura 5.3: Ejemplo de determinación del grado de pertenencia a los diferentes conjuntos borrosos de la variable ΔTa 89
Figura 5.4: Funciones de pertenencia para los conjuntos borrosos de la variable $RECO1$ .. 90
Figura 5.5: Ejemplo del cálculo del valor de verdad del antecedente de una regla 92
Figura 5.6: Ejemplo de implicación 93
Figura 5.7: Resultados de diferentes métodos de desborrosificación 95
Figura 5.8: Herramientas y bibliotecas utilizadas en la implementación de las propuestas relacionadas con el método de rectificación de desviaciones en la producción 97
Figura 6.1: Vista superior y lateral de un panel termoaislante 100
Figura 6.2: Operaciones para la manufactura de los paneles termoaislantes en la línea piloto de GAMHE 101

Figura 6.3: Línea piloto del grupo GAMHE 102
Figura 6.4: Secuencia de actuación de los elementos de la línea piloto de GAMHE 102
Figura 6.5: Estandarización de las variables de entrada 104
Figura 6.6: Matrices de correlación para Tp 106
Figura 6.7: Matrices de correlación para Ti 107
Figura 6.8: Optimización de hiperparámetros de MLP para Tp 109
Figura 6.9: Optimización de hiperparámetros de SVR para Tp 110
Figura 6.10: Optimización de hiperparámetros de XGBoost para Tp 111
Figura 6.11: Optimización de hiperparámetros de *k*NN para Tp 112
Figura 6.12: Valores reales vs. predicciones de los diferentes modelos para Tp 115
Figura 6.13: Optimización de hiperparámetros de MLP para Ti 117
Figura 6.14: Optimización de hiperparámetros de SVR para Ti 118
Figura 6.15: Optimización de hiperparámetros de XGBoost para Ti 119
Figura 6.16: Optimización de hiperparámetros de *k*NN para Ti 120
Figura 6.17: Valores reales vs. predicciones de los diferentes modelos para Ti 123
Figura 6.18: Frontera de Pareto estimada por la metodología sin restricciones adicionales 125
Figura 6.19: Conjunto de Pareto para las variables seleccionadas sin restricciones adicionales 125
Figura 6.20: Frontera de Pareto estimada por la metodología con la restricción $ad = 27000$ mm/min 126
Figura 6.21: Conjunto de Pareto para las variables seleccionadas con la restricción $ad = 27000$ mm/min 127
Figura 6.22: Frontera de Pareto estimada por la metodología con las restricciones $vs = 7.7$ m/s y $vu = 70\%$ 128
Figura 6.23: Conjunto de Pareto para las variables seleccionadas con las restricciones $vs = 7.7$ m/s y $vu = 70\%$ 128
Figura 6.24: Detección de una deriva de concepto repentina 130
Figura 6.25: Detección de una deriva de concepto incremental 131

Figura 6.26: Detección de una deriva de concepto gradual 132

Figura 6.27: Captura de la interfaz gráfica del gemelo digital de la línea piloto creada en FlexSim 133

Figura 6.28: Comportamiento frente a una perturbación instantánea en las cintas transportadoras y una deriva de concepto repentina en el centro de mecanizado 137

Figura 6.29: Comportamiento frente a una perturbación instantánea en la fresadora CNC y una deriva de concepto incremental en el robot UR5e 138

Figura 6.30: Comportamiento frente a perturbaciones instantáneas en el robot Stäubli RX90 y la fresadora CNC KERN Evo, y una deriva de concepto gradual en el robot UR5e ... 140

Figura 6.31: Comportamiento frente a perturbaciones instantáneas en el robot Stäubli RX90 y las cintas transportadoras y derivas de concepto repentina en la fresadora CNC KERN Evo, gradual en el centro de mecanizado Deckel Maho e incremental en el robot UR5e 140

Figura B.1: Distribución de los datos antes y después de la estandarización 172

Figura B.2: Matrices de correlación para la salida $y0$ 173

Figura B.3: Matrices de correlación para la salida $y1$ 174

Figura B.4: Optimización de hiperparámetros de MLP para salida $y0$ 175

Figura B.5: Optimización de hiperparámetros de SVR para salida $y0$ 176

Figura B.6: Optimización de hiperparámetros de XGBoost para $y0$ 177

Figura B.7: Optimización de hiperparámetros de *k*NN para salida $y0$ 178

Figura B.8: Valores reales vs. predichos por los diferentes modelos para $y0$ 180

Figura B.9: Optimización de hiperparámetros de MLP para salida $y1$ 182

Figura B.10: Optimización de hiperparámetros de SVR para salida $y1$ 183

Figura B.11: Optimización de hiperparámetros de XGBoost para $y1$ 184

Figura B.12: Optimización de hiperparámetros de *k*NN para salida $y1$ 185

Figura B.13: Valores reales vs. predichos por los diferentes modelos para $y1$ 187

Figura B.14: Frontera de Pareto real para ZDT4 189

Figura B.15: Frontera de Pareto estimada por la metodología 189

Figura B.16: Comparación de la solución encontrada con el espacio realizable 190
Figura B.17: Conjuntos de Pareto real y obtenido 190

Lista de Tablas

Tabla 1.1: Desafíos para la adopción de la Industria 4.0 4

Tabla 2.1: Aplicaciones de IA en la fabricación 27

Tabla 4.1: Rangos explorados durante la optimización de hiperparámetros 67

Tabla 6.1: Rangos de trabajo experimentales definidos para las variables en la línea piloto de GAMHE..... 103

Tabla 6.2: Hiperparámetros seleccionados y resultados de validación para Tp 114

Tabla 6.3: Hiperparámetros seleccionados y resultados de validación para Ti 122

Tabla 6.4: Parametrización utilizada en las pruebas de supervisión 129

Tabla 6.5: Parametrización utilizada en las pruebas de corrección de desviaciones en la productividad 135

Tabla B.1: Hiperparámetros seleccionados y resultados de validación para la salida $y0$ ... 179

Tabla B.2: Hiperparámetros seleccionados y resultados de validación para la salida $y1$ ... 186

Abreviaturas y Acrónimos

5G	Red celular de quinta generación
ADWIN	*Adaptive Windowing* (Ventana adaptativa)
AE	*Autoencoder* (Autocodificador)
AMQP	*Advanced Message Queuing Protocol* (Protocolo Avanzado de Colas de Mensajes)
AutoML	*Automated Machine Learning* (Aprendizaje Automático Automatizado)
CAR	Centro de Automática y Robótica
CAx	*Computer-Aided Technologies* (Tecnologías Asistidas por Ordenador)
CNC	*Computer Numerical Control* (Control Numérico Computarizado)
CNN	*Convolutional Neural Network* (Red Neuronal Convolucional)
CoAP	*Constrained Application Protocol* (Protocolo de Aplicación Restringida)
CSIC	Consejo Superior de Investigaciones Científicas
CSKR	*Commonsense Knowledge-based Reasoning* (Razonamiento basado en Sentido Común)
FIS	*Fuzzy Inference System* (Sistema de Inferencia Borroso)
GA	*Genetic Algorithm* (Algoritmo Genético)
GAMHE	*Group of advanced Automation of Machines, Highly Complex Processes and Environments* (Grupo de Control Avanzado de Máquinas, Procesos y Entornos Complejos)
GPR	*Gaussian Process Regression* (Proceso Gaussiano para Regresión)
HTTP	*Hypertext Transfer Protocol* (Protocolo de Transferencia de Hipertexto)
IA	Inteligencia Artificial

iGENZERO	Generador de recomendaciones inteligentes centradas en el ser humano en fabricación Cero Defectos para una reducción de residuos
IMSA	*Intelligent Manufacturing System Architecture* (Arquitectura para Sistema de Fabricación Inteligente)
IoT	*Internet of Things* (Internet de las Cosas)
JSON	*JavaScript Object Notation* (Notación de Objetos de JavaScript)
KITT4SME	*platform-enabled KITs of arTificial intelligence FOR an easy uptake by SMEs* (Kits de inteligencia artificial disponibles en una plataforma para una adopción sencilla por parte de las Pequeñas y Medianas Empresas)
*k*NN	*k-Nearest Neighbors* (*k*-Vecinos más Cercanos)
KPI	*Key Performance Indicator* (Indicador Clave de Desempeño o Cifra de Mérito)
LASSO	*Least Absolute Shrinkage and Selection Operator* (Operador de Encogimiento y Selección Mínima Absoluta)
LSTM	*Long Short-Term Memory network* (red de Memoria a Corto y Largo Plazo)
LwM2M	*Lightweight Machine to Machine protocol* (protocolo Máquina a Máquina Ligero)
MES	*Manufacturing Execution System* (Sistema de Ejecución de Fabricación)
MLP	*Multi-layer Perceptron* (Perceptrón Multicapa)
MLOps	*Machine Learning Operations* (Operaciones de Aprendizaje Automático)
MQTT	*Message Queuing Telemetry Transport* (Transporte Telemétrico de Colas de Mensajes)
NGSI-v2	*Next Generation Service Interfaces v2* (Interfaces de Servicio de Próxima Generación v2)
NSGA-II	*Non-dominated Sorting Genetic Algorithm II* (Algoritmo Genético de Ordenación No Dominada II)

oneM2M	*one Machine to Machine* (una Máquina a Máquina)
OPC-UA	*Open Platform Communications Unified Architecture* (Arquitectura Unificada de Comunicaciones de Plataforma Abierta)
RAMI 4.0	*Reference Architectural Model for Industry 4.0* (Modelo de Arquitectura de Referencia para la Industria 4.0)
RAMP	*Robotics and Automation Market Place* (Mercado de Robótica y Automatización)
REST API	*Representational State Transfer Application Programming Interface* (Interfaz de Programación de Aplicaciones de Transferencia de Estado Representacional)
RF	*Random Forest* (Bosque Aleatorio)
RNSGA-II	*Reference Point-based Non-dominated Sorting Genetic Algorithm II* (Algoritmo Genético de Ordenación No Dominada basado en Puntos de Referencia II)
ROS2	*Robot Operating System 2* (Sistema Operativo de Robots 2)
RTPS	*Real Time Publish-Subscribe protocol* (protocolo de Publicación y Suscripción en Tiempo Real)
SVM	*Support Vector Machine* (Máquina de Vectores de Soporte)
SVR	*Support Vector Regression* (Vectores de Soporte para Regresión)
SELFRECO	Auto-Reconfiguración en sistemas ciberfísicos industriales basada en gemelos digitales e IA. Métodos y aplicación en una línea piloto de Industria 4.0
TPE	*Tree-structured Parzen Estimator* (Estimador de Parzen estructurado en forma de Árbol)
UPM	Universidad Politécnica de Madrid
XGBoost	*Extreme Gradient Boosting* (Refuerzo de Gradientes Extremo)

1. Introducción

1.1. Sistemas ciberfísicos e Industria 4.0. Prolegómenos

El término revolución industrial se refiere a un cambio significativo de paradigma en la forma de producir bienes. Una revolución industrial se caracteriza por la introducción de nuevas tecnologías que cambian el panorama industrial de su época, pero que también producen un profundo impacto en la economía, la sociedad y la cultura (Kurt, 2019). Hasta la fecha ha habido cuatro revoluciones industriales, las cuales se representan en la Figura 1.1. La primera revolución industrial ocurrió en el período comprendido entre el fin del siglo XVIII y el principio del XIX, y se caracterizó por la mecanización de la producción y la introducción de la máquina de vapor (Galati & Bigliardi, 2019). Este fue un punto de inflexión en la historia de la humanidad, cuando el modelo tecnológico industrial sustituyó al trabajo manual. Entre el fin del siglo XIX y principios del XX tuvo lugar la segunda revolución industrial, impulsada por el uso de la energía eléctrica (C. Zhang & Yang, 2020). En este período, la introducción de las líneas de ensamblaje contribuyó a la consolidación de la producción en serie. El telégrafo y el teléfono cambiaron radicalmente la forma de comunicarse y el motor de combustión interna propició la creación de nuevas formas de transporte. La tercera revolución industrial ocurrió en la segunda mitad del siglo XX y se caracterizó por la introducción de la electrónica, especialmente las computadoras, y el internet en los procesos productivos (Popkova et al., 2019). Los avances en las tecnologías de la información y la mecatrónica permitieron automatizar un gran número de industrias y escalarlas a nivel global. Finalmente, la cuarta revolución industrial, la cual aún se está desarrollando, está caracterizada por el surgimiento e integración de nuevas tecnologías que están cambiando profundamente la vida cotidiana, la producción y las relaciones de trabajo (Gajdzik et al., 2021).

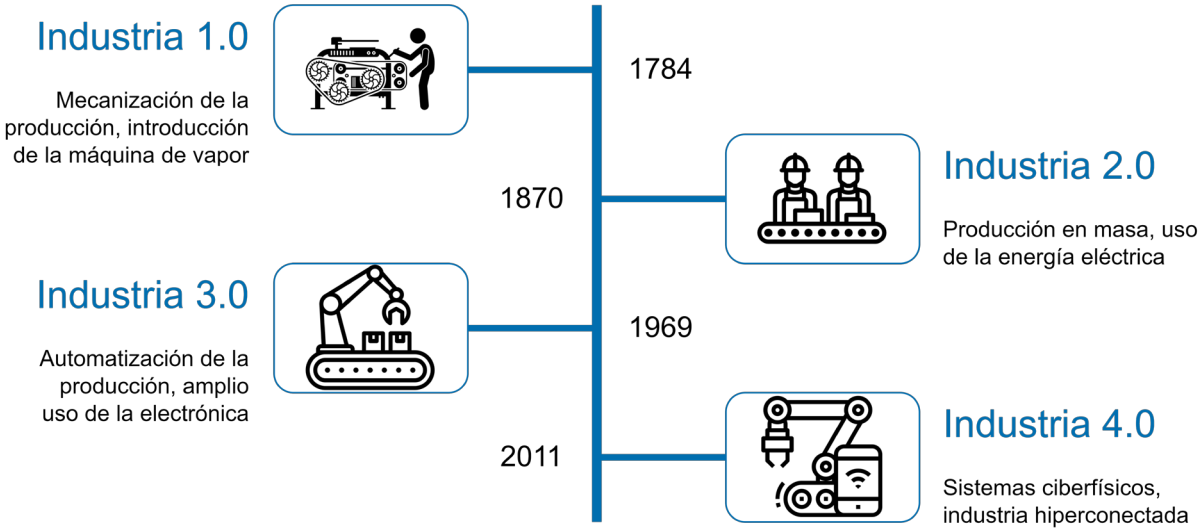

Figura 1.1: Línea de evolución de las revoluciones industriales

La cuarta revolución industrial es también conocida en Europa como Industria 4.0, fabricación inteligente o transformación digital de la industria. El concepto de Industria 4.0[1] fue presentado en 2011, poco antes de la inauguración de la Feria de Hannover (Culot et al., 2020; Hughes et al., 2022). La idea principal del concepto era denotar un tipo emergente de sistema productivo donde los espacios real y virtual se fusionan, en lo que se conoce como sistema ciberfísico[2]. Globalmente, existe una convergencia hacia este nuevo paradigma industrial, por lo que varios países han propuesto su propia estrategia análoga a la Industria 4.0 de Europa, por ejemplo, *Smart Manufacturing*[3] en Estados Unidos, *Society 5.0* en Japón, *Made in China 2025* en China, etc., evidenciando que este es un objetivo estratégico para el presente y futuro de la industria (Semeraro et al., 2021).

Las principales tendencias tecnológicas que inicialmente dieron soporte a los sistemas ciberfísicos fueron la Inteligencia Artificial (IA) industrial, la computación periférica y en la nube, la conectividad a la red celular de quinta generación (5G) en la fábrica, la robótica colaborativa, los sistemas intralogísticos autónomos y las infraestructuras de datos de confianza (Guerra et al., 2021; Kagermann & Wahlster, 2022). Desde ese momento, otras tecnologías, tales como

[1] Se ha adoptado el término Industria 4.0 derivado de su equivalente alemán Industrie 4.0.
[2] Se ha adoptado el término ciberfísico del término en inglés *Cyber-Physical*, de acuerdo con la Real Academia Española en términos similares tales como ciberespacio o cibernauta.
[3] Por coherencia con la terminología o nomenclatura en inglés y para evitar el uso excesivo de anglicismos, en varias partes de este trabajo se han utilizado términos originales en inglés. Dichos términos pueden ser distinguidos por el uso de la letra cursiva.

la fabricación aditiva, la realidad aumentada o los gemelos digitales, se han asociado cada vez más con la Industria 4.0, hasta el punto de ser consideradas actualmente parte de su núcleo (Alcácer & Cruz-Machado, 2019). La Figura 1.2 muestra algunas de las tecnologías habilitadoras de la Industria 4.0. Estas sientan las bases para la creación de sistemas de producción hiperconectados e inteligentes que ofrecen beneficios tales como el incremento en la productividad y la eficiencia, mejoras en la transferencia de conocimientos y el trabajo colaborativo, más flexibilidad y agilidad, mayores ingresos y más facilidad para cumplir las normativas (Butt, 2020).

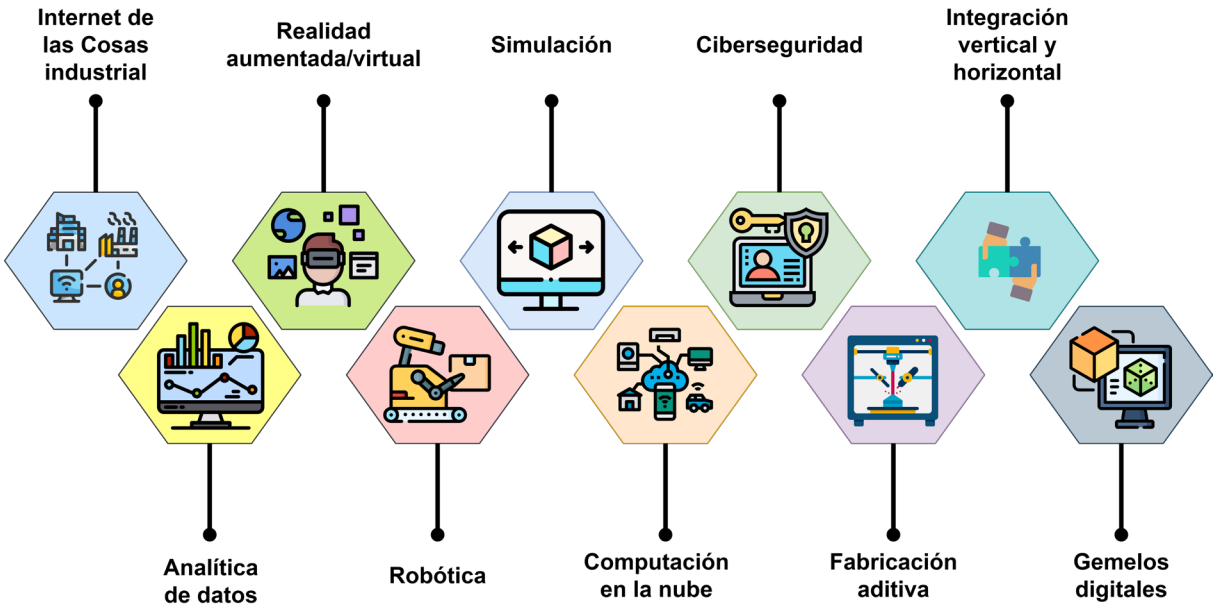

Figura 1.2: Tecnologías habilitadoras de la Industria 4.0

1.2. Motivación y contexto

El valor añadido derivado de la Industria 4.0 se estima que sea entre 70 y 140 miles de millones de euros para la industria manufacturera en Alemania hasta 2025 (Schuh et al., 2020). Otros estudios pronostican un incremento en la productividad industrial cercano al 30% en los países que adopten la Industria 4.0 (K. Zhou et al., 2015). Adicionalmente, el impacto previsto en algunos aspectos de la vida diaria es enorme. Por ejemplo, se estima que el 50% de los trabajos actuales en Suecia pueden ser reemplazados por el uso de la tecnología bajo el paradigma de la Industria 4.0 en los próximos 20 años, provocando una necesidad masiva de

recualificación de la fuerza laboral (C. Chauhan et al., 2021; Kurt, 2019). Sin embargo, a pesar de estas predicciones, la adopción de la Industria 4.0 aún se encuentra en una fase muy temprana. De acuerdo a una encuesta reciente, solo el 5.6% de las empresas alemanas en el ámbito de la ingeniería de maquinarias y equipos se encuentran en una etapa avanzada de adopción de la Industria 4.0, mientras que cerca del 80% de ellas aún operan a un nivel de principiante (L. Li et al., 2022). De manera similar, menos del 10% de las empresas chinas reportan haber llevado a cabo una fabricación inteligente transparente, predictiva y adaptativa. La mayoría de estas empresas aún están en la etapa de informatización, conexión y visualización (Luthra et al., 2020). Los bajos niveles de adopción de la Industria 4.0 son aún más evidentes cuando se analiza el panorama de las pequeñas empresas. En el caso de las pequeñas fábricas coreanas, solo el 5% ha adoptado la fabricación inteligente; mientras que, en otros países asiáticos solo el 20% de estas fábricas utilizan tecnologías como la IA o los gemelos digitales (Ghobakhloo & Iranmanesh, 2021).

Este nuevo paradigma plantea retos que las compañías deben superar para poder llevar a cabo una adopción exitosa. Algunos de estos desafíos afectan a todas las compañías por igual, mientras que otros dependen de las especificidades de cada caso. El estudio de estos retos es actualmente un tema de gran interés en el mundo académico. La Tabla 1.1 resume algunos estudios sobre este tema.

Tabla 1.1: Desafíos para la adopción de la Industria 4.0

Referencia	Retos analizados	Región/país	Empresas con bajo grado de madurez tecnológica
(Dobrowolska & Knop, 2020)	- exigencia de una mayor cualificación de la mano de obra - necesidad de cambios en los modelos de negocio	Europa	No
(Bogoviz et al., 2019)	- ausencia de políticas estatales de formación - insuficientes recursos financieros	África, Asia y Sudamérica	No

(Kamble et al., 2018)	- ausencia de estándares de implementación - falta de comprensión de las ventajas de las nuevas tecnologías - cuestiones de seguridad y privacidad - cambios en la estructura de las organizaciones	India	No
(Moeuf et al., 2018)	- desaprovechamiento del potencial de las nuevas tecnologías	No especificado	Sí
(Hughes et al., 2022)	- expectativas poco realistas de adopción sin inversión y apoyo público significativo	No especificado	Sí
(Ivanov et al., 2021)	- comprensión inadecuada de la naturaleza de la Industria 4.0 - falta de acceso al capital - falta de visión estratégica - falta de personal competente	Global	No
(Masood & Sonntag, 2020)	- brecha entre multinacionales y pequeñas empresas - gran número de tecnologías que hay que conocer - dificultad para evaluar a las pequeñas empresas	Reino Unido	Sí
(Raj et al., 2020)	- bajo nivel de madurez de la tecnología preferida - ausencia de estándares - calidad de los datos - resistencia al cambio	Global	No
(Masood & Egger, 2019)	- aceptación limitada por parte de los usuarios - falta de implementación de instrucciones adaptativas	No especificado	No
(Aoun et al., 2021)	- dificultades para la recolección, gestión y análisis de datos - preocupación por el espionaje industrial - agravamiento de la brecha de desigualdad de ingresos	No especificado	No

Al analizar estos desafíos se hace evidente que aún no existe un camino claro hacia la transformación digital. En este contexto, las empresas de fabricación con bajo

nivel de adopción del paradigma de Industria 4.0 deben hacer frente a desafíos inexistentes para las compañías que ya han comenzado su transformación digital. Las grandes compañías usualmente ya poseen un cierto grado de integración de tecnologías habilitadoras de la Industria 4.0, mientras que en el resto de las fábricas suele haber un bajo nivel de comprensión de los fundamentos y beneficios de estas tecnologías. Las empresas con bajo nivel de adopción de la Industria 4.0 también requieren de más recursos económicos para afrontar el proceso de transformación digital, debido a que, inicialmente, deben invertir en actualizar su infraestructura para, luego, invertir en las soluciones. Por otra parte, mientras que los empleados de ciertas empresas pueden tener cultura digital, en otras empresas los trabajadores pueden estar más reticentes a asimilar nuevas tecnologías y estar preocupados por perder su empleo debido a estas. Adicionalmente, las empresas con bajo nivel de adopción de la Industria 4.0 suelen carecer de experticia interna y conocimientos técnicos necesarios para implementar soluciones con el fin de transformar digitalmente la empresa, por lo que tienen que recurrir a socios externos. La situación de falta de personal altamente cualificado es común en países como España, donde la mayor parte del parque empresarial está constituido por microempresas con menos de 10 trabajadores (Confederación Española de la Pequeña y Mediana Empresa, 2023), como puede verse en la Figura 1.3.

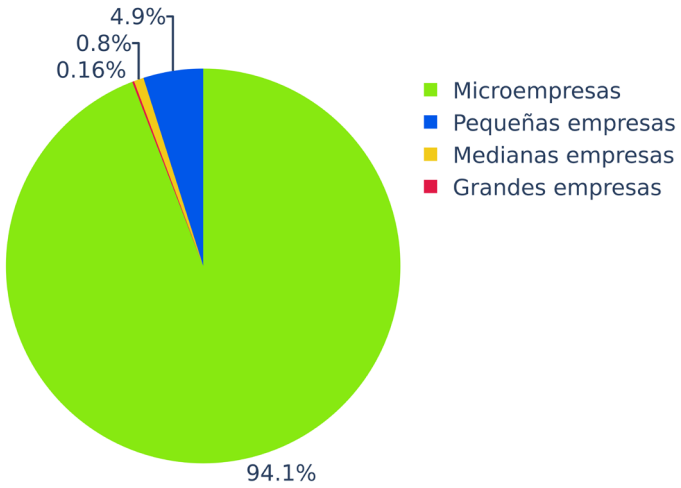

Figura 1.3: Estructura del parque empresarial en España

Con el objetivo de incentivar la transformación hacia la Industria 4.0 en las empresas con bajo nivel de adopción de este paradigma, existe una necesidad

creciente de soluciones que sean fáciles de asimilar, requieran una cualificación mínima del personal, puedan adaptarse a una amplia gama de procesos productivos y tengan en cuenta cuestiones prácticas. Para lograr este objetivo, se debe seguir trabajando en la investigación y desarrollo de dos temas: la automatización del modelado y optimización de los procesos productivos y la búsqueda de métodos generalizables para tratar la variabilidad de los procesos. Los avances en estos temas permitirán mejorar el rendimiento y la fiabilidad de los sistemas ciberfísicos y también contribuirán a la simplificación del proceso de desarrollo de los gemelos digitales (Ferrer et al., 2018).

Esta tesis está motivada por trabajos previos sobre sistemas ciberfísicos de procesos productivos, particularmente los enfocados en el desarrollo de gemelos digitales. Anteriormente se han utilizado enfoques genéricos para desarrollar soluciones lo más flexibles posible (Iliuță et al., 2023). Sin embargo, la generación de modelos del proceso productivo y su optimización involucran una serie de pasos cuya ejecución se realiza manualmente en la mayoría de los casos. Además, una gran parte de los trabajos que abordan la reconfiguración de los sistemas productivos se centra principalmente en el reordenamiento físico de los activos (Song et al., 2023). Este enfoque no es realista para muchas empresas, ya que generalmente implica tener que detener la producción y desplazar equipos pesados, lo que requiere tiempo y recursos adicionales. Asimismo, al centrarse estos trabajos en otros aspectos, generalmente no profundiza en las potencialidades de reparametrización de los procesos.

En esta tesis se asume que la reparametrización es la modificación mediante software de los parámetros de las variables más relevantes que influyen directamente en el comportamiento de los procesos productivos, principalmente en la eficiencia, productividad, calidad, entre otros. Se considera que la reparametrización mediante software de parámetros de máquinas y dispositivos un caso particular de la reconfiguración, que puede ser automática[4] o mediante recomendaciones inteligentes a operadores y tecnólogos[5].

Considerando las cuestiones descritas anteriormente, esta tesis contribuye al progreso en el estado del arte en el campo del modelado automático y la optimización de procesos productivos, así como en el desarrollo de soluciones para la reconfiguración mediante reparametrización automática de sistemas

[4] La actuación o modificación de los parámetros de las máquinas y/o dispositivos se realiza sin intervención humana.
[5] El ser humano actúa como un elemento activo en el bucle de actuación.

ciberfísicos. Una de las novedades del trabajo es que se enfoca en integrar los gemelos digitales e IA en empresas con bajo nivel de adopción del paradigma de la Industria 4.0. Con este fin, se presenta una metodología de aprendizaje automático automatizado (AutoML), la cual permite generar modelos y optimizar procesos generando recomendaciones inteligentes para el personal con conocimientos mínimos de estos temas. También se presenta una segunda metodología enfocada a la reconfiguración basada en la Lógica Borrosa, y específicamente en un sistema de inferencia borrosa para garantizar el mantenimiento de los niveles productivos. A diferencia de trabajos anteriores, estas metodologías están enfocadas a empresas con bajo nivel de adopción de la Industria 4.0, donde existe una infraestructura y conocimientos limitados sobre las tecnologías de este paradigma, pero hay una gran necesidad de soluciones que sean fáciles de asimilar e integrar.

El marco de trabajo para el desarrollo de esta tesis es la línea de investigación del Grupo de Control Avanzado de Máquinas, Procesos y Entornos Complejos (GAMHE), adscrito al Centro de Automática y Robótica (CAR), un centro de investigación con titularidad compartida entre la Universidad Politécnica de Madrid (UPM) y el Consejo Superior de Investigaciones Científicas (CSIC). GAMHE cuenta con más de 30 años de experiencia en el desarrollo y aplicación de métodos de modelado, control y supervisión basados en IA, enfocados, especialmente, a los sistemas ciberfísicos industriales. Inicialmente, la tesis se desarrolló como parte del proyecto europeo "*platform-enabled KITs of arTificial intelligence FOR an easy uptake by SMEs*" (KITT4SME) y fue continuada bajo el marco de los proyectos nacionales "Auto-Reconfiguración en sistemas ciberfísicos industriales basada en gemelos digitales e IA. Métodos y aplicación en una línea piloto de Industria 4.0" (SELFRECO) y "Generador de recomendaciones inteligentes centradas en el ser humano en fabricación Cero Defectos para una reducción de residuos" (iGENZERO). Estos proyectos tienen como principal objetivo proveer a las empresas de fabricación con bajo nivel de madurez tecnológica paquetes digitales para la introducción de manera eficaz de la IA en sus procesos productivos. Específicamente, KITT4SME busca proporcionar servicios de IA sobre una arquitectura basada en una plataforma digital estándar de código abierto, que sea personalizable por módulos.

1.3. Objetivos de la investigación

El objetivo principal de esta tesis es diseñar, implementar y validar dos métodos basados en IA dirigidos hacia la reconfiguración y supervisión automática de

sistemas ciberfísicos basada en gemelos digitales e Inteligencia Artificial. El primer método es una metodología de aprendizaje automático automatizado para el modelado y la optimización de procesos productivos en la industria de fabricación, enfatizando en las cuestiones tecnológicas y prácticas, para contribuir al desarrollo de sistemas ciberfísicos basados en gemelos digitales; mientras que el segundo se basa en un sistema de inferencia borroso para garantizar la productividad y la eficiencia de los procesos productivos en la industria de manufactura.

Para lograr este objetivo principal, se abordarán cuatro objetivos específicos. Cada uno de estos objetivos específicos se compone de varias metas parciales. La manera de abordar estos objetivos específicos ayudará a identificar los desafíos clave y las posibles estrategias para resolver eficazmente el problema planteado. Los objetivos específicos son los siguientes:

1. Diseñar una metodología genérica de aprendizaje automático automatizado para la generación de modelos que utilice directamente los datos de producción y las especificaciones del proceso y sea capaz de proporcionar parametrizaciones óptimas para el proceso productivo:
   (a) Identificar los pasos necesarios para generar modelos que representen a un proceso productivo a partir de los datos de producción.
   (b) Definir una estructura de datos de entrada al método de aprendizaje automático automatizado.
   (c) Desarrollar una metodología de aprendizaje automático automatizado para la generación de modelos de los procesos productivos.
   (d) Explorar estrategias para mitigar los efectos de los modelos de baja fiabilidad.
   (e) Integrar en la metodología procedimientos de optimización prometedores.
2. Diseñar estrategias para hacer frente a los cambios en los procesos productivos:
   (a) Identificar los diferentes tipos de cambios que pueden afectar la capacidad de predicción de los modelos generados.
   (b) Desarrollar una estrategia de actualización de los modelos para mantener su fiabilidad.

(c) Desarrollar una metodología genérica utilizando sistemas de inferencia borrosa para mitigar cambios no deseados en la productividad.

3. Diseñar un gemelo digital de un sistema productivo:
   (a) Diseñar la estructura de un gemelo digital de la línea piloto de GAMHE.
   (b) Desarrollo de un gemelo digital de la línea piloto de Industria 4.0 situada en el laboratorio de GAMHE.
4. Integrar de la solución en un sistema ciberfísico y su posterior validación:
   (a) Implementar las metodologías de aprendizaje automático automatizado, sistema de inferencia borroso y las estrategias complementarias en módulos funcionales.
   (b) Integrar los módulos en la plataforma KITT4SME y, posteriormente, desplegarlos en el laboratorio de GAMHE.
   (c) Utilizar el módulo de aprendizaje automático automatizado como capa de inteligencia del gemelo digital que representa de la línea piloto del laboratorio de GAMHE.
   (d) Evaluar las soluciones desarrolladas.

1.4. Esquema de la tesis

A continuación, se presenta una breve descripción del contenido de los siguientes capítulos de esta tesis:

- Capítulo 2: Se realiza un análisis bibliográfico de las principales temáticas de la Industria 4.0 abordadas en esta tesis. Primeramente, se tratan los sistemas ciberfísicos, en particular los enfocados a la industria. A continuación, se presentan los gemelos digitales como parte del espectro de tecnologías para el desarrollo de sistemas ciberfísicos industriales, donde se introduce el concepto de gemelo digital de 4 componentes. Seguidamente, se trata el uso de la IA en la fabricación, donde se remarca que los métodos que forman parte de esta tecnología son los que dan soporte a la creación de la capa de inteligencia en los gemelos digitales. Por último, se exploran trabajos relacionados con el tema de investigación tratado.

- Capítulo 3: Se presenta la plataforma KITT4SME que sirve como marco de trabajo para el desarrollo de esta tesis, donde se deberán implementar las propuestas. Además, se describen a grandes rasgos dichas propuestas, las cuales consisten en la creación de dos módulos basados en IA y en su aplicación a una línea piloto de Industria 4.0.
- Capítulo 4: Se describe la metodología diseñada para el módulo de reparametrización óptima de procesos productivos, la cual está basada en un flujo de trabajo de aprendizaje automático automatizado. También se describe la estructura del gemelo digital de la línea piloto y se profundiza en su integración con el módulo de reparametrización óptima diseñado. Por último, se especifican las herramientas seleccionadas para la implementación tanto del módulo de reparametrización óptima de la plataforma KITT4SME, así como de los componentes necesarios para la línea piloto.
- Capítulo 5: Se describe la metodología diseñada para el módulo de reconfiguración dirigido a la corrección de las desviaciones en la productividad basado en un sistema de inferencia borroso. Se detallan las herramientas utilizadas para el diseño, la implementación e integración en la línea piloto.
- Capítulo 6: Se realiza una descripción de la línea piloto de Industria 4.0 donde se validarán las soluciones desarrolladas en esta tesis. A continuación, se evalúa el módulo de reparametrización automática y óptima que se comporta como una capa de inteligencia del gemelo digital de la línea piloto. Posteriormente, se comprueba la capacidad del gemelo digital de coevolucionar[6] con el proceso productivo. También se presenta la interfaz visual desarrollada para el gemelo digital. Finalmente, se evalúa el funcionamiento de la reconfiguración automática para la corrección de desviaciones en la productividad mediante la modificación de parámetros de máquinas/robots y dispositivos en la línea piloto GAMHE.

[6] En el contexto de este trabajo se entiende por coevolución a la capacidad de la parte virtual de un sistema ciberfísico de reflejar los cambios que ocurren en la parte física y viceversa.

- Capítulo 7: Se presentan las conclusiones de esta tesis, derivadas de los resultados obtenidos. Además, se analizan futuras líneas de trabajo, identificando áreas potenciales de interés en las que se podrían desarrollar investigaciones posteriores.

2. Estado del arte. Una aproximación científico-técnica

2.1. Introducción

En la actualidad, la industria está atravesando una serie de cambios sin precedentes, guiados por el surgimiento de tecnologías avanzadas que permiten desarrollar sistemas de producción de mayor complejidad y flexibilidad, a la vez que se busca incrementar su eficiencia. En este capítulo se abordan elementos clave para comprender las transformaciones que están ocurriendo en la industria moderna, partiendo de los sistemas ciberfísicos como núcleo fundamental de estos cambios. Además, se profundiza en ciertas técnicas cada vez más extendidas en este contexto, como son los gemelos digitales y el aprendizaje automático. Se presentan las características de estos elementos, determinadas aplicaciones y los desafíos que aún no han sido resueltos por los trabajos desarrollados hasta el momento.

2.2. Sistemas ciberfísicos industriales

Los sistemas ciberfísicos son aquellos donde se integran los procesos del mundo real con los que ocurren en el espacio computacional (Gürdür Broo et al., 2021). Estos se caracterizan por ser sistemas integrados en red que permiten dotar a los objetos físicos de capacidades de procesamiento de información, así como de comunicación con su entorno. En ellos se utilizan sensores y unidades de cómputo para supervisar y analizar los procesos del mundo físico y, en caso de ser necesario, también se incorporan actuadores y mecanismos de control o toma de decisiones con la capacidad de influir sobre dichos procesos (Klötzer et al., 2017). Este tipo de sistemas se encuentran presenten en diversas áreas del conocimiento y de la innovación tecnológica como pueden ser la agricultura, los sistemas financieros, los sistemas de transporte, la salud, la fabricación, el sector energético y las ciudades inteligentes, entre otras (Jahromi & Kundur, 2020).

Los sistemas ciberfísicos industriales son un subconjunto de los sistemas ciberfísicos diseñados específicamente para los entornos industriales con el objetivo de transformar las cadenas de valor convencionales mediante el desarrollo

de productos y procesos inteligentes capaces de cubrir la brecha entre el mundo físico y el virtual (K. Zhang et al., 2023). La Figura 2.1 representa la interconexión entre los espacios físico y virtual en este tipo de sistemas. Los sistemas ciberfísicos industriales utilizan sensores, controladores, actuadores, máquinas, robots y aplicaciones informáticas para recolectar datos del mundo real, procesarlos y utilizarlos para el control de los procesos productivos (Pivoto et al., 2021). La principal ventaja de estos sistemas es que la estrecha cooperación entre los componentes cibernéticos y físicos aumenta la autonomía, eficiencia, funcionalidad, fiabilidad y flexibilidad en el entorno industrial (Ding et al., 2019). El gran número de investigaciones relacionadas con los sistemas ciberfísicos industriales evidencia el interés de la comunidad científica por estos sistemas, situándolos como el elemento central de la fabricación inteligente.

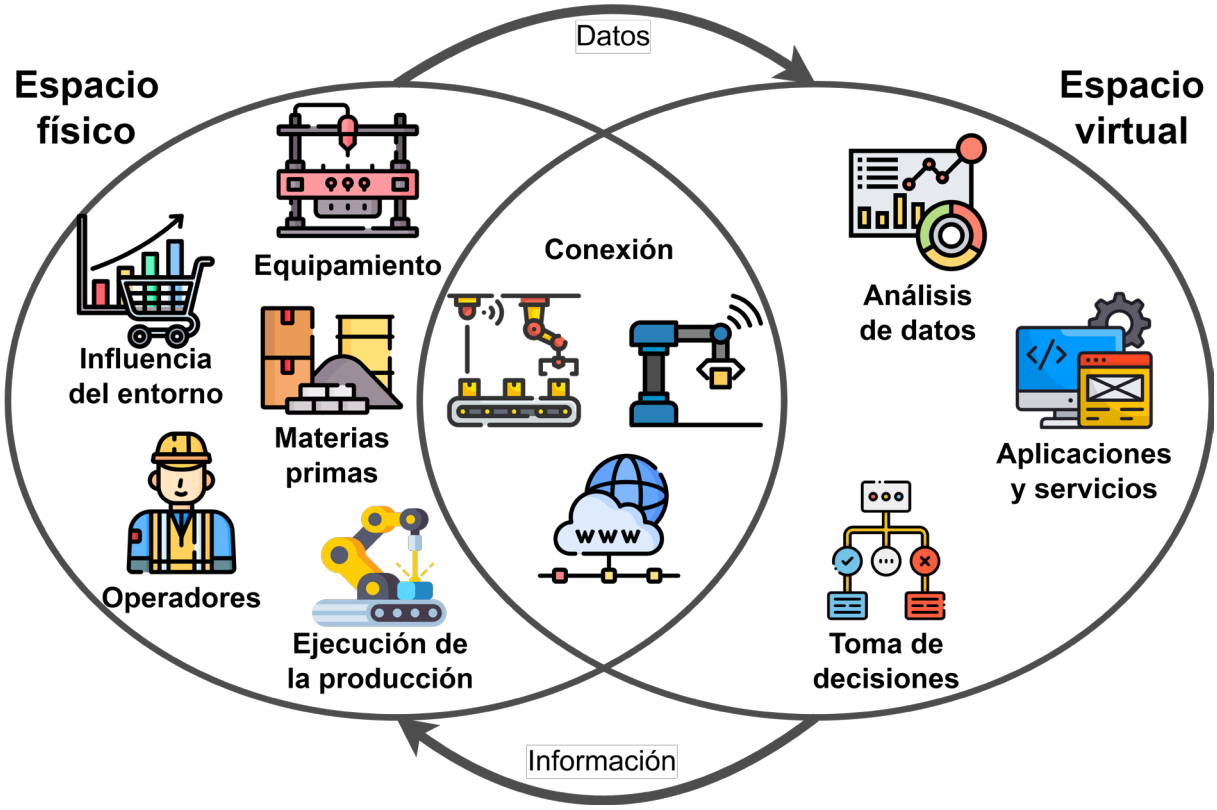

Figura 2.1: Sistemas ciberfísicos industriales

Aunque las tecnologías que permiten el desarrollo de los sistemas ciberfísicos industriales están bien identificadas, los investigadores aún debaten las características que estos sistemas deben tener. Un estudio reciente proporciona un

listado de propiedades de los sistemas ciberfísicos industriales como las más prominentes en la literatura relacionada (Napoleone et al., 2020):

- Encapsulado de la complejidad/heterogeneidad
- Interoperabilidad
- Conectividad
- Comunicación
- Capacidad de conexión en red
- Orientación a los servicios
- Modularidad
- Autonomía
- Capacidades propias
- Descentralización
- Integración
- Virtualización
- Capacidades de tiempo real
- Capacidades computacionales
- Inteligencia
- Cooperación
- Colaboración
- Reconfiguración dinámica
- Adaptabilidad

Actualmente existe la tecnología necesaria para el desarrollo de sistemas ciberfísicos industriales. Sin embargo, la aplicación de estas tecnologías de forma práctica a gran escala para construir fábricas inteligentes basadas en sistemas ciberfísicos industriales está lejos de convertirse en una realidad (Napoleone et al., 2020). Una de las barreras para la transformación digital en la industria manufacturera es la falta de metodologías de reconfiguración y adaptación bien establecidas para la reparametrización óptima de los sistemas ciberfísicos industriales (Villalonga et al., 2020). Es decir, en muchos casos, aun cuando se cuenta con el equipamiento necesario, no se poseen los métodos para explotar todo su potencial.

2.2.1. Arquitecturas de referencia

Actualmente, los principios subyacentes de la Industria 4.0 se encuentran bien establecidos y son profundamente considerados en la fabricación debido a sus beneficios potenciales. Sin embargo, inicialmente la estructura de la Industria 4.0 no estaba definida claramente, sino que era más bien una visión para el futuro de la fabricación que abarcaba varias tecnologías emergentes. Para facilitar el desarrollo de soluciones enfocadas hacia la Industria 4.0, varias iniciativas propusieron sus propias arquitecturas de referencia. Las siguientes subsecciones cubren brevemente algunas de dichas arquitecturas de referencia.

2.2.1.1. Reference Architectural Model for Industry 4.0

En Europa, el gobierno alemán promovió una arquitectura de referencia estandarizada para el diseño y desarrollo de proyectos de Industria 4.0 llamado *Reference Architectural Model for Industry 4.0* (RAMI 4.0). Esta arquitectura enfatiza los aspectos tecnológicos de la Industria 4.0 al combinar y mapear todos sus elementos y componentes en una estructura tridimensional como se muestra en la Figura 2.2 (Bretz et al., 2022). Desde su introducción, RAMI 4.0 se ha convertido en una arquitectura de referencia para la Industria 4.0 ampliamente conocida y ha sido adoptada por un gran número de empresas y organizaciones de todo el mundo.

La dimensión de niveles jerárquicos de RAMI 4.0 está orientada a la integración de los sistemas de control de la empresa (Baptista & Barata, 2021). Los productos, en el nivel más bajo, pueden participar como componentes activos en el sistema productivo al estar dotados de capacidades de comunicación. Los dispositivos de campo consisten en los sensores y actuadores que interactúan directamente con el producto, mientras que los dispositivos de control se encargan de procesar la información recopilada por los dispositivos de campo y, basándose en esta, toman decisiones sobre la ejecución del proceso productivo. El nivel de estación incluye máquinas herramientas, robots o vehículos inteligentes. El nivel de centros de trabajos se refiere a los diferentes departamentos y plantas productivas que forman parte de una empresa. Finalmente, el mundo conectado se refiere a las redes externas como las asociaciones con clientes y socios comerciales, así como a servicios basados en internet.

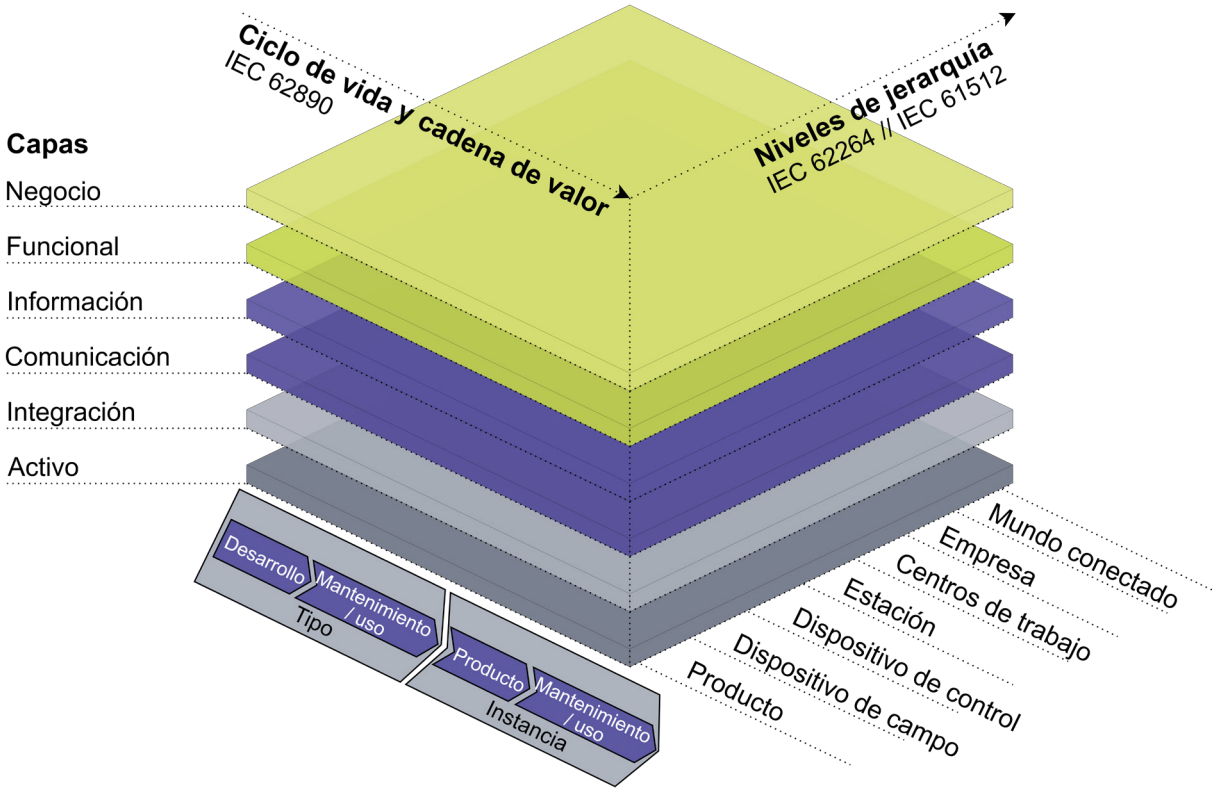

Figura 2.2: Modelo RAMI 4.0

La dimensión del ciclo de vida y el flujo de valor de RAMI 4.0 representa la gestión de los activos, tanto para productos como para sistemas, desde su desarrollo hasta su explotación. En esta dimensión los tipos e instancias son conceptos diferentes. Por ejemplo, durante las etapas de diseño de un producto, este es considerado como un tipo, mientras que, al comenzar el proceso real de fabricación, pasa a ser una instancia. Es decir, los productos son instancias de un tipo. Por otra parte, las capas de interoperabilidad, también llamadas capas funcionales, permiten descomponer los elementos de la Industria 4.0 en las diferentes funciones que llevan a cabo, desde las operaciones físicas hasta las funciones relacionadas con el modelo de negocio a nivel empresarial.

2.2.1.2. Smart Manufacturing Ecosystem

Smart manufacturing es el nombre más comúnmente utilizado en Estados Unidos para designar a los procesos productivos que han incorporado tecnologías que permiten la digitalización de las fábricas y la total interconexión de sus elementos. El creciente interés por estos sistemas en Estados Unidos está motivado por la necesidad de mejorar la calidad, productividad y rentabilidad de los productos en

un mercado cada vez más competitivo. En el proceso de búsqueda y creación de un marco de referencia estándar que favoreciese el desarrollo de estos sistemas, el Instituto Nacional de Estándares y Tecnología del Departamento de Comercio de Estados Unidos propuso el *Smart Manufacturing Ecosystem* (Y. Lu et al., 2015). La Figura 2.3 muestra la arquitectura que se presentó en esta propuesta. Esta arquitectura comprende tres dimensiones de los sistemas de fabricación inteligente: el producto, el sistema de producción y el negocio; y se enfoca en la interrelación de estas dimensiones con la pirámide de fabricación. De esta manera se busca conseguir un control avanzado en el proceso productivo y una toma de decisiones óptima en la empresa.

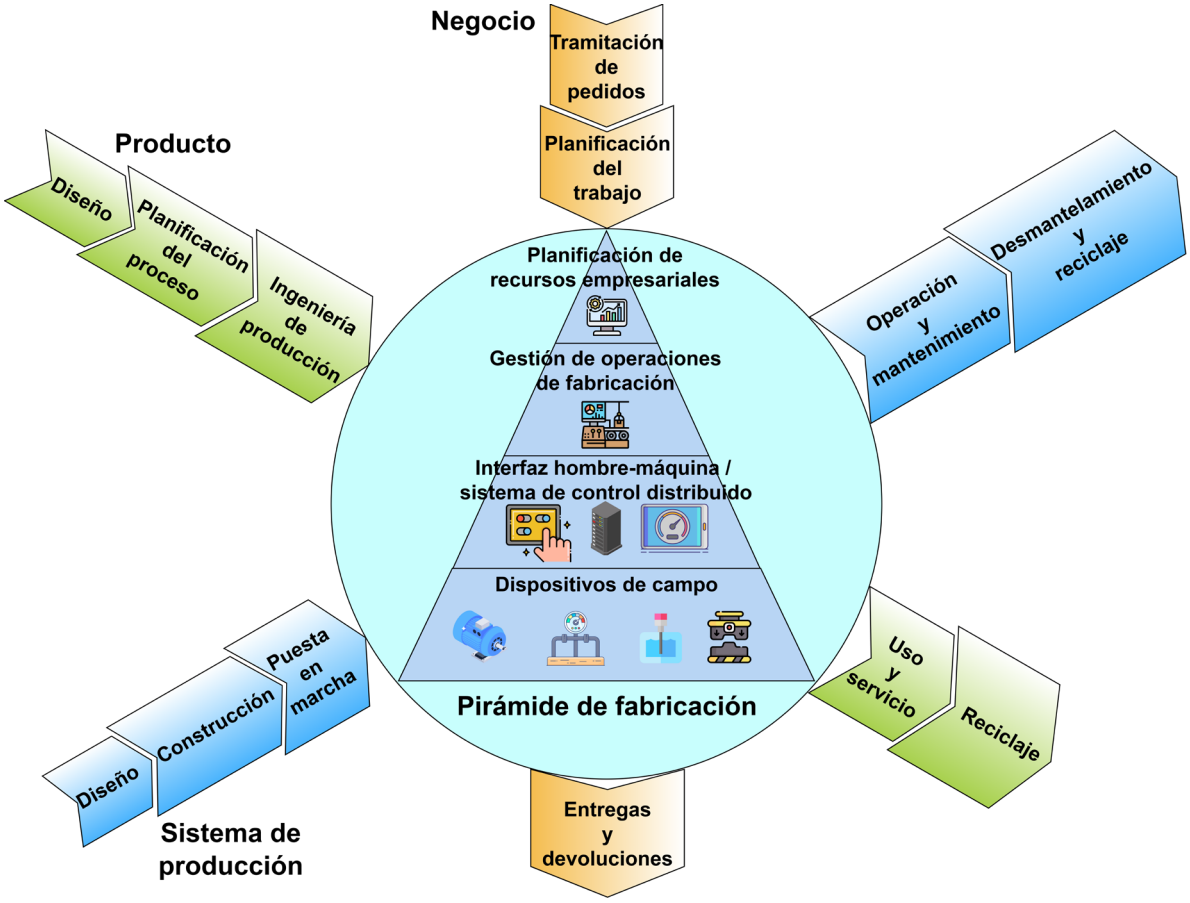

Figura 2.3: Modelo del ecosistema *Smart Manufacturing*

La dimensión del producto abarca 6 fases desde su concepción hasta el fin de su vida útil: (1) diseño, (2) planificación del proceso, (3) ingeniería de producción, (4) fabricación, (5) uso y servicio y (6) reciclaje. Las tecnologías asistidas por computador (CAx) son ampliamente usadas, sobre todo en las primeras fases de

esta dimensión, ya que permiten crear modelos del producto de gran precisión, fabricarlo y comenzar a explotarlo en un corto tiempo. Por otra parte, el desarrollo de herramientas de trazabilidad ha sido de gran ayuda para implementar los flujos de información necesarios para el control del producto, especialmente una vez que ha comenzado su explotación.

La dimensión del sistema de producción se enfoca en el diseño, la construcción, el funcionamiento y, una vez que ha concluido la vida útil, el desmantelamiento de todo el proceso productivo. En esta dimensión también se utilizan ampliamente las tecnologías CAx para las fases previas a la operación de la planta, mientras que, en las fases posteriores, tienen más relevancia las herramientas avanzadas de automatización, operación y mantenimiento.

Por último, la dimensión de negocio cubre las relaciones entre los clientes y los proveedores de servicio de manufactura. Aquí se encuentran las herramientas que permiten la especificación de requisitos y las comunicaciones entre aplicaciones empresariales. Además, en esta dimensión se abordan las interacciones necesarias para la gestión de la cadena de suministro.

2.2.1.3. Intelligent Manufacturing System Architecture

El gobierno chino, tomando como referencia los planes estratégicos desarrollados en Europa con respecto a la Industria 4.0, decidió lanzar su propia iniciativa llamada *Made in China 2025*. Esta iniciativa tiene como objetivo transformar la industria china para aumentar la capacidad de fabricación y producir productos de alta tecnología (Q. Li et al., 2018). En este contexto se define la arquitectura *Intelligent Manufacturing Systems Architecture* (IMSA) como referencia para la implementación del nuevo paradigma industrial. La representación de esta arquitectura se muestra en la Figura 2.4. IMSA, al igual que RAMI 4.0, también define una arquitectura tridimensional compuesta, en este caso, por el ciclo de vida, la jerarquía del sistema y las funciones inteligentes. La diferencia más notable entre estas dos arquitecturas es que en IMSA no se separa el ciclo de vida en dos etapas, sino en una serie de cinco pasos que abarcan desde el diseño hasta la explotación del producto.

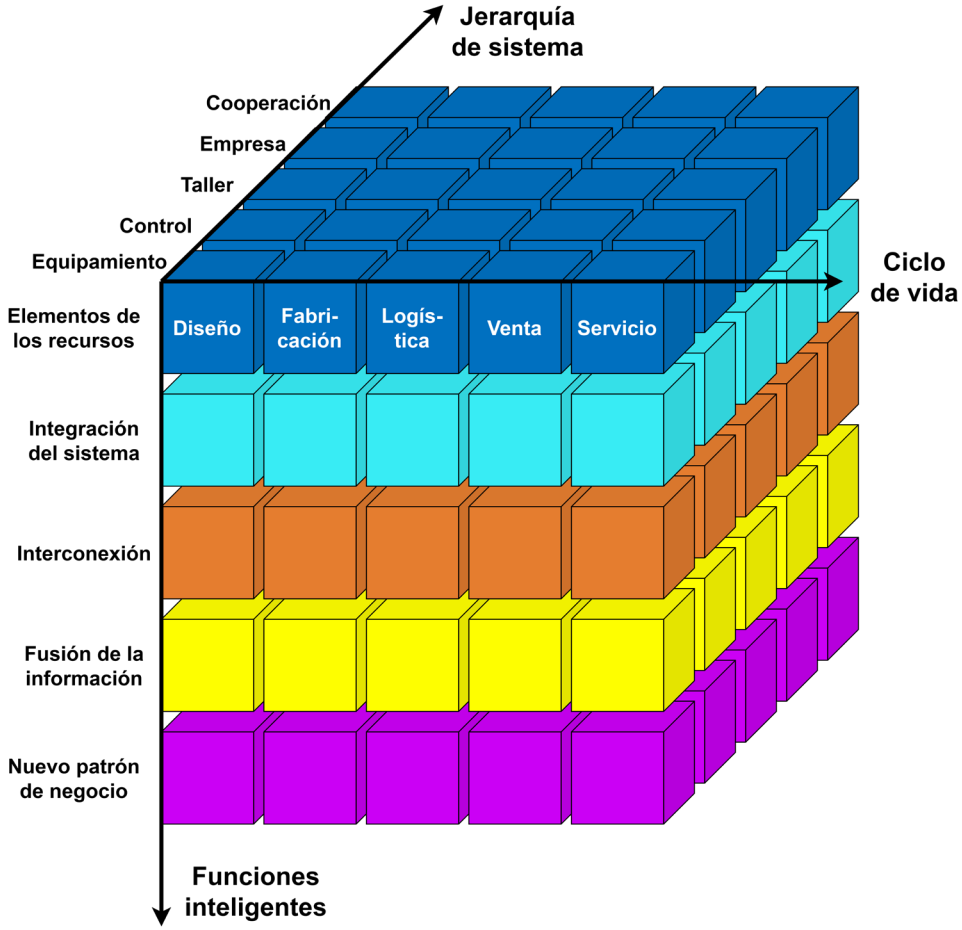

Figura 2.4: Modelo IMSA

2.2.2. Limitaciones de las arquitecturas de referencia

Al analizar las particularidades de cada una de estas arquitecturas queda claro que todas giran en torno al concepto de sistema ciberfísico (Pivoto et al., 2021). Sin embargo, a pesar de proporcionar un marco conceptual de gran valor, estas arquitecturas de referencia presentan severas limitaciones en cuanto a los aspectos prácticos del desarrollo de sistemas ciberfísicos industriales. Estas arquitecturas presentan una visión global de la Industria 4.0, pero faltan las definiciones de los microservicios que deben estar presentes y, además, no se brindan directrices sobre cómo implementar las soluciones (Moghaddam et al., 2018). Es decir, las arquitecturas existentes solo son marcos abstractos, por lo que existe la necesidad de directrices más concretas que muestren como implementar, por ejemplo, el procesamiento de la información en las aplicaciones de la Industria 4.0 (Kirmse et al., 2019).

Por otra parte, a pesar de que estas arquitecturas de referencia indican el empleo de algunos estándares, estos no cubren todos los aspectos relacionados y solamente marcan, en última instancia, un grupo de pautas que deben ser verificables una vez que se ha desarrollado la solución. Entre los estándares que aún deben ser definidos para para este tipo de sistemas se encuentran los relacionados con ciberseguridad, servicios de fabricación basados en la nube, la integración de la cadena de valor y la analítica de datos (Wang et al., 2021).

Esta ausencia de estándares y de un camino claro para la implementación de las Industria 4.0 provoca que cada empresa se trace su propia ruta hacia la transformación digital como se muestra en la Figura 2.5. En este contexto, existe una mayor complejidad para completar los ciclos de recopilación, procesamiento, combinación y análisis de datos, así como para devolver los resultados a las fuentes (Kirmse et al., 2019).

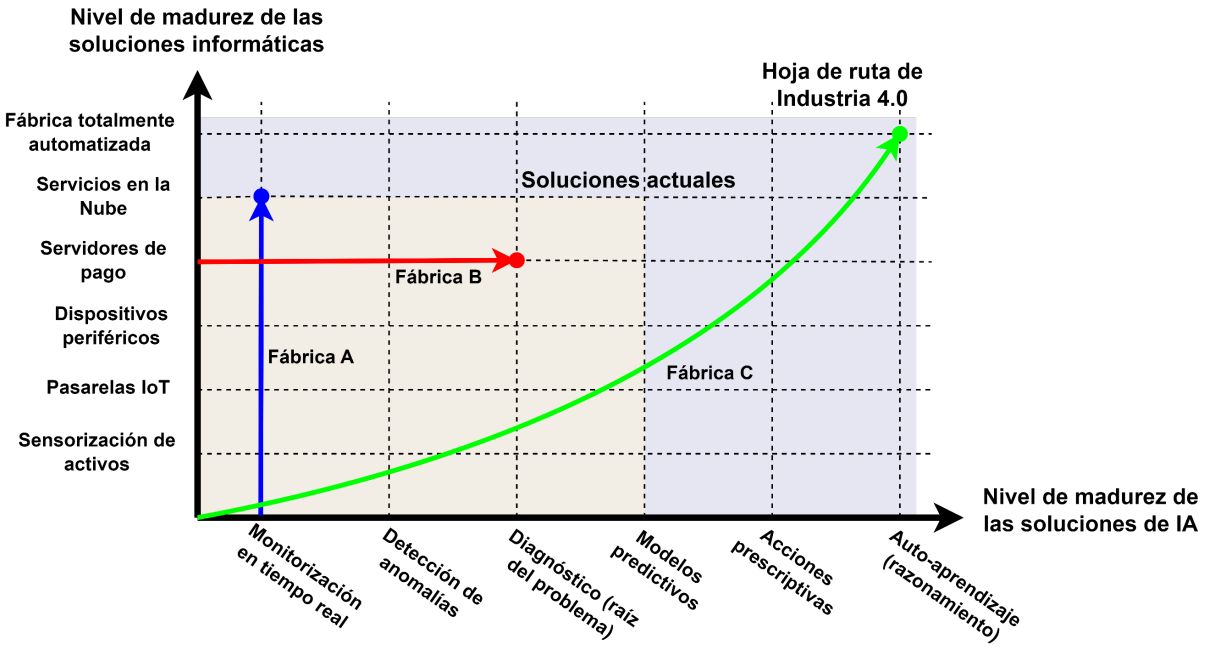

Figura 2.5: Rutas hacia la transformación digital

2.2.2.1. Aplicaciones basadas en los modelos de referencia

Algunos ejemplos de implementación de soluciones basadas en las arquitecturas de referencia presentadas anteriormente en empresas con un bajo nivel de adopción del paradigma de la Industria 4.0 incluyen procesos de producción de

interruptores (G. Chen et al., 2020), de casquillos para cierres de ejes mecánicos (Y. Lu & Asghar, 2020), de tableros (Leng et al., 2019), de componentes mecánicos prensados (Canizo et al., 2019), luminarias industriales (Garcia et al., 2019), entre otros. Sin embargo, estas constituyen soluciones puntuales, que al no seguir una metodología establecida son muy complejas de adaptar y replicar en otros procesos productivos. Además, en muchos casos, las soluciones no se encuentran bien documentadas o no existe una evaluación fidedigna de su verdadero impacto en el proceso productivo.

2.3. Gemelos digitales en la fabricación

Entre las tecnologías que dan soporte a la Industria 4.0 se encuentran los gemelos digitales. Estos son representaciones virtuales de sus contrapartes físicas con las que se encuentran en una combinación orgánica, en la cual existe una comunicación recíproca y una coevolución a través de interacciones en ambas direcciones (Qi et al., 2021). Los gemelos digitales ofrecen un medio único para lograr la integración ciberfísica que es cada vez más aceptado en distintos ámbitos. Estos pueden proporcionar perspectivas de gran valor sobre el rendimiento de los procesos, lo que ayuda a tomar decisiones mejor fundamentadas. También pueden ayudar a detectar posibles problemas en fases tempranas del desarrollo de sistemas, probar y perfeccionar nuevas ideas antes de implantarlas en la práctica y optimizar los procesos que representan.

El modelo generalizado de gemelo digital es el que consta de tres componentes primarios tal como se muestra en la Figura 2.6: (a) entidades físicas en el espacio real, (b) las conexiones de datos e información que enlazan a las entidades virtuales y físicas, y (c) entidades virtuales en el espacio digital (Qi & Tao, 2018). Sin embargo, debido a la continua ampliación y actualización de los requisitos de las aplicaciones, el desarrollo de gemelos digitales presenta nuevas tendencias y demandas, especialmente las relacionadas con un grupo de servicios que el gemelo digital debe ofrecer (Haber et al., 2019; Mohammed et al., 2022). Para cumplir con estos nuevos requisitos se han propuesto modelos extendidos formados por cuatro componentes, como se muestra en la Figura 2.7, e incluso otros formados por cinco componentes. Los modelos de cuatro componentes están formados por las entidades físicas, los modelos virtuales, los servicios inteligentes[7] y las conexiones

[7] También denominados como capa de inteligencia

entre todos estos elementos, mientras que los de cinco componentes agregan también los datos como un elemento más (Barbieri et al., 2021; Tao et al., 2018).

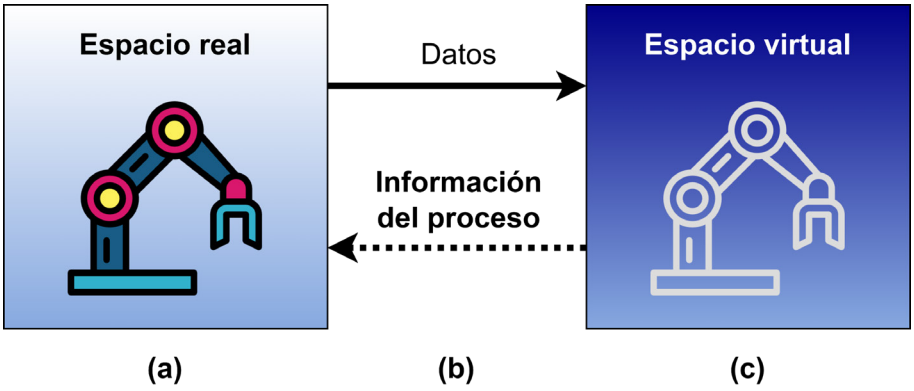

Figura 2.6: Representación básica de un gemelo digital

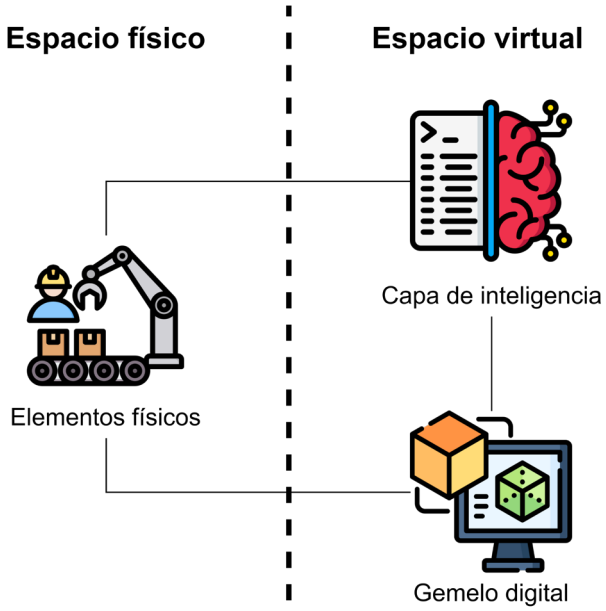

Figura 2.7: Representación de un gemelo digital de cuatro elementos

Los gemelos digitales son particularmente útiles en el contexto de los sistemas de fabricación inteligentes y reconfigurables. El enfoque tradicional para el desarrollo de un sistema de fabricación se centra en producir un solo tipo/generación/familia de productos, y por esta razón cuando es necesario rediseñar o reconfigurar el sistema para adaptarlo a un nuevo tipo/generación/familia de productos se incurre en costes exorbitantes. Dado que la reconfigurabilidad y adaptabilidad del diseño

son ambiguas en términos de cambios indeterministas e impredecibles del producto, es difícil identificar un diseño que sea adaptable para diferentes tipos/generación/familia de productos. Como resultado, un diseño óptimo para un producto determinado puede no ser ejecutable para la siguiente generación del producto en un sistema tradicional (Leng et al., 2021). Sin embargo, en los sistemas orientados a la reconfigurabilidad, los gemelos digitales proveen una plataforma para la evaluación rápida de las estrategias de reconfiguración basadas en arquitecturas abiertas de máquinas herramientas y grandes volúmenes de datos (Jeon & Suh, 2018; Leng et al., 2020). Esta es un área de investigación con muchas cuestiones que aún necesitan ser respondidas, tales como la necesidad de integrar técnicas de optimización computacional en los gemelos digitales para mejorar la efectividad y eficiencia del diseño de manera que la reconfiguración pueda ser automatizada y optimizada inteligentemente (Kurniadi et al., 2018; C. Zhang et al., 2021).

Por otra parte, en la práctica, los gemelos digitales deben hacer frente a otros desafíos. A pesar del gran interés de las empresas en desarrollar gemelos digitales para su quehacer cotidiano, la mayoría de ellas no están familiarizadas con las tecnologías y herramientas para hacerlo. Este no es un problema exclusivo de la industria, muchos investigadores y practicantes tampoco tienen claro que tecnologías y herramientas utilizar (Qi et al., 2021). Además, los gemelos digitales son sistemas complejos que requieren largos períodos de desarrollo, implementación y optimización. Actualmente, para crear un gemelo digital, un especialista o grupo de especialistas deben recolectar y fusionar diversos tipos de datos, así como modelar las diferentes partes del objeto o sistema, desarrollar los servicios requeridos y, finalmente, integrar todos estos componentes en una solución funcional.

2.3.1. Niveles de diseño de gemelos digitales en la fabricación

Generalmente, los gemelos digitales en la fabricación se diseñan enfocados a uno de los siguientes tres niveles: producto, proceso o sistema de sistemas (Onaji et al., 2022). Un gemelo digital de un producto normalmente se centra en la modelación física (forma geométrica, materiales, características), así como en las interacciones con otros elementos. Un gemelo digital de un proceso está más orientado hacia los componentes y equipos que forman parte del proceso de fabricación. En estos la modelación física pasa a un papel secundario, mientras que las características de comportamiento tales como la capacidad productiva, el flujo de trabajo y las

operaciones pasan a tener mayor relevancia. Este nivel comprende desde los procesos que se llevan a cabo utilizando un solo equipo hasta los que involucran todo un taller (Bao et al., 2019). Finalmente, los gemelos digitales a nivel de sistema de sistemas se centran en la colaboración entre procesos, la cual puede ocurrir dentro de una misma industria entre diferentes líneas de producción o entre diferentes plantas industriales (Taylor et al., 2020). Estos prestan más atención a la interoperabilidad y los servicios como, por ejemplo, las cadenas de suministro.

De los tipos de gemelos digitales analizados, los que más atención han tenido en la fabricación han sido los enfocados a nivel de proceso. Aprovechando las ventajas que propicia este tipo de gemelo digital en cuanto a la monitorización y simulación de la producción, han sido aplicados para optimizar parámetros de máquinas en líneas de ensamblaje (C. Zhang et al., 2019), replanificar la producción frente a la ocurrencia de fallas (Barbieri et al., 2021; Villalonga et al., 2021), detectar cuellos de botella en líneas productivas y reorganizarlas para disminuir la tasa de bloqueos (Gong et al., 2019), determinar la secuencia óptima en la cual realizar operaciones de mecanizado (G. Zhou et al., 2020), etc. A pesar de no ser un requisito indispensable, la mayoría de estos sistemas integran servicios basados en IA, siguiendo el modelo de cuatro componentes. Por esta razón, agilizar el proceso de desarrollo de estos servicios y garantizar su fiabilidad está ganando cada vez más relevancia.

2.4. Inteligencia Artificial en los procesos de fabricación

El término IA se refiere a la replicación de inteligencia similar a la humana en máquinas y computadoras en una manera que simule las funciones cognitivas humanas para resolver problemas de naturaleza variada (Javaid et al., 2021; Tripathi, 2021). La IA es un campo de investigación muy amplio que incluye una gran cantidad de métodos. Estos pueden ofrecer valiosas soluciones computacionales para el procesamiento de grandes volúmenes de datos dando soporte a sistemas de toma de decisiones. Además, pueden procesar el lenguaje natural para extraer información útil de documentos o crear representaciones del conocimiento de un determinado dominio, entre muchas otras aplicaciones (Mo et al., 2023).

Disímiles problemas que surgen en la industria, tales como la toma de decisiones en situaciones complejas o el procesamiento de volúmenes de datos que superan la

capacidad humana, pueden ser abordados con la IA. Esta tecnología les brinda a las compañías un nivel avanzado de análisis y les permite transformar completamente sus sistemas de fabricación al incorporar capacidades de percibir el entorno, ajustarse a necesidades externas y extraer información de los procesos, sirviendo como soporte además a modelos complejos de producción como son la fabricación distribuida o los modelos de servicios extendidos (Wan et al., 2021).

El uso de la IA en la fabricación no es un fenómeno nuevo, sin embargo, desde la introducción de la Industria 4.0, su aplicación a los procesos productivos se ha incrementado exponencialmente. Esto es consecuencia de las capacidades de los más recientes avances en IA y del crecimiento en la disponibilidad de datos debido al uso de sensores inteligentes y dispositivos y máquinas que recolectan continuamente información relacionada con la producción. Sin embargo, a pesar del incremento del uso de la IA en la fabricación, aún hay varios aspectos que deben ser atendidos, lo cual se puede evidenciar en la Figura 2.8, que muestra los resultados de una encuesta a empresas europeas sobre el estado actual de madurez en la adopción de la IA (Bettoni et al., 2021). Mientras continúe avanzando la tecnología de IA y se vuelva más accesible, el número de aplicaciones que serán desarrolladas para resolver problemas complejos en fabricación seguirá creciendo, al igual que el nivel de madurez de adopción de la IA (Alexopoulos et al., 2020).

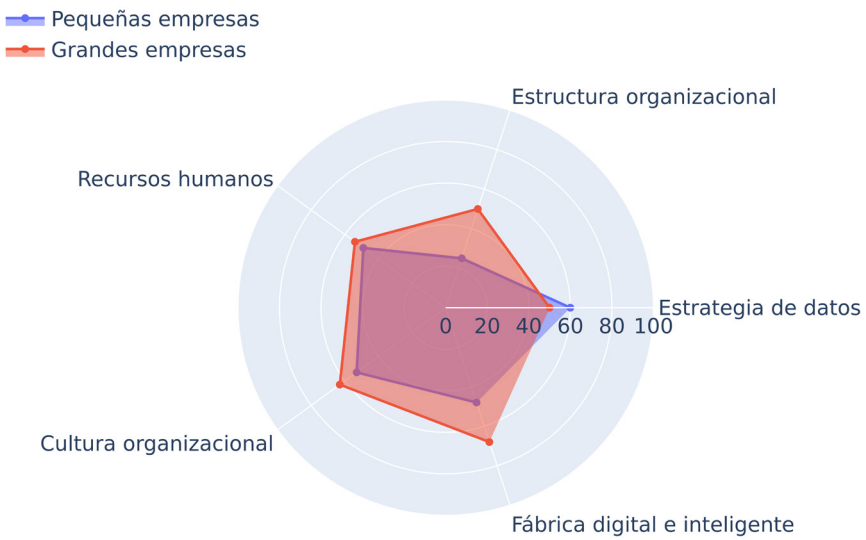

Figura 2.8: Evaluación del nivel de madurez en la adopción de la IA

2.4.1. Aplicaciones de la Inteligencia Artificial en la fabricación

La IA ha tenido un impacto sustancial en las tareas de fabricación que requieren un cierto grado de intelecto. Al recolectar datos continuamente, los sistemas de IA pueden identificar el estado de los productos y equipos para apoyar en los procesos de toma de decisiones con el objetivo de incrementar la eficiencia, detectar defectos, prevenir accidentes o reprogramar la producción (Javaid et al., 2021). Otros estudios han utilizado la IA en la fabricación para incorporar estos algoritmos en sistemas de asistencia y aprendizaje (Fahle et al., 2020). La Tabla 2.1 resume algunas aplicaciones recientes de IA en la industria. Tanto el rango de tareas donde la IA puede aplicarse como el número de métodos que pueden ser aplicados a una tarea particular son considerables. Adicionalmente, hay casos donde los diferentes algoritmos de IA son aplicados en diferentes áreas del mismo proceso productivo, creando entornos complejos e integrales de manufactura inteligente. Por otra parte, varios algoritmos también pueden ser combinados para resolver un único problema, obteniéndose soluciones altamente especializadas.

Tabla 2.1: Aplicaciones de IA en la fabricación

Área	Aplicación	Tarea	Algoritmo	Referencia
Control de calidad	Detección de defectos en cilindros a presión	clasificación de imágenes	CNN	(Cruz et al., 2020)
	Detección de defectos en componentes de turbinas de aviones	segmentación semántica	U-Net	(Cannizzaro et al., 2022)
Mantenimiento predictivo	Detección temprana de daños en dispositivos electromecánicos	estimación de vida útil restante	LSTM + FIS	(Castano et al., 2022)
	Monitorización y predicción de la degradación de equipamiento de corte	monitorización basada en condiciones	SVM	(Acernese et al., 2021)
Planificación de la producción	Ajuste de los requerimientos operacionales en compañías empacadoras dependiendo de la utilización de las máquinas	predicción de la velocidad operacional	CNN + LSTM + AE	(Essien & Giannetti, 2020)

	Programación de la producción en dependencia de las órdenes y el tiempo estimado de finalización	predicción del tiempo de finalización de los productos	RF	(Qiu et al., 2020)
Optimización de procesos	Determinación de los parámetros operacionales óptimos en la producción de láminas de acero de alta resistencia	mejora de las propiedades del producto	*k*NN + GA	(Jha et al., 2023)
	Selección de parámetros adecuados de impresión para manufactura aditiva	predicción del enfriamiento	GPR	(Osswald et al., 2020)
Robótica colaborativa	Automatización de la colaboración humano-robot en líneas de ensamblaje de vehículos	control de robots	CSKR	(Conti et al., 2022)

A pesar de que varias técnicas han sido utilizadas en un amplio rango de aplicaciones de fabricación, aún quedan muchos desafíos y cuestiones por responder. Como consecuencia, los investigadores continúan trabajando en temas como el razonamiento basado en datos para desarrollar inteligencia aplicada, computación periférica, ciberseguridad en entornos industriales, etc. (Rai et al., 2021). Debe hacerse un gran esfuerzo para crear sistemas de IA que tengan en cuenta todo el proceso de producción, ya que la Industria 4.0 requiere colaboración en todas las fases productivas. Actualmente, el control de dispositivos, el manejo de recursos y la gestión de la cadena de suministros son las principales áreas de aplicación de la IA en la fabricación. Por lo tanto, la creación de sistemas que combinen aplicaciones en estas áreas permitirá mejorar la utilización del inventario, el seguimiento de los activos y la explotación de recursos (Javaid et al., 2021).

2.4.2. Aprendizaje automático en la fabricación

Las técnicas de IA, tales como el aprendizaje automático, permiten la generación de inteligencia aplicable mediante el tratamiento de los datos recopilados con el fin de impulsar la productividad sin necesidad de modificar drásticamente los recursos

requeridos. Esto se logra con posterioridad al proceso de entrenamiento donde una gran cantidad de datos son utilizados por los algoritmos para adaptar sus parámetros de aprendizaje a los patrones subyacentes en los datos. Mientras más datos sean utilizados para entrenar a los algoritmos, estos son capaces de aprender relaciones más complejas y se pueden realizar mejores predicciones para beneficio de la empresa. Los algoritmos de aprendizaje automático tienen la capacidad de identificar patrones complejos de fabricación, lo que posibilita el desarrollo de sistemas de apoyo a la toma de decisiones para un gran número de tareas en la manufactura como la optimización de procesos, mantenimiento predictivo, planificación de la producción y control de calidad, entre otros (Rai et al., 2021).

Entre las diferentes ramas de IA que se aplican en la fabricación, el aprendizaje automático es la que mayor crecimiento ha experimentado en los últimos años (Sarker, 2021). Se pueden encontrar aplicaciones de aprendizaje automático en cada una de las fases del ciclo de vida de un producto, comenzando desde el diseño hasta su reciclaje (Rai et al., 2021). Generalmente, en entornos de fabricación con gran disponibilidad de datos, se utiliza el aprendizaje supervisado, es decir, algoritmos a los cuales para entrenarlos se les debe especificar cuáles son los valores de las variables de entrada y de salida (Dogan & Birant, 2021).

2.4.2.1. Aprendizaje automático automatizado

A pesar del incremento que ha tenido el uso del aprendizaje automático en la industria en los últimos años, generalmente el proceso de desarrollo de soluciones que utilizan esta rama de la IA aún se realiza de forma tradicional. Esto produce dos grandes limitaciones: (1) la dependencia de expertos en la materia para lograr obtener modelos con un alto nivel de fiabilidad y robustez y (2) la prolongación de los tiempos de desarrollo, ya que normalmente se realizan varias iteraciones de un procedimiento compuesto por varios pasos, hasta lograr el resultado deseado (K. Chauhan et al., 2020).

Recientemente, para intentar solucionar estas desventajas se ha propuesto el uso de metodologías de aprendizaje automático automatizado. Estas metodologías buscan automatizar el proceso de desarrollo de una solución de aprendizaje automático, atendiendo a que generalmente este proceso se compone de un conjunto de operaciones preestablecidas como son el preprocesamiento de los datos, el análisis sistémico y de variables, la selección y el entrenamiento de modelos, y la optimización de los hiperparámetros de estos, como se muestra en la Figura 2.9.

El aprendizaje automático automatizado permite ahorrar tiempo y esfuerzo en la realización de estas tareas (Truong et al., 2019).

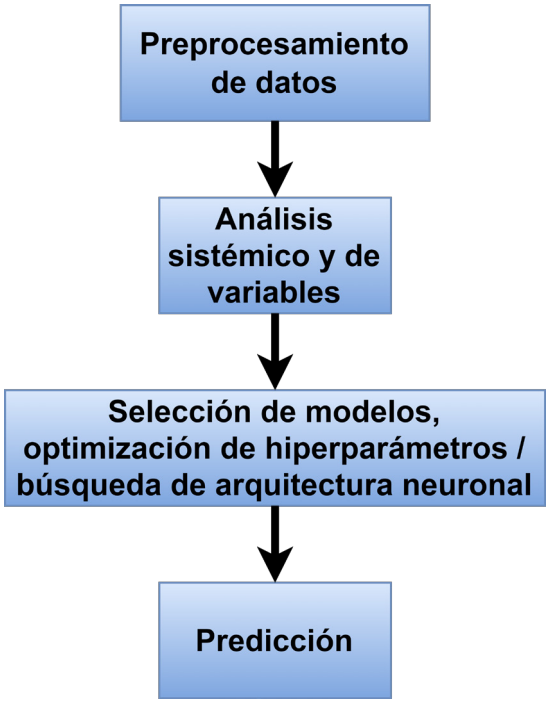

Figura 2.9: Pasos típicos para desarrollar una solución de aprendizaje automático

2.4.2.2. Clasificación de los sistemas de aprendizaje automático

En dependencia del nivel de automatización en el desarrollo e implementación de un sistema de aprendizaje automático, Google ha propuesto su clasificación en tres niveles. Estos son los llamados niveles de operaciones de aprendizaje automático (MLOps): Nivel 0, Nivel 1 y Nivel 2 (Garg et al., 2021; Google, 2023). Los sistemas considerados de Nivel 0 son aquellos donde todo el flujo de trabajo se realiza manualmente mediante la ejecución de secuencias de comandos para llevar a cabo cada una de las tareas necesarias en la generación de un modelo. La Figura 2.10 muestra el flujo de trabajo en estos sistemas. Al ejecutarse uno a uno los pasos, se va realizando un análisis interactivo de los resultados obtenidos por el modelo y, en caso de no ser satisfactorios se puede volver a iterar para generar otro modelo. Una vez que se ha alcanzado el resultado deseado se procede a desplegar el modelo, también de manera manual. Debido a la forma en que se trabaja a este nivel, los modelos son actualizados con muy baja frecuencia, por lo que generalmente no son capaces de adaptarse a cambios dinámicos en el proceso que se esté modelando.

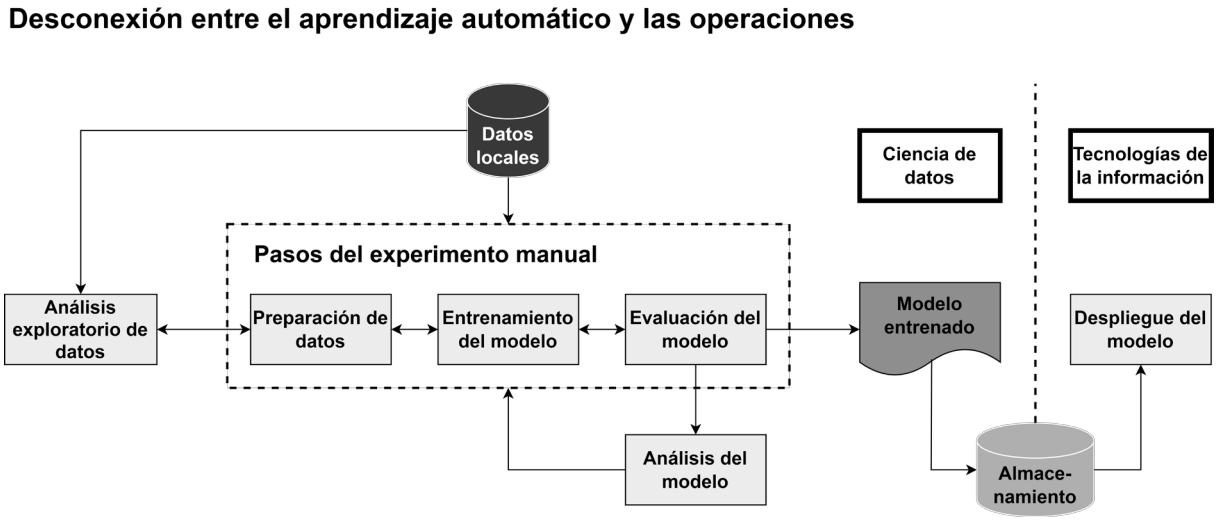

Figura 2.10: Nivel 0 de MLOps (proceso manual)

En el caso de los sistemas de Nivel 1, se automatiza todo el proceso de creación del modelo en lo que se conoce como aprendizaje automático automatizado[8]. Es decir, estos sistemas son capaces de recibir los datos y devolver un modelo entrenado, tal como se muestra en la Figura 2.11. De esta manera se logra obtener un flujo de trabajo reproducible, en el cual las iteraciones de los experimentos se realizan de manera rápida y sin necesidad de la ejecución y evaluación manual en cada uno de sus pasos. Este tipo de sistemas idealmente se encuentran encapsulados en contenedores, siendo independientes del entorno de ejecución. Otra de las ventajas que introducen los sistemas de Nivel 1 es que, si se adquieren más datos sobre el proceso, los modelos pueden ser reentrenados de manera rápida e incluso puede llegar a programarse de manera sistemática cuando debe ejecutarse este reentrenamiento. Aunque reentrenar el modelo de manera sistemática es de gran ayuda para lidiar con los cambios dinámicos de los procesos, puede ocurrir que la frecuencia de actualización no esté acorde con la velocidad en la que cambia la tendencia en los datos. Para hacer frente a estas situaciones donde el desempeño del modelo puede sufrir deterioro o degradación, se requiere realizar una monitorización activa que indique cuando es necesario reentrenar el modelo para que sea capaz de adaptarse a los nuevos patrones en los datos.

[8] En la clasificación de los sistemas de aprendizaje automático propuesta por Google se utiliza el término experimento orquestado. En esta tesis se ha utilizado indistintamente dicho término y el término aprendizaje automático automatizado.

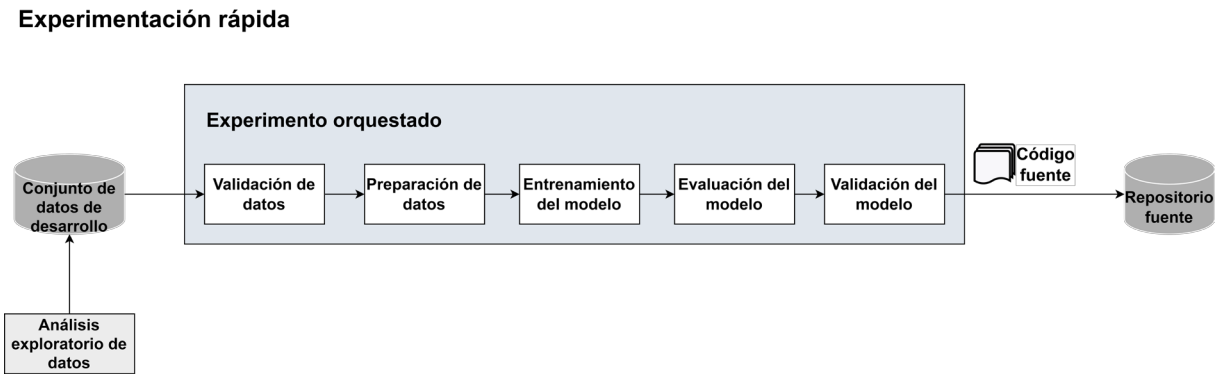

Figura 2.11: Nivel 1 de MLOps (automatización del flujo de trabajo de aprendizaje automático)

Teniendo en cuenta esta necesidad de tener modelos que se actualicen a medida que se requiera, se ha propuesto el uso de sistemas de Nivel 2. En estos, se automatiza tanto el proceso de creación del modelo, como los elementos necesarios para evaluar su desempeño y ejecutar el reentrenamiento, tal como se muestra en la arquitectura de ejemplo de la Figura 2.12. Además, estos sistemas son capaces de desplegar de forma automática los modelos para comenzar a utilizarlos sin necesidad de intervención humana. Idealmente, estos deben implementar los conceptos de integración continua/entrega continua, de manera que cuando se actualice un componente del sistema, todos los cambios necesarios se realicen sin necesidad de intervención externa. Generalmente, debido a la alta complejidad de estos sistemas, solo se implementan en aplicaciones críticas y basándose en arquitecturas de computación en la nube compuestas por múltiples servicios. En el estado actual, muchos de los conceptos sobre estos sistemas son aún especulativos, ya que no existen herramientas para llevarlos a la práctica de manera eficiente, sobre todo en los casos donde no hay manera de comparar los valores reales con la predicción del modelo para realizar la evaluación del desempeño de este último. Por otra parte, las herramientas que sí se utilizan aún están en estado temprano de desarrollo por lo que sufren frecuentes cambios y, si son actualizadas, pueden provocar inestabilidad en los sistemas existentes (Symeonidis et al., 2022).

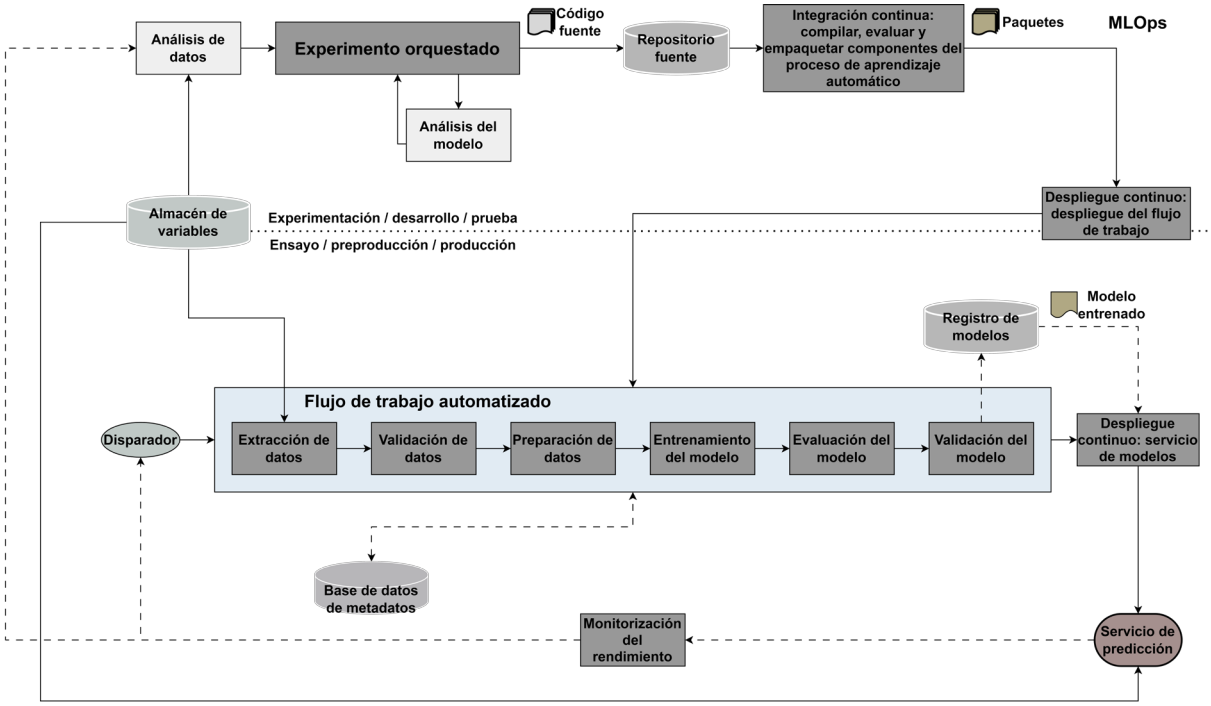

Figura 2.12: Ejemplo de Nivel 2 de MLOps (automatización del flujo de trabajo de integración continua / despliegue continuo)

2.4.2.3. Panorama actual del desarrollo de soluciones de aprendizaje automático

Actualmente, la mayoría de los sistemas de aprendizaje automático, tanto en la academia como en la industria, aún se desarrollan con Nivel 0 de automatización, es decir, manualmente. Debido a las limitaciones de este enfoque, se han comenzado a desarrollar productos de Nivel 1, como es el caso de herramientas comerciales como Google Cloud AutoML, Microsoft Azure Machine Learning, H2O Driverless AI, entre otras, las cuales son capaces de generar modelos de IA utilizando aprendizaje automático automatizado. Sin embargo, estas herramientas funcionan como una caja negra, ya que las compañías que las desarrollaron intentan proteger su tecnología de cara a la competencia. Esta falta de transparencia imposibilita conocer cómo se llevan a cabo los procesos internos. Por otra parte, también existen soluciones de código abierto como auto-sklearn, TPOT o Auto-Keras, las cuales están ganando cada vez más popularidad. No obstante, todas estas herramientas se enfocan en resolver el problema combinado de selección de algoritmos y optimización de hiperparámetros. En muchas aplicaciones industriales esto solo es una parte del problema, ya que después de obtener el modelo se requiere integrarlo en un algoritmo de optimización que permita mejorar el desempeño del proceso productivo (Cruz et al., 2023).

En el caso de las soluciones de Nivel 2, según el conocimiento del autor, no existe un único producto que proporcione todas las funcionalidades necesarias. Hasta ahora, el enfoque adoptado consiste en desarrollar arquitecturas que integren un conjunto avanzado de herramientas, las cuales se encargan de automatizar cada una de las tareas requeridas (Tamburri, 2020). Esto, aunque permite obtener como resultado final un sistema funcional completamente automatizado, presenta el inconveniente de que para su creación se necesita de personal altamente cualificado para llevar a cabo todas las configuraciones necesarias. Además, debido a la complejidad de dichas arquitecturas, estas pueden no ser idóneas para las empresas con bajo nivel de madurez tecnológica, siendo más adecuadas para empresas que basan su modelo de negocio en torno al análisis y procesamiento intensivo de datos, en lugar de a la fabricación (Mäkinen et al., 2021).

2.5. Gemelos digitales e Inteligencia Artificial en la reconfiguración y supervisión. Contexto actual

Gran parte de los trabajos que actualmente se llevan a cabo sobre el desarrollo de gemelos digitales para la industria buscan introducir el uso de la IA, mayoritariamente el aprendizaje automático. Sin embargo, son pocos los trabajos que abordan este tema desde una perspectiva general y automatizable en lugar de diseñar soluciones específicas para cada caso analizado.

Por ejemplo, se ha considerado la generación de modelos de simulación que sirvan como base para la creación de gemelos digitales utilizando técnicas de aprendizaje automático y minería de datos de producción (Friederich et al., 2022). Pero en este caso se requiere de intervención humana durante algunos puntos críticos de la creación de los modelos de simulación como se puede ver en la Figura 2.13, por lo que no es un proceso completamente automatizado. Además, este trabajo solo se desarrolla hasta el punto de replicar el comportamiento de la línea productiva, mientras que la posibilidad de optimizar la producción solo es tratada de manera hipotética.

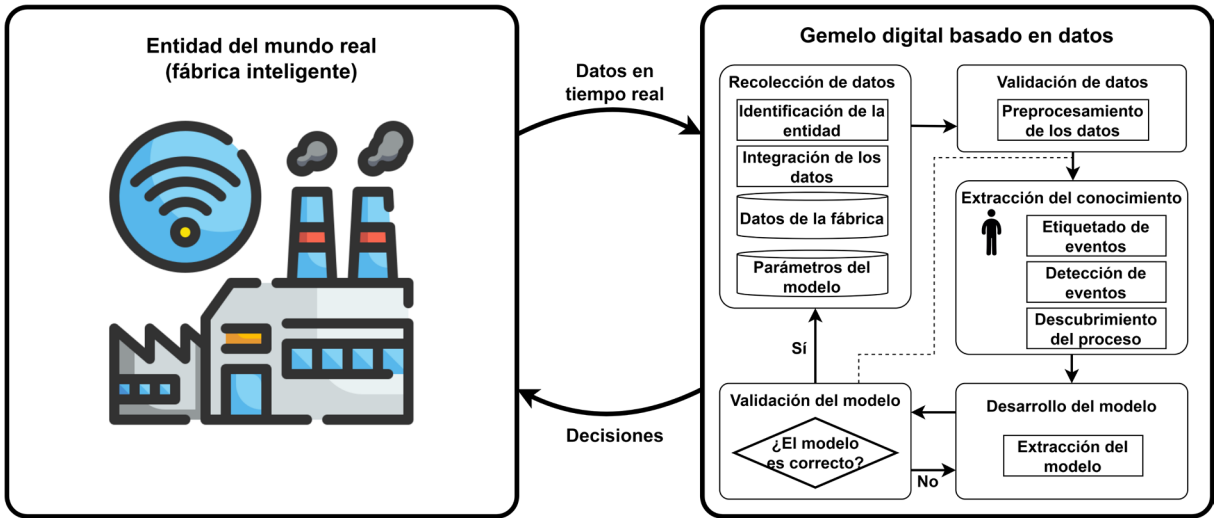

Figura 2.13: Marco de desarrollo de gemelos digitales utilizando datos de producción

Igualmente se han propuesto soluciones que utilizan directamente algunos enfoques de aprendizaje automático automatizado para crear los servicios inteligentes de gemelos digitales (Yu et al., 2023), pero solo durante la fase de implementación del sistema. Es decir, que no se realiza una monitorización del rendimiento de los modelos creados para determinar si se deben reentrenar, ni se tiene en cuenta la posibilidad de automatizar el proceso de reentrenamiento.

También se ha presentado un marco formal para la creación de gemelos digitales que sean capaces de optimizar el proceso productivo que representan y puedan actualizarse continuamente (Min et al., 2019). Por el momento, este marco formal carece de validación práctica y en él no se precisan aspectos claves sobre cómo llevar a cabo la actualización de los modelos que conforman el gemelo digital de manera automatizada. Además, solo se contempla la optimización monoobjetivo del proceso considerando únicamente el beneficio económico, lo que limita su aplicación en casos que se requiera optimizar varios objetivos.

Por otra parte, se han ideado procedimientos para la generación automática de modelos de sistemas productivos utilizando directamente registros de eventos (Lugaresi & Matta, 2021). Este procedimiento presenta una solución efectiva para hacer frente al problema de la generación de modelos extremadamente complejos que terminan siendo de poca utilidad. Sin embargo, está orientado a modelos que representen la estructura lógica o el flujo de trabajo en la producción sin profundizar en las capacidades de reparametrización. Además, para poder implementar dicho procedimiento se requiere inicialmente contar con un Sistema

de Ejecución de Fabricación (MES), lo cual no se corresponde con la situación de la mayoría de las empresas con nivel bajo o medio de madurez tecnológica.

Recientemente también se ha propuesto el método de detección temprana de fallas mostrado en la Figura 2.14 (Lv & Xie, 2022). En este método se tiene en cuenta la monitorización del rendimiento del modelo para determinar cuándo este debe ser reentrenado, o incluso si se debe modificar su estructura o realizar una nueva selección de variables. Sin embargo, en este trabajo solo se hace una descripción teórica del sistema propuesto, no se profundiza en la forma de implementar los distintos elementos de dicho sistema y no se considera cómo optimizar el proceso productivo para prevenir o disminuir las fallas.

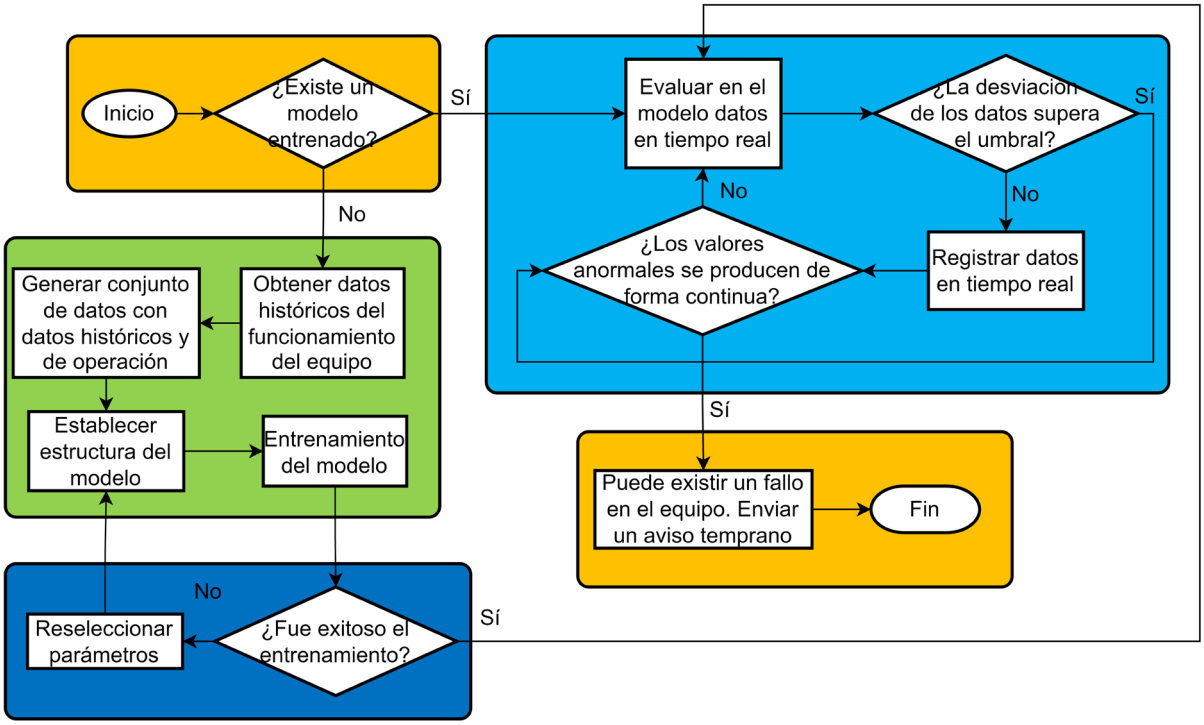

Figura 2.14: Método de detección temprana de fallas

Otros trabajos se han centrado en la introducción de nuevas técnicas de modelado y optimización bayesiana de hiperparámetros en las metodologías de aprendizaje automático automatizado para la gestión de grandes volúmenes de datos (Karras et al., 2023), pero en este caso tampoco se considera como utilizar el modelo generado para llevar a cabo una optimización del proceso de donde provienen los datos ni cómo hacer frente a la variabilidad de estos.

Por otra parte, también se ha trabajado en el desarrollo de sistemas que permiten llevar a cabo la optimización de los procesos productivos de manera automática atendiendo a la variabilidad de los datos (De Blasi et al., 2023). A pesar de que en este caso se describe como realizar un reajuste automático del modelo frente a cambios en los datos, existen varias limitaciones. Por ejemplo, no se realiza la selección de variables, modelos, ni la optimización de hiperparámetros. Es decir, solo se realiza un reentrenamiento del mismo modelo con datos actualizados, pasando por alto pasos en el proceso de desarrollo de una solución de aprendizaje automático que contribuyen a aumentar el rendimiento y la fiabilidad de los modelos.

Tras analizar los diferentes métodos y enfoques encontrados en la literatura relacionada con el desarrollo de componentes de IA para gemelos digitales se ha detectado que ninguno de ellos cumple simultáneamente con las siguientes condiciones:

- Ser generalizables a un amplio rango de procesos productivos.
- Utilizar los modelos para, además de representar, también optimizar el proceso productivo.
- Ser capaces de ejecutar todo el flujo de trabajo para crear la capa de inteligencia de forma completamente automatizada.
- No requerir de infraestructura o sistemas específicos que generalmente no existen en las empresas con nivel bajo o medio de madurez tecnológica.
- Ser capaces de monitorizar su rendimiento y actualizar los modelos automáticamente, en caso de ser necesario.
- No requerir conocimientos profundos sobre IA para poder utilizarlos.

Debido a estas limitaciones relacionadas con la creación de la capa de servicios inteligentes, actualmente, el cuello de botella en el proceso de desarrollo de los gemelos digitales se encuentra en estos componentes y, además, se requiere de personal altamente especializado para poder implementar la solución. Por otra parte, el hecho de no tener en cuenta la necesidad de reentrenar o reajustar los modelos en los procesos donde la variabilidad de los datos es un fenómeno frecuente, puede conducir a experiencias negativas en el uso de estos sistemas.

2.6. Conclusiones

En este capítulo se revisaron varios aspectos de la Industria 4.0 necesarios para comprender el contexto en el que se desarrolla la tesis, abarcando desde los sistemas ciberfísicos industriales hasta la utilización del aprendizaje automático en la manufactura. Se describieron varias arquitecturas de referencia para la implementación del paradigma de la Industria 4.0 y se discutieron sus limitaciones de cara a una adopción efectiva del mismo, destacándose la poca profundización en los aspectos prácticos. También se introdujo el concepto de gemelo digital y se vio como puede ser descrito mediante un conjunto de 4 elementos, destacándose la importancia de la capa de inteligencia para lograr una representación efectiva de los escenarios de manufactura. Se declararon los niveles de diseño de los gemelos digitales, determinándose el nivel de proceso como el más adecuado en el contexto de esta tesis. Además, en este capítulo se analizó el uso de la IA en la fabricación, específicamente las técnicas de aprendizaje automático. Se revisaron soluciones que permiten la automatización del flujo de trabajo a la hora de emplear estas técnicas y se vieron ejemplos concretos aplicados a la industria. Sin embargo, en los trabajos revisados se detectaron ciertas limitaciones como la falta de integración de las etapas de modelado de los procesos productivos con la optimización de estos, no tomar en cuenta aspectos prácticos como la monitorización del rendimiento de los modelos para su reentrenamiento oportuno, o la omisión de operaciones clave en el proceso de modelado.

3. Arquitectura

3.1. Introducción

Para impulsar la adopción de la Industria 4.0 a gran escala se requiere la existencia de infraestructuras tecnológicas sólidas que faciliten el desarrollo de soluciones basadas en las nuevas tecnologías dirigidas a los diferentes actores dentro del ámbito industrial, adaptándose a sus necesidades. En este capítulo se describe la arquitectura KITT4SME, la cual involucra una serie de módulos que permitirán a las empresas contar con servicios enfocados a la digitalización de la fabricación. Además, se propone el diseño y desarrollo de dos módulos basados en IA para esta plataforma, así como su aplicación en soluciones específicas para una línea piloto de Industria 4.0, incluyendo el desarrollo de un gemelo digital y la preservación de la capacidad productiva de las máquinas/robots y dispositivos de fabricación.

3.2. Descripción de la arquitectura KITT4SME

En el capítulo anterior se presentaron varias arquitecturas de referencia y se pudo ver que su principal inconveniente consistía en que se limitaban a propiciar un marco teórico para la Industria 4.0, pero no profundizaban en los aspectos prácticos de su implementación. Esto trae como resultado que las empresas que no estén preparadas adecuadamente para adoptar este nuevo paradigma pueden quedar expuestas a soluciones digitales extremadamente complejas y con un alcance mucho más amplio de lo que realmente necesitan, entorpeciendo el proceso de transición digital. Con el objetivo de facilitar la adopción de soluciones digitales, especialmente las basadas en IA, el proyecto KITT4SME está desarrollando una plataforma de paquetes digitales personalizables en dependencia de las necesidades e intereses de las empresas (Gladysz et al., 2023).

La plataforma KITT4SME pretende ofrecer un grupo de servicios inteligentes en una arquitectura en la nube, los cuales puedan ser consumidos individual o conjuntamente para digitalizar procesos productivos. Estos servicios deben permitir monitorizar los procesos productivos, diagnosticarlos, generar información útil para mejorar su gestión, y actuar sobre ellos en caso de ser

necesario, entre otras funcionalidades. Como puede verse en la Figura 3.1, la plataforma KITT4SME se basa en FIWARE. FIWARE proporciona una infraestructura modular de código abierto para la Internet de las Cosas (IoT), la cual habilita la integración de sistemas de información y gestión a nivel de fábrica, sensores IoT, *wearables*, máquinas herramientas, robots y dispositivos portátiles, entre otras fuentes de datos. Al tener una capa intermedia con protocolos de comunicación estandarizados se pueden desarrollar módulos de alto nivel para ofrecer servicios inteligentes basados en datos. Una vez que estos servicios hayan sido creados y estén disponibles, las empresas pueden enfocarse en la interconexión de sus procesos productivos con la plataforma y en la interoperabilidad con los servicios, sin tener que preocuparse por el resto de aspectos técnicos de la solución. Así es posible acortar los tiempos de desarrollo para la adopción de la Industria 4.0 y se puede poner la IA a disposición de las empresas con menor nivel de madurez tecnológica y presupuesto limitado.

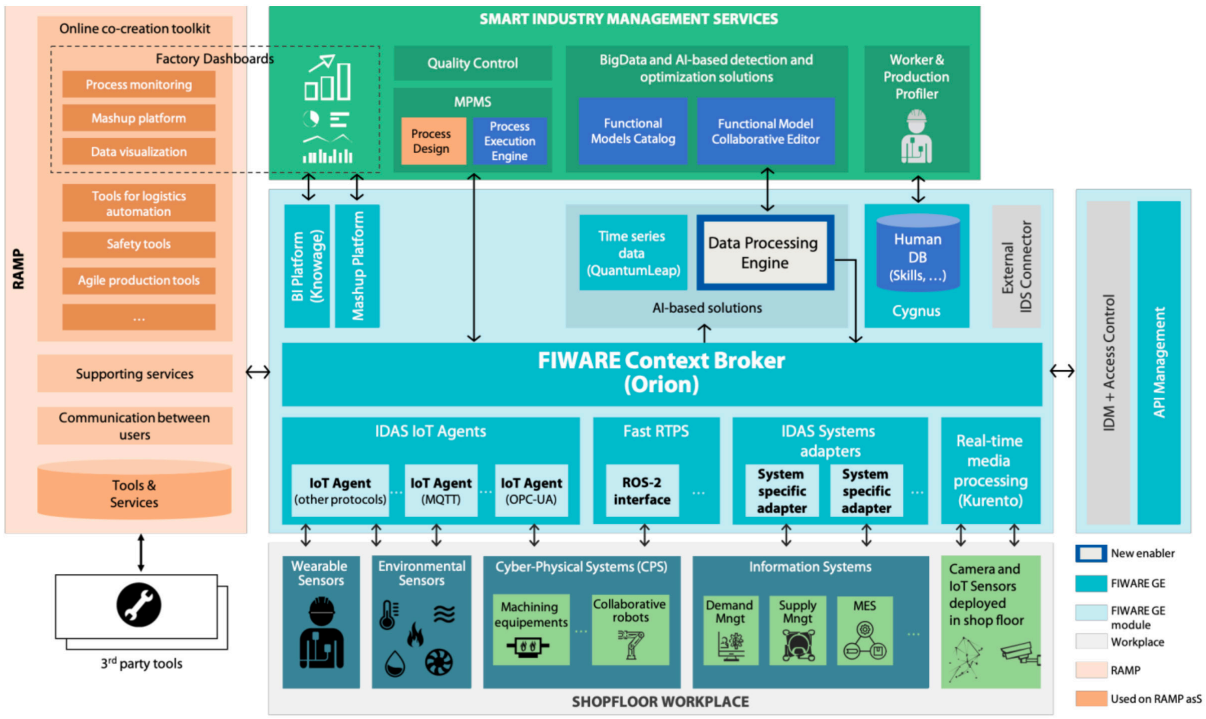

Figura 3.1: Arquitectura propuesta por KITT4SME

En un análisis más detallado, se puede ver que la capa inferior de esta arquitectura es la encargada de la recolección de datos de los sistemas y dispositivos en la fábrica, los cuales alimentan la base de conocimientos del sistema y, además,

reciben las recomendaciones o ejecutan las órdenes provenientes de niveles superiores. En esta capa se encuentran los sistemas ciberfísicos, los sistemas de información y diferentes tipos de sensores.

En la segunda capa, FIWARE provee un conjunto de habilitadores genéricos los cuales sirven de interfaces para las diferentes fuentes de información contempladas en la arquitectura. Estos habilitadores se utilizan para facilitar la integración de los componentes de la capa inferior que no se comuniquen directamente siguiendo la especificación *Next Generation Service Interfaces v2* (NGSI-v2). Opcionalmente, se puede prescindir de utilizar los habilitadores si los componentes de la capa inferior soportan comunicaciones basadas directamente en esta especificación. Entre estos habilitadores que provee FIWARE están:

- IDAS, el cual ofrece una amplia gama de agentes destinados a la interconexión de dispositivos mediante los protocolos IoT más utilizados (LwM2M sobre CoAP, JSON y Ultralight2.0 sobre HTTP/MQTT/AMQP, oneM2M u OPC-UA).
- Fast RTPS, el cual adopta el *middleware* ROS2 para interactuar con sistemas robóticos.
- Kurento, el cual permite el procesamiento multimedia en tiempo real, de manera que las cámaras de vídeo se puedan tratar como sensores, así como la incorporación de funciones de aplicación avanzadas (comunicaciones audiovisuales integradas, realidad aumentada, etc.).

La siguiente capa contiene el componente central y obligatorio de FIWARE, el bróker de contexto Orion. Este permite la gestión de la información de forma descentralizada y a gran escala. Orion proporciona una Interfaz de Programación de Aplicaciones de Transferencia de Estado Representacional (REST API) que permite realizar actualizaciones, consultas o suscripciones a la información de los dispositivos, componentes y procesos de una fábrica.

La próxima capa contiene componentes propios de FIWARE, los cuales utilizan Orion como base para las comunicaciones internas. A pesar de ser parte del núcleo de FIWARE, estos componentes no son de uso obligatorio. Entre las funcionalidades que se ofrecen en esta capa se encuentran las relacionadas con la persistencia de datos, la verificación de transacciones, la detección de intrusos y el uso de motores de procesamiento de datos. Aunque FIWARE proporciona sus

habilitadores genéricos propios para la persistencia de datos, también admite la utilización de fuentes o bases de datos de terceros.

Por último, la arquitectura KITT4SME propone una capa superior, llamada Servicios de Gestión de Industria Inteligente, donde se alojen servicios propios. Esta capa debe incluir componentes de *Big Data* e IA que procesarán los datos para detectar patrones y correlaciones, anomalías, fallos de funcionamiento, comportamientos inesperados o valores atípicos. Además, algunos de estos módulos deberán proporcionar funcionalidades de valor añadido como la optimización de procesos o mejoras de la calidad. Los resultados de estos módulos podrán ser inyectados en Orion para enviarlos al destinatario correspondiente o desencadenar mecanismos de actuación. Adicionalmente, la arquitectura KITT4SME contempla la integración con herramientas de control de acceso y con el mercado *Robotics and Automation Market Place* (RAMP), el cual permite ofrecer los diferentes componentes de la arquitectura como servicio. Así, se facilita la creación de una solución personalizada en línea donde cada empresa decide que componentes utilizar.

3.3. Métodos de reconfiguración y supervisión inteligente y automática en KITT4SME. Propuesta

Entre los servicios de industria inteligente que debe ofrecer la plataforma KITT4SME se encuentra la optimización de los procesos productivos. En este caso, la optimización se aborda desde el punto de vista de la reparametrización de los activos del proceso productivo. Para poder llevar a cabo la reparametrización óptima de un proceso productivo mediante un enfoque guiado por datos es necesario, primeramente, generar modelos de cada indicador de la producción que se quiera optimizar y, a continuación, haciendo uso de la capacidad de inferencia de los modelos creados, ejecutar la estrategia de optimización en sí misma. Por tanto, la primera propuesta de este trabajo consiste en el desarrollo de un módulo para la arquitectura KITT4SME que permita ofrecer el servicio de reparametrización basada en IA de procesos productivos (*Insight Generator*). En resumen, este módulo funcionaría como un sistema de aprendizaje automático de Nivel 1, donde a partir de los datos de producción se realiza un experimento orquestado para crear los modelos, con la particularidad de que en este caso también dentro del experimento orquestado se lleva a cabo la optimización del proceso productivo.

Además del interés evidente que implica contar con un módulo que ofrezca estas prestaciones para garantizar un correcto aprovechamiento de las capacidades productivas de la fábrica, también se deben considerar las ventajas que ofrece este para el desarrollo de elementos más avanzados de la Industria 4.0, como son los gemelos digitales. Los gemelos digitales se basan en modelos que representan el comportamiento y las características del proceso analizado. En este caso, se puede acelerar el proceso de creación de gemelos digitales ya que los modelos se generarían de manera automática en la plataforma KITT4SME, que funcionaría como la capa de inteligencia de un gemelo digital de cuatro elementos. Con la finalidad de demostrar las capacidades del módulo *Insight Generator* para acelerar el proceso de digitalización de una empresa, la segunda propuesta del trabajo consiste en desarrollar un gemelo digital de un proceso productivo que consuma los servicios de dicho módulo.

Debido a que los modelos creados por *Insight Generator* son mantenidos en la plataforma desde la cual se ofrece el servicio de predicción/inferencia, se reduce la complejidad en el proceso de desarrollo del gemelo digital, a la vez que se minimiza el consumo de recursos computacionales propios y se simplifica el mantenimiento. Así, se puede centrar el trabajo en la interacción con la plataforma y en otros aspectos de los gemelos digitales que no son abordados por la arquitectura KITT4SME. Entre los aspectos que se pretenden abordar con el desarrollo del gemelo digital se contemplan tres funcionalidades: gestión, visualización y supervisión. La funcionalidad de gestión está orientada a interactuar con *Insight Generator* para enviar los datos y la configuración necesarios para entrenar por primera vez los modelos, consumir el servicio de predicción de los modelos y ejecutar el proceso de optimización cada vez que ocurra un cambio en las condiciones de producción. Además, este servicio se encargará de la difusión de los resultados tanto de la predicción como de la optimización ejecutados en *Insight Generator*. Por su parte, la funcionalidad de visualización se basará en una réplica virtual tridimensional del escenario de producción que permitirá examinar el comportamiento del sistema bajo el régimen actual de trabajo o en simulación, utilizando información del servicio de gestión. Por último, la funcionalidad de supervisión permitirá ofrecer capacidades básicas de un sistema de Nivel 2 al comparar las predicciones de los modelos con los valores reales de los indicadores medidos en la fábrica y, si se detecta deterioro en el desempeño de los modelos, indicar que estos se deben entrenar nuevamente con datos actualizados.

Por otra parte, la arquitectura KITT4SME contempla ofrecer un servicio que permita la reconfiguración inteligente y automática de máquinas/robots y dispositivos de un sistema de producción para mantener la productividad, evitando desviaciones indeseadas. Dicho servicio aseguraría que los equipos mantengan un rendimiento cercano al esperado, evitando de esta forma que ocurran potenciales retrasos en la producción, subutilización de los recursos e ineficiencia si disminuye la productividad de algún equipo o, por el contrario, que se generen potenciales cuellos de botella, agotamiento de los recursos y disminución de la calidad si se incrementa la productividad de un equipo por encima del nivel esperado. Por tanto, la tercera propuesta de este trabajo consiste en desarrollar un módulo para la arquitectura KITT4SME que permita determinar las acciones correctivas para contrarrestar las desviaciones en el rendimiento de los equipos productivos (*Reco*). Para demostrar las funcionalidades de *Reco* se desarrollará un componente que facilite su integración con un proceso productivo.

La Figura 3.2 muestra una representación conceptual de un sistema que contempla las propuestas hechas anteriormente. Además, una vez que se hayan desarrollado todos los elementos necesarios para el sistema presentado, este trabajo propone su despliegue y evaluación en la línea piloto del grupo GAMHE. De esta forma se podrán evaluar los aportes para la digitalización de un proceso productivo.

3.4. Conclusiones

En este capítulo primeramente se presentó la plataforma KITT4SME la cual se basa en FIWARE y ha sido diseñada para superar las limitaciones de otras arquitecturas enfocadas a la Industria 4.0 introducidas previamente. Esta plataforma busca proporcionar un conjunto de servicios para facilitar la transformación digital en la industria, entre los que se encuentran módulos para la optimización de los procesos productivos y la corrección de desviaciones en la productividad. Partiendo de la necesidad de crear estos módulos, se propuso su desarrollo e integración en la plataforma. Además, se planteó validar los módulos desarrollados mediante su aplicación en una línea piloto de Industria 4.0. De acuerdo a la propuesta, el módulo de optimización de procesos productivos deberá actuar como capa de inteligencia de un gemelo digital, el cual también deberá desarrollarse.

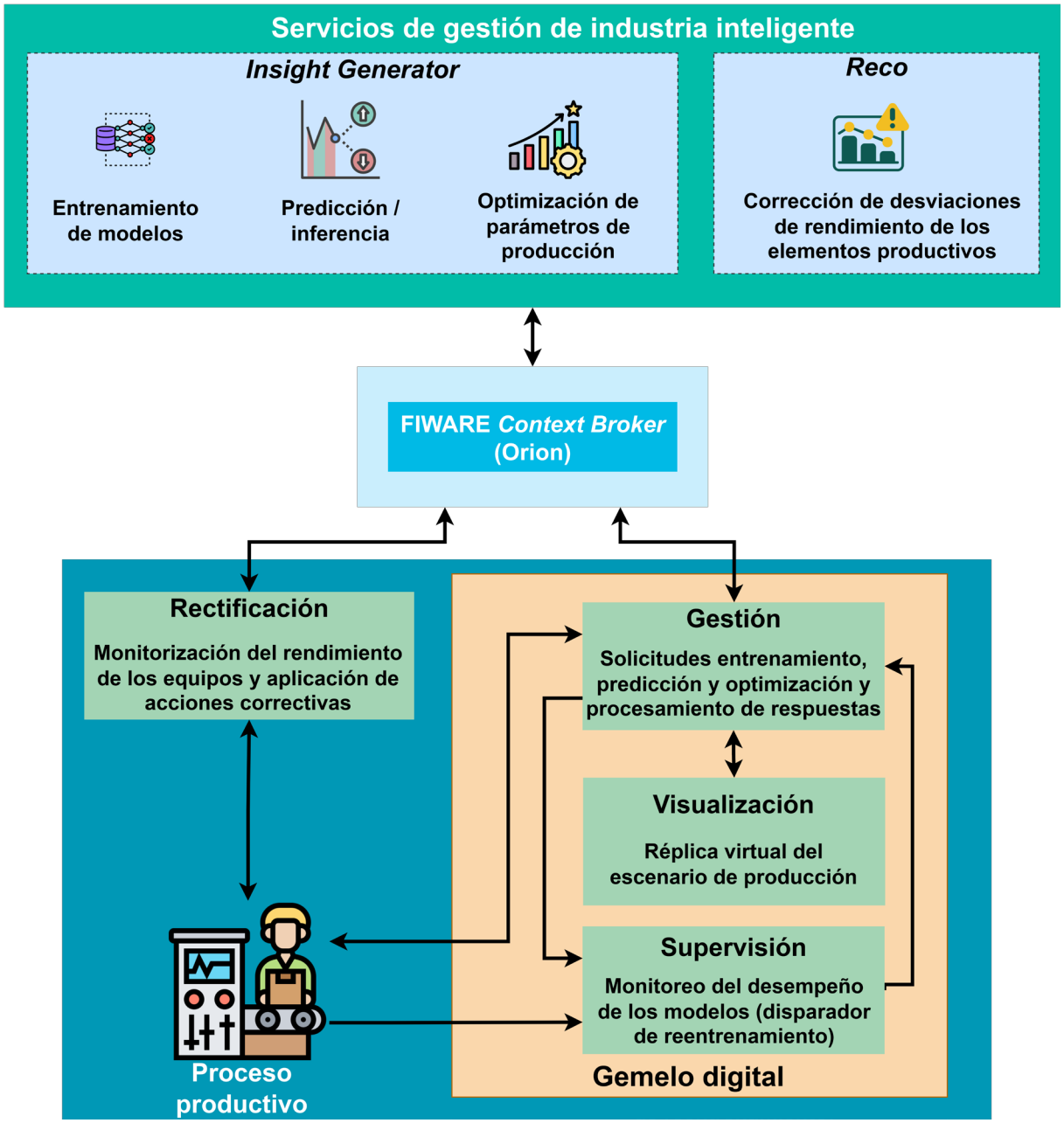

Figura 3.2: Representación conceptual del sistema propuesto

4. Metodología para la reparametrización automática y óptima

4.1. Introducción

En el capítulo anterior se propuso la creación de dos módulos de IA para ser integrados en la capa de Servicios de Gestión de Industria Inteligente de la arquitectura KITT4SME. En este capítulo se aborda el primero de ellos, el cual se enfoca a la reparametrización automática y óptima de la producción a partir de datos históricos con el fin de generar recomendaciones inteligentes: *Insight Generator*. Este módulo se compone de tres servicios: el entrenamiento/reentrenamiento de los modelos, la predicción/inferencia y la optimización de los parámetros de producción. Cada uno de estos servicios se puede utilizar de forma individual o también se pueden usar de manera conjunta para articular un experimento orquestado que sea capaz de modelar un proceso productivo a partir de datos recopilados y determinar regímenes de trabajo óptimos. Una de las particularidades del módulo es que es independiente de la aplicación, de manera tal que pueda ser utilizado en un amplio rango de procesos productivos. En el capítulo se presentan los diferentes algoritmos y técnicas utilizadas en cada una de las etapas de la metodología propuesta para desarrollar *Insight Generator*. Además, se describe como se utilizará este módulo como soporte para la creación del gemelo digital de una línea piloto y se profundiza en las diferentes funcionalidades del gemelo digital. Adicionalmente, se presenta el conjunto de herramientas que se utilizarán para la implementación de los diferentes servicios y funcionalidades descritas.

4.2. Metodología

Con el auge de la personalización masiva, en muchos procesos productivos se deben procesar pequeños lotes de un producto con determinadas características y, seguidamente, procesar otros lotes de productos con características diferentes. Esto es posible gracias a los sistemas de fabricación reconfigurables, los cuales pueden cambiar las operaciones que se realizan en una línea productiva, variar el orden de dichas operaciones o modificar la capacidad productiva para satisfacer los

cambios en la demanda (Mehrabi et al., 2002). Sin embargo, estos constantes cambios de régimen de trabajo propician que el sistema productivo no opere de manera óptima debido a la dificultad que supone hallar los parámetros adecuados para cada uno de los elementos de dicho sistema en un ambiente tan dinámico.

Para abordar esta problemática se ha diseñado una metodología de aprendizaje automático automatizado para la reparametrización automática y óptima de la producción a partir de datos históricos del proceso productivo y así, poder generar recomendaciones inteligentes. Esta metodología ha sido desarrollada a partir de la extensión de trabajos previos con objetivos afines (Cruz et al., 2023). La metodología se compone de los servicios de entrenamiento/reentrenamiento de modelos, la predicción/inferencia y la optimización de parámetros de producción. Estos servicios se integran en un módulo de manera que, a partir de los datos de producción y un archivo de configuración, se ejecutan de manera automática para devolver los parámetros que harían funcionar al sistema productivo de manera óptima. Una vez que se haya ejecutado la metodología por primera vez, los servicios de predicción/inferencia se pueden consumir de manera independiente, por ejemplo, para determinar cómo funcionaría el sistema productivo bajo determinadas restricciones o encontrar los parámetros óptimos bajo distintas restricciones.

El propósito de la metodología es ser independiente del proceso productivo para que pueda ser utilizada por empresas con bajo nivel de adopción de la Industria 4.0 de diversos sectores. Sin embargo, debido a la gran variedad de formas en que se almacenan y gestionan los datos en las diferentes empresas, la metodología presupone que las empresas formatearán sus datos y prepararán el fichero de configuración de acuerdo con las pautas descritas en el Anexo A.

De manera general, la metodología propuesta trata de añadir al flujo de trabajo descrito en la Figura 2.9 la tarea de optimización de la producción. Sin embargo, en un proceso productivo normalmente se desean optimizar varios Indicadores Claves de Desempeño (KPIs) simultáneamente, por lo que es un problema de optimización multiobjetivo que puede ser descrito por la siguiente ecuación:

$$minimizar\ \bar{f}(x) = \begin{bmatrix} \bar{f}_1(x) \\ \bar{f}_2(x) \\ \vdots \\ \bar{f}_m(x) \end{bmatrix} \tag{1}$$

sujeto a restricciones de igualdad:

$$g_i(x) = 0, \quad i = 1, 2, \dots, p \tag{2}$$

y a restricciones de desigualdad:

$$h_i(x) \leq 0, \quad i = 1, 2, \dots, q \tag{3}$$

donde x es el vector de parámetros y variables de decisión del proceso productivo para el cual se desean hallar los valores que minimicen al vector de m funciones objetivo $\bar{f}(x)$, utilizado en este caso para estimar el valor de los KPIs del proceso productivo, a la vez que se cumplen con las restricciones g_i y h_i. Si se desea maximizar un KPI en lugar de minimizarlo, se debe reformular la función objetivo correspondiente, negándola.

En este caso, cada elemento del vector $\bar{f}(x)$ es un modelo de aprendizaje automático que debe ser entrenado para estimar un KPI del proceso productivo. Por esta razón, algunas etapas del flujo de trabajo de la Figura 2.9 deben ejecutarse tantas veces como KPIs se hayan definido hasta la obtención de un modelo para cada uno y, posteriormente, llevar a cabo la optimización, tal como se muestra, de modo simplificado, en la Figura 4.1. Para poder ejecutar todas las operaciones sin interrumpir el flujo de trabajo es necesario proveer, desde el inicio, el archivo de configuración que contiene los objetivos para cada KPI y las restricciones.

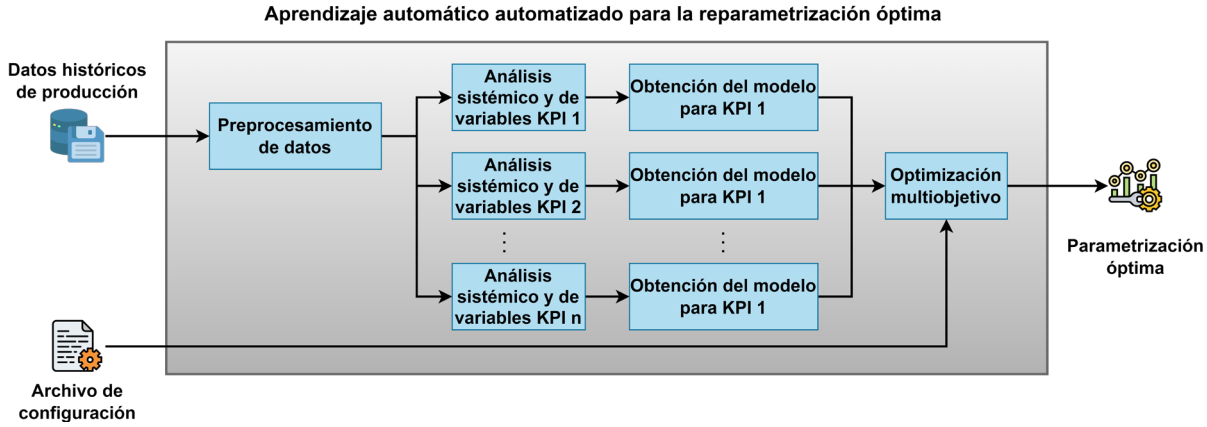

Figura 4.1: Metodología para la reparametrización óptima de los procesos productivos

Cada uno de los bloques incluidos en la Figura 4.1 se compone de varias operaciones, las cuales serán analizadas en las siguientes secciones. Adicionalmente, la Figura 4.2 muestra de manera más detallada el flujo de trabajo para ejecutar la metodología.

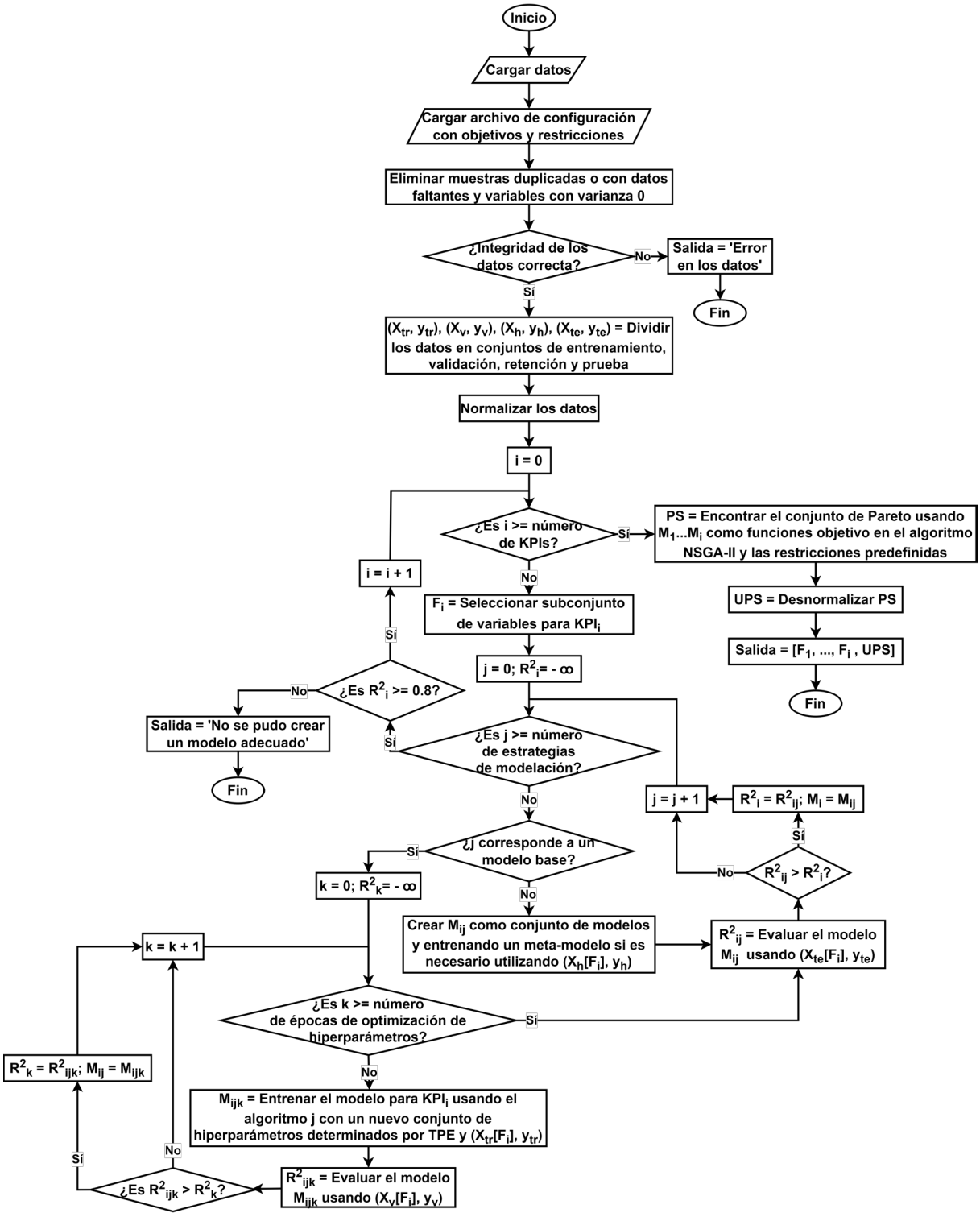

Figura 4.2: Flujo de trabajo de la metodología

4.2.1. Preprocesamiento de datos

Para ejecutar la metodología se debe de disponer de un conjunto de datos que pueda ser estructurado en formato tabular, donde cada fila represente una muestra. El conjunto de datos debe estar compuesto por n variables de entrada $X_{raw} = [X_1, X_2, \dots, X_n]$ y m variables de salida $y_{raw} = [y_1,\ y_2, \dots, y_m]$. El primer paso a ejecutar en la metodología es el preprocesamiento de los datos. El objetivo de este paso es llevar a cabo una transformación $(X_{raw}, y_{raw}) \rightarrow (X, y)$, en la que el conjunto de datos o *dataset* en bruto se transforma en un nuevo *dataset* que puede ser utilizado por los algoritmos de aprendizaje automático. Mediante el preprocesamiento se tratan de reducir las inconsistencias y redundancias, lo que conduce a un conjunto de datos más confiables y precisos que, a su vez, permiten mejorar el rendimiento y la capacidad de generalización de los modelos (Hajer et al., 2020).

La primera operación de preprocesamiento incluida en la metodología consiste en eliminar las muestras con datos faltantes. En entornos industriales, generalmente las muestras incompletas son consecuencia de fallos en los instrumentos de medición o de las comunicaciones. Estas muestras se deben gestionar tratando de mantener las propiedades estadísticas y la distribución original de los datos. La eliminación de muestras incompletas es un método común para lidiar con *datasets* con datos faltantes aleatoriamente (Ji et al., 2023). A continuación, se eliminan las muestras duplicadas, es decir, aquellas en las que tanto los valores de entrada como los valores de salida coinciden. Al eliminar las redundancias se busca evitar que los modelos presenten sesgos debido a la mayor frecuencia de ciertas muestras que no aportan información útil adicional. Por otra parte, si la varianza de alguna variable de entrada es cero, significa que su valor es el mismo en todo el *dataset*, por lo que tampoco aporta información útil y es eliminada.

Seguidamente, se verifica la integridad de los datos no eliminados. En este caso, se comprueba que los datos contenidos en el *dataset* sean numéricos. La mayoría de los algoritmos de aprendizaje automático son implementados de manera que utilizan variables de entrada numéricas, por lo que otros tipos de datos, como los categóricos, deben ser codificados utilizando un esquema de codificación ordinal o *one-hot* antes de iniciar la ejecución de la metodología propuesta. Además, los KPIs de un proceso productivo deben ser métricas cuantitativas, lo que implica que los

modelos que serán generados en pasos posteriores de la metodología serán modelos de regresión[9].

A continuación, se reorganizan las muestras de manera aleatoria y se divide el *dataset* en subconjuntos de entrenamiento, validación, retención y prueba. De esta forma, se asegura que los datos se distribuyan de manera uniforme entre los diferentes subconjuntos, lo cual ayuda a aumentar la capacidad de generalización del modelo y reducir el sobreajuste, ya que durante el entrenamiento y validación habrá más variabilidad en las muestras. El subconjunto de entrenamiento se utilizará para ajustar los modelos de aprendizaje automático, el de validación para optimizar sus hiperparámetros, el de retención para entrenar un metamodelo y el de prueba para evaluar el desempeño de los modelos. El porcentaje de datos que se asignará a cada subconjunto dependerá de la cantidad total de muestras en el *dataset*. Si el *dataset* contiene menos de 10000 muestras se dividirá en 70%/10%/10%/10% para los subconjuntos de entrenamiento, validación, retención y prueba, respectivamente. Por otra parte, si el *dataset* contiene más de 10000 muestras entonces se dividirá en 85%/5%/5%/5% para los subconjuntos de entrenamiento, validación, retención y prueba, respectivamente. Esta decisión se tomó para que en las situaciones donde hay alta disponibilidad de datos se puedan utilizar más muestras en el entrenamiento.

Por último, cada una de las n variables de entrada es normalizada, utilizando la siguiente ecuación:

$$X_{n_norm} = \frac{X_n - \mu_n}{\sigma_n} \tag{4}$$

donde X_n representa el vector que contiene los datos de una variable de entrada en uno de los subconjuntos de datos, μ_n representa el valor promedio de la n-ésima variable en el conjunto de entrenamiento y σ_n representa la desviación estándar de la n-ésima variable también en el conjunto de entrenamiento[10]. Las variables en los cuatro subconjuntos de datos son normalizadas utilizando la información estadística del conjunto de entrenamiento ya que estos son los datos que se utilizan para ajustar el modelo. De esta forma, se evita el filtrado de información de los subconjuntos de validación, retención y prueba en el proceso de entrenamiento, manteniendo así la integridad de la evaluación del modelo, ya que esta se lleva a cabo con datos que el modelo no ha procesado previamente.

[9] Un modelo de regresión es aquel en el que la variable de salida o dependiente puede tomar valores continuos
[10] Este método de normalización también es conocido estandarización

La normalización tiene como principal objetivo eliminar la diferencia de escala entre las variables. Este paso es de gran importancia en la metodología propuesta ya que en etapas posteriores se realiza un proceso de selección de variables. Al normalizar los datos se evita que en el proceso de selección de variables haya algún sesgo relacionado con las escalas o unidades de medida.

Por otra parte, la normalización es una operación fácilmente reversible. Debido a que en otras etapas de la metodología se deben encontrar combinaciones óptimas de las variables de entrada y esta búsqueda se realiza en el espacio normalizado, es necesario poder revertir la transformación de los valores a la escala con la que se trabaja en el proceso real y así, poder utilizarlos. Para revertir la operación de normalización se utiliza la siguiente ecuación:

$$x_n = x_{n_norm} \times \sigma_n + \mu_n \tag{5}$$

donde x_{n_norm} representa el dato normalizado de la n-ésima variable cuyo valor se quiere revertir y x_n representa el valor en la escala original.

4.2.2. Análisis sistémico y de variables

El desempeño de un modelo de aprendizaje automático depende, en gran medida, de las variables utilizadas. Muchos *datasets*, aún después de haber sido preprocesados, contienen variables poco informativas para un problema dado. Por esta razón, es común incluir en los flujos de trabajo de aprendizaje automático una etapa de análisis sistémico y de variables. Este análisis supone modificar el *dataset* para lograr una representación alternativa que proporcione información relevante y significativa para mejorar la capacidad predictiva y el desempeño de los modelos.

Dos de las técnicas más utilizadas en el análisis sistémico son la creación de variables y la selección de variables. La creación de variables implica utilizar la información contenida en el *dataset* así como conocimientos del dominio del problema tratado para diseñar nuevas variables con una alta capacidad predictiva. Por otra parte, la selección de variables supone determinar cuáles son las variables más informativas y conservarlas mientras se descarta al resto. De esta forma se reduce la dimensionalidad de los datos, lo que permite utilizar modelos más compactos con mayor capacidad de generalización y que se ejecuten en menor tiempo de cómputo (Al-Tashi et al., 2020). Debido a que la metodología propuesta no ha sido diseñada para ningún dominio en específico y, por tanto, no existe conocimiento *a priori* de la aplicación, no se contempla la creación de variables. Sin embargo, la selección de variables sí ha sido incluida en la metodología. Al

determinar cuáles son las variables relevantes para predecir un KPI se cuenta con un nivel básico de explicabilidad en la metodología.

La selección de variables se enfoca principalmente en tres tipos de métodos: métodos de filtrado, métodos de envoltura y métodos integrados. Aunque ciertos estudios incluyen también el Análisis de Componentes Principales dentro de los métodos de selección de variables, este es un método de compresión de datos que permite proyectar los datos en un espacio dimensional más reducido. A pesar de las ventajas de este método, con él no se pueden determinar las variables relevantes para predecir un KPI, por lo que no se incluyó en la metodología.

Entre los métodos de filtrado, la selección univariada es el enfoque más utilizado. En este se llevan a cabo pruebas estadísticas para determinar cuáles son las variables de entrada que tienen mayor influencia en la variable de salida. Las pruebas estadísticas más comunes para selección de variables incluyen los coeficientes de correlación de Pearson (Nasir et al., 2020) y Spearman (J & S, 2018), el análisis de varianza (Ramírez et al., 2018), el coeficiente de correlación de rango de Kendall (He et al., 2018), la prueba chi-cuadrado (Bahassine et al., 2020) y el análisis de información mutua (Sharmin et al., 2019). Estas pruebas generalmente son aplicadas en dependencia de los tipos de datos de entrada y salida presentes en el *dataset*. Para los *datasets* con datos de entrada y salida numéricos, como los que contempla la metodología propuesta, se suelen utilizar los coeficientes de correlación de Pearson para identificar dependencias lineales y de Spearman para detectar dependencias no lineales monotónicas (Gao et al., 2019). Para calcular los coeficientes de correlación de Pearson (r_{xy}) y de Spearman (ρ) para una muestra compuesta de n pares de datos $\{(x_i, y_i)\}_{i=1}^{n}$, se utilizan las siguientes ecuaciones:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{6}$$

$$\rho_{xy} = 1 - \frac{6\sum_{i=1}^{n} D_{x_i,y_i}}{n(n^2 - 1)} \tag{7}$$

donde $\bar{x}$ y $\bar{y}$ son los valores medios de las variables para la muestra analizada. En el caso del coeficiente de correlación de Spearman, los datos son ordenados y reemplazados por su respectivo rango de ordenamiento y D_{x_i,y_i} se calcula como la diferencia de rango correspondiente a x_i y y_i. Ambos coeficientes toman valores en el rango $[-1, 1]$, donde los valores cercanos a los extremos indican una alta correlación y los valores cercanos a 0 indican una baja correlación.

Por otra parte, los métodos de envoltura encapsulan un modelo específico y lo utilizan para seleccionar las variables de entrada al entrenarlo varias veces con diferentes combinaciones de estas y evaluar los resultados obtenidos. El modelo base que se utiliza generalmente para los métodos de envoltura es la máquina de vectores de soporte (SVM). Uno de los métodos de envoltura más utilizados es la Eliminación Recursiva de Variables, en el cual se entrena el modelo base utilizando todas las variables de entrada y, se determina como cada una de ellas influye individualmente en el cálculo del error variando sus valores. Posteriormente, las que menos influencia tengan son eliminadas y se vuelve a iniciar el proceso con las variables restantes. Este proceso se repite hasta reducir el número de variables al valor deseado (Albashish et al., 2021). A pesar de la popularidad de los métodos de envoltura estos presentan la desventaja de tener que entrenar en múltiples ocasiones al modelo base, lo que conlleva a que estos métodos tengan un coste computacional elevado.

Finalmente, los métodos integrados determinan la contribución de las variables de entrada a la disminución del error durante la creación de un modelo. Es decir, que en estos métodos la selección de variables y el entrenamiento del modelo se realizan a la vez, al ajustarse los parámetros internos del modelo para determinar la ponderación/importancia que se le da a cada variable. Dos de los tipos de modelos más utilizados en los métodos integrados son *Least Absolute Shrinkage and Selection Operator* (LASSO) y *Random Forest* (RF). En el caso de LASSO, al ajustar el modelo, si el coeficiente correspondiente a una variable tiene un valor pequeño significa que dicha variable no tiene mucha influencia en la predicción. En el caso de RF se busca disminuir una medida de la impureza (varianza para problemas de regresión). Si como resultado de una bifurcación basada en una variable se reduce más la impureza que con otra variable entonces, se considera que la primera tiene más relevancia. De esta forma se calcula el promedio de la reducción de impureza de cada variable considerando todos los árboles de decisión y se seleccionan las variables de entrada que más contribuyan a reducir la impureza. El coste computacional de estos métodos se encuentra en un nivel intermedio entre los métodos de filtrado y los de envoltura.

Debido a que la metodología ha sido diseñada para poder ser aplicada en una amplia variedad de procesos y, que no se conoce de antemano la complejidad de los datos, es decir, cuantas variables o muestras se deberán procesar, se optó por utilizar los métodos de menor gasto computacional. En el proceso de selección de variables para una salida y, se decide si eliminar una variable x, verificando si se

cumplen simultáneamente las siguientes condiciones $|r_{xy}| < 0.2$ y $|\rho_{xy}| < 0.2$. Lo que quiere decir que se toman en cuenta tanto el coeficiente de correlación de Pearson como el de Spearman. De esta forma se espera obtener un subconjunto de variables con alto poder predictivo al incluir predictores con valores significativos de correlación lineal o no lineal. Sin embargo, puede darse el caso que debido a la complejidad de la relación entre los datos de entrada y salida no se detecte ninguna dependencia lineal o no lineal monótona. Si tras aplicar el proceso de selección descrito previamente son descartadas todas las variables, entonces se obvia el resultado del proceso de selección de variables y se utilizan todas en la creación de los modelos.

Adicionalmente se verifica si existe una multicolinealidad significativa entre cada par de variables de entrada x_i y x_j para todo $i \neq j$ en $i, j = 1,2, \dots n$, donde n es el número de variables de entrada. Esta verificación se realiza evaluando la siguiente condición: $|r_{x_i x_j}| > 0.95$. Si algún par de variables de entrada cumple la condición entonces una de ellas es descartada aleatoriamente.

4.2.3. Diseño y procedimiento para el modelado automático

Una vez que se ha finalizado la etapa de análisis sistémico y de variables se deben construir los modelos para cada variable de salida. Esta etapa incluye tanto el entrenamiento de cuatro tipos diferentes de modelos de aprendizaje automático (MLP, SVR, XGBoost, *k*NN), así como la optimización de los hiperparámetros de cada uno de ellos. Además, se consideran dos técnicas de aprendizaje de conjunto basadas en los modelos obtenidos previamente: combinación de predicciones y metamodelo mixto. Una vez que se cuenta con todos los modelos, se realiza una comparación y se selecciona el de mejor desempeño.

Esta estrategia de exploración sistemática de diferentes técnicas de modelado se realiza ya que no se puede asumir que un mismo modelo funcionará bien para representar todos los procesos productivos. Tal como establece el teorema de *No Free Lunch*, no existe un algoritmo universalmente superior que pueda considerarse el mejor para todos los conjuntos de datos o problemas (Adam et al., 2019). Al probar múltiples modelos, se tiene la oportunidad de evaluar cómo se comporta cada uno para el problema específico que se esté tratando, lo cual permite elegir de una manera práctica el modelo más adecuado.

4.2.3.1. Perceptrón multicapa

Las redes neuronales artificiales son modelos computacionales inspirados en la estructura y funcionamiento del cerebro humano. Estas son capaces de aprender patrones y relaciones complejas a partir de datos. Las redes neuronales se componen de múltiples unidades básicas de procesamiento llamadas neuronas artificiales o nodos, las cuales realizan operaciones matemáticas a partir de datos de entrada para generar un valor de salida. La Figura 4.3 muestra una representación de una neurona artificial.

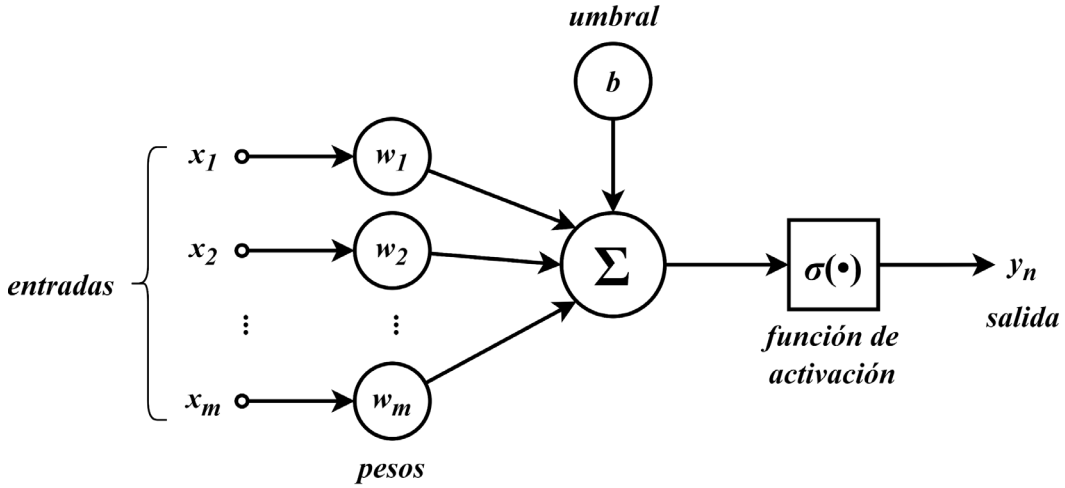

Figura 4.3: Neurona artificial

La salida y_n de una neurona artificial con m entradas se calcula mediante la siguiente ecuación:

$$y_n = \sigma\left(\sum_{i=1}^{m} w_i \times x_i + b\right) \tag{8}$$

donde σ es una función de activación (generalmente no lineal), x_i es el valor de la i-ésima entrada, w_i es el peso correspondiente a dicha entrada, el cual permite ponderar su contribución en el cálculo de la salida y b es un valor conocido como umbral, el cual permite ajustar el nivel de activación de la neurona.

Al interconectar varias neuronas artificiales se pueden obtener estructuras más complejas, tales como el perceptrón multicapa (MLP), el cual es un tipo de red neuronal utilizado ampliamente debido a su capacidad de generalización, sobre todo en el caso de funciones no lineales con representaciones complejas y datos de

alta dimensionalidad (Lotfi & Rezaee, 2018). En este tipo de red cada una de las neuronas de una capa está conectada con todas las neuronas de la capa anterior y de la capa siguiente, tal como se muestra en la Figura 4.4. La primera capa (de entrada) recibe los datos de entrada, que se propagan a través de una o más capas intermedias (ocultas) hasta llegar a la última capa (de salida), donde se genera la predicción. En el caso de la metodología propuesta, como se desea tener un modelo individual para cada salida, la última capa estaría compuesta por una única neurona, aunque en otras situaciones esta capa podría contener múltiples neuronas.

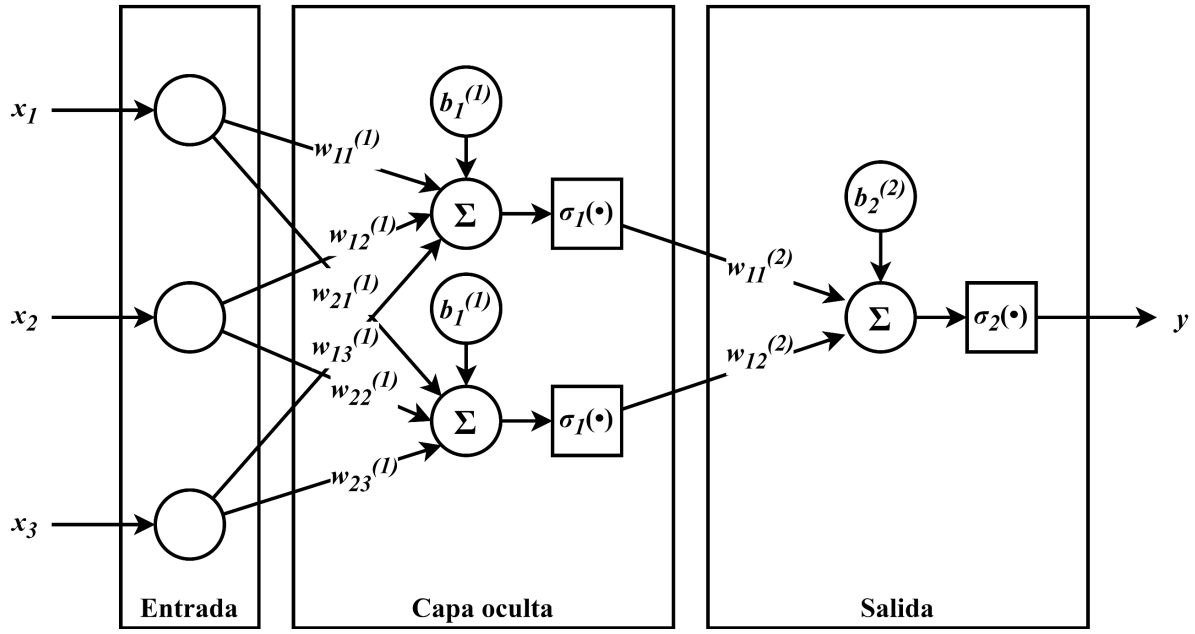

Figura 4.4: Perceptrón multicapa

En este tipo de red, la salida $y_c \in \mathbb{R}^{j\times 1}$ de la capa c se calcula mediante la siguiente ecuación:

$$y_c = \sigma(Wx + b) \tag{9}$$

donde σ es la función de activación de la capa, $W \in \mathbb{R}^{j\times d}$ es la matriz que contiene los pesos que asocian cada entrada con cada una de las j neuronas de la capa, $x \in \mathbb{R}^{d\times 1}$ es el vector de entradas a la capa, $b \in \mathbb{R}^{j\times 1}$ es el vector de umbrales.

Este modelo es capaz de ajustar los valores de los pesos y umbrales de cada capa mediante un proceso de entrenamiento iterativo que se inicia tomando un conjunto de datos con n muestras y realizando una inferencia con la red. Seguidamente, se

compara la salida esperada $y \in \mathbb{R}^{n\times 1}$ con la inferida $\hat{y} \in \mathbb{R}^{n\times 1}$ y se cuantifica la discrepancia entre estos valores mediante una función de pérdida, como el error medio cuadrático ($\mathcal{E}$) descrito mediante la siguiente ecuación:

$$\mathcal{E} = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2 \tag{10}$$

A continuación, se calculan las derivadas parciales del error con respecto a los pesos y umbrales utilizando el algoritmo de retropropagación del error y se ajustan los valores de estos parámetros en la dirección de la minimización del error mediante la siguiente ecuación:

$$p \leftarrow p - \alpha * \frac{\partial \mathcal{E}}{\partial p} \tag{11}$$

donde p representa el vector de umbrales o la matriz de pesos, en dependencia de los parámetros que se estén actualizando y α es la tasa de aprendizaje, la cual sirve para controlar la magnitud de los ajustes realizados a los parámetros de la red.

Este proceso de realizar una inferencia y utilizar la retropropagación del error para ajustar los parámetros de la red neuronal es repetido durante un número determinado de épocas o hasta que se cumpla algún otro criterio de parada, tras lo cual se considera que la red ha sido entrenada. El proceso de entrenamiento descrito puede ser optimizado mediante técnicas como *Root Mean Square Propagation* o *Adaptive Moment Estimation* (Kingma & Ba, 2014). Una vez que termina el entrenamiento, la red neuronal puede ser utilizada para realizar predicciones.

4.2.3.2. Vectores de soporte para regresión

Vectores de soporte para regresión (SVR) es un algoritmo de aprendizaje automático que busca encontrar un hiperplano que sea capaz de representar la relación entre las variables de entrada y la variable de salida. El objetivo es que el hiperplano minimice la diferencia entre las salidas predichas y los valores reales de entrenamiento, sujeto a una restricción de tolerancia de error (Zanjani et al., 2020). Esta restricción permite cierto margen de error en las predicciones, lo que proporciona una mayor flexibilidad para manejar datos ruidosos o atípicos. Para casos lineales con d variables de entrada, al aplicar SVR se intenta resolver el siguiente problema:

$$minimizar: \frac{1}{2}\|w\|^2 \tag{12}$$

$$sujeto\ a: |y_i - wx_i - b| \leq \varepsilon, \qquad para\ i = 1, 2, \ldots, n \tag{13}$$

donde $w \in \mathbb{R}^d$ y $b \in \mathbb{R}$ son los parámetros que definen al hiperplano, ε es el margen de error (también conocido como tubo ε-insensitivo), $x_i \in \mathbb{R}^{d\times 1}$ y $y_i \in \mathbb{R}$ son los datos de entrada y salida de la i-ésima muestra del conjunto de datos de entrenamiento compuesto por n muestras.

En la práctica, es posible que no se pueda encontrar una solución a dicho problema debido a que haya muestras que no se encuentren dentro del tubo ε-insensitivo, como se puede ver en la Figura 4.5.

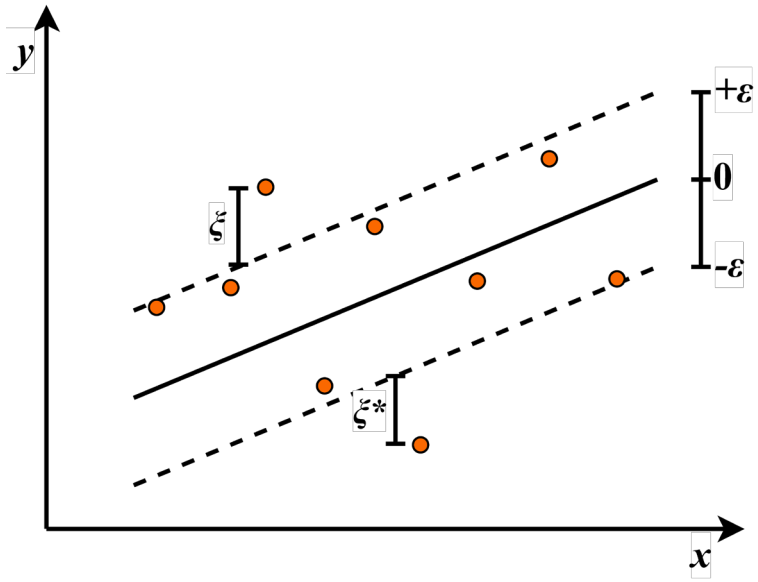

Figura 4.5: Puntos que exceden el margen de error permitido

En estos casos, se introducen las variables de holgura ξ y ξ^* para cada muestra permitiendo que estas se encuentren fuera del tubo ε-insensitivo, pero aplicando una penalización por ello. Teniendo esto en cuenta, se debe redefinir el problema de la siguiente manera:

$$minimizar: \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{14}$$

$$sujeto\ a: y_i - wx_i - b \leq \varepsilon + \xi_i, \qquad para\ i = 1, 2, \ldots, n \tag{15}$$

$$wx_i + b - y_i \leq \varepsilon + \xi_i^*, \qquad para\ i = 1, 2, \dots, n \qquad (16)$$
$$\xi_i \geq 0, \qquad para\ i = 1, 2, \dots, n \qquad (17)$$
$$\xi_i^* \geq 0, \qquad para\ i = 1, 2, \dots, n \qquad (18)$$

donde la constante C es un parámetro de regularización que controla la penalización aplicada a las muestras que se encuentran fuera del tubo ε-insensitivo ayudando a prevenir el sobreajuste.

Sin embargo, el enfoque lineal descrito no es adecuado para problemas de regresión más complejos donde es muy difícil encontrar el hiperplano deseado en el espacio dimensional original de las variables. Por esta razón, es común utilizar algún tipo de *kernel* ($\varphi(x)$) para mapear las variables de entrada a un espacio dimensional superior donde sea más sencillo encontrar el hiperplano que cumpla las condiciones descritas, tal como se muestra en la Figura 4.6. Entre los diferentes tipos de *kernels* que se suelen utilizar en estos casos se encuentran los lineales, los gaussianos y los polinomiales.

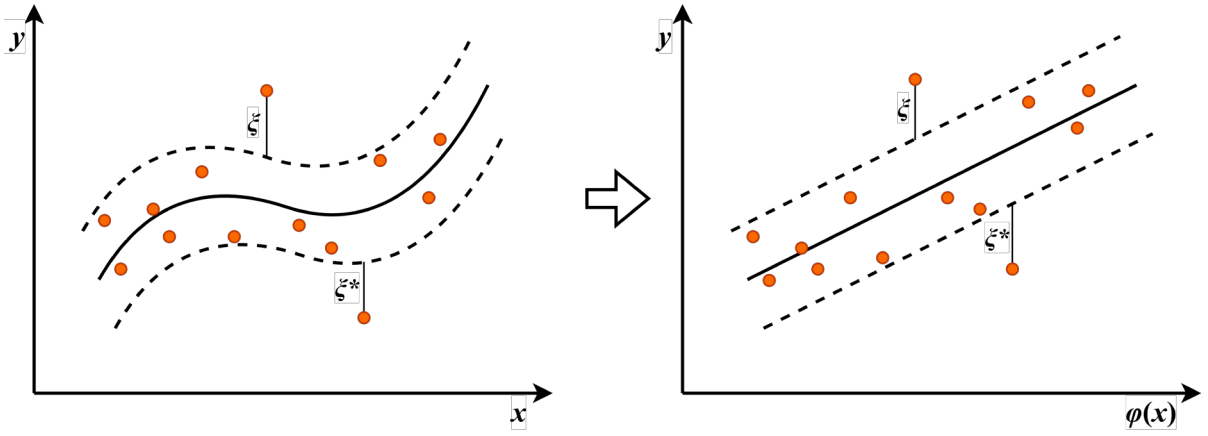

Figura 4.6: Aplicación de un *kernel* para transformar un problema no lineal a uno lineal

4.2.3.3. Refuerzo de gradientes extremo

Refuerzo de gradientes extremo (XGBoost) implementa una versión considerablemente eficiente de un algoritmo de conjunto de modelos llamado refuerzo del gradiente, en el cual se combinan las predicciones de múltiples árboles de regresión relativamente simples para crear un modelo más robusto y preciso (T. Chen & Guestrin, 2016). En este algoritmo, los árboles de regresión se crean de manera secuencial. El primer árbol se entrena con los datos de entrada/salida y, a

continuación, cada nuevo árbol se entrena para intentar predecir los residuales de los errores producidos por los árboles anteriores. Para crear cada árbol se utiliza como objetivo la expansión de Taylor de segundo orden de la función de pérdida, la cual se requiere minimizar. La siguiente ecuación describe la función objetivo $q^{(t)}$ para el t-ésimo árbol:

$$q^{(t)} = \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \omega(f_t) \tag{19}$$

donde n es el número de muestras en el conjunto de entrenamiento, $l(y_i, \hat{y}_i^{(t-1)})$ es el valor de la función de perdida para el árbol precedente, x_i es el vector de entradas de la i-ésima muestra, y_i es la salida real de la i-ésima muestra, $\hat{y}_i$ es la salida predicha para la i-ésima muestra, f_t es la función que permite calcular la salida del t-ésimo árbol, ω es una función que permite medir la complejidad del árbol analizado, mientras que g_i y h_i se definen de la siguiente forma:

$$g_i = \partial_{\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}_i^{(t-1)}\right) \tag{20}$$

$$h_i = \partial^2_{\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}_i^{(t-1)}\right) \tag{21}$$

Por otra parte, ω se define de la siguiente forma:

$$\omega = \gamma L + \frac{1}{2} \lambda \sum_{j=1}^{L} v_j^2 \tag{22}$$

donde γ y λ son parámetros de regularización, L es el número de hojas del árbol y v es el vector de puntuaciones en las hojas.

XGBoost implementa además una combinación de técnicas avanzadas de paralelismo computacional, supervisión de la memoria caché, regularización, poda de árboles y control de sobreajuste para intentar mantener una alta capacidad de generalización a la vez que se entrena en un período de tiempo relativamente corto comparado con otros algoritmos. La popularidad de XGBoost en los últimos años está dada por su alto rendimiento y velocidad de ejecución, pudiendo, además, escalar de manera eficiente para manejar grandes conjuntos de datos, lo que ha propiciado que mediante su uso se hayan ganado varias competiciones de aprendizaje automático y que sea la base de un gran número de aplicaciones industriales.

La predicción en este modelo se realiza combinando las salidas de todos los árboles, tal como se muestra en la Figura 4.7. Específicamente, para calcular el valor predicho por este modelo para la i-ésima muestra $\hat{y}_i$ a partir del vector de entradas x_i se utiliza la siguiente ecuación:

$$\hat{y}_i = \sum_{t=1}^{T} f_t(x_i) \tag{23}$$

donde T es el número de árboles y f_t es la función que permite calcular la salida del t-ésimo árbol.

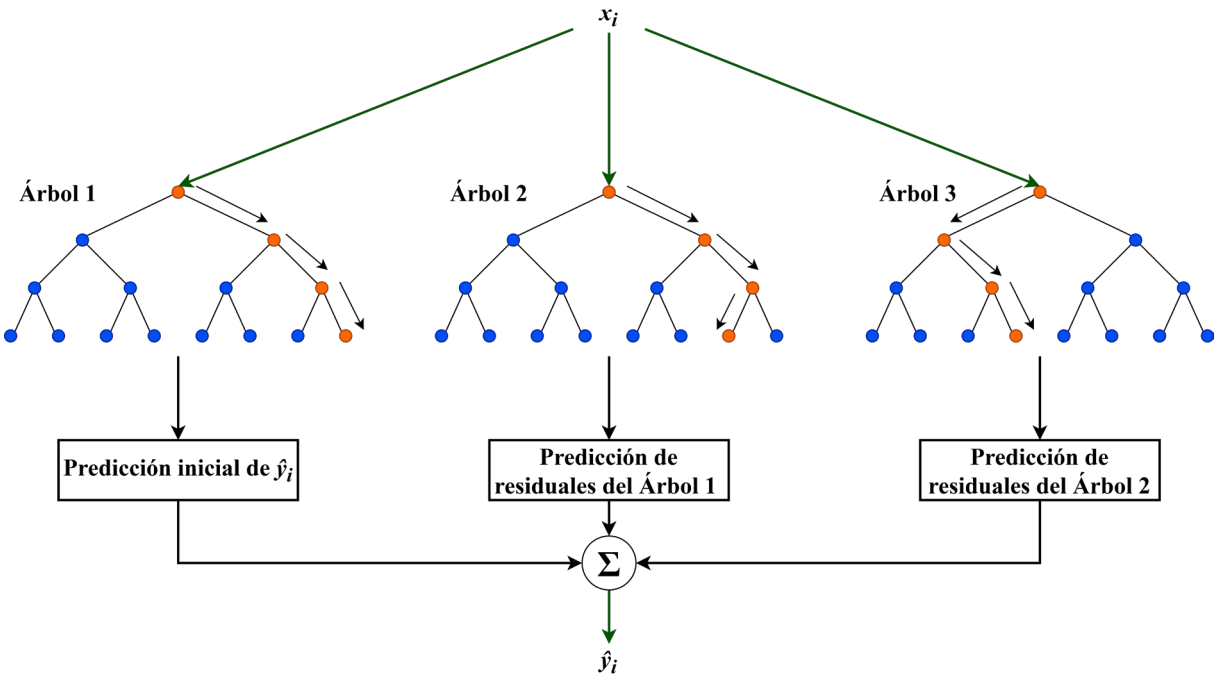

Figura 4.7: Predicción con XGBoost

4.2.3.4. *k*-vecinos más cercanos

k-vecinos más cercanos (*k*NN) es un algoritmo de aprendizaje automático de fácil implementación y gran robustez, demostrando, habitualmente, un buen desempeño en problemas no lineales. En problemas de regresión, para calcular la salida de este algoritmo primeramente se deben determinar cuáles son las k muestras (vecinos) más cercanas en el conjunto de datos de entrenamiento (Durbin et al., 2021), tal como se muestra en la Figura 4.8.

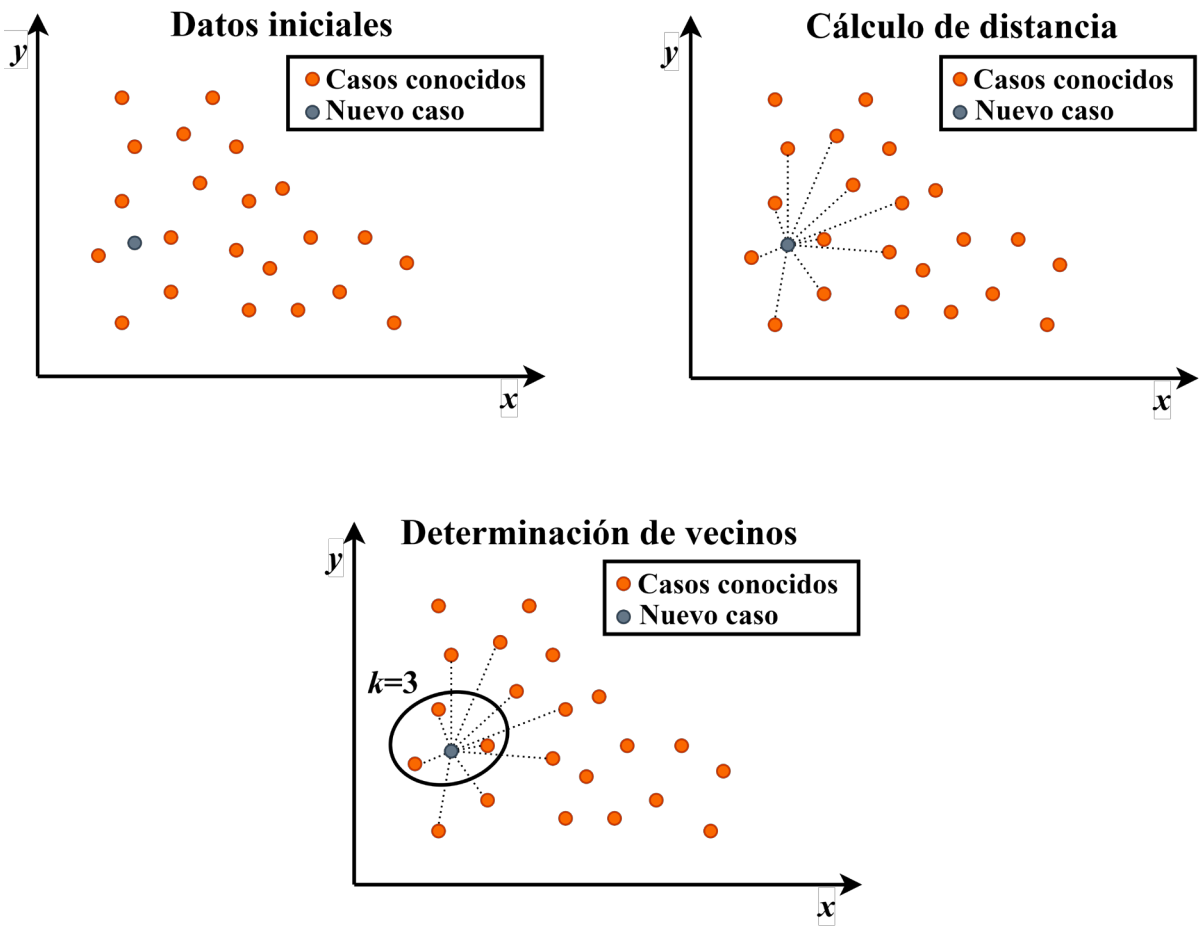

Figura 4.8: Determinación de los k-vecinos más cercanos

Generalmente, la distancia hacia los vecinos más cercanos se calcula utilizando la distancia de Minkowski (D) en el espacio n-dimensional de las variables de entrada, según la siguiente ecuación:

$$D = \left(\sum_{i=1}^{n} |x_i - v_i|^p \right)^{\frac{1}{p}} \tag{24}$$

donde $x \in \mathbb{R}^d$ es el vector con los valores de las entradas de la muestra cuya salida se quiere predecir, $v \in \mathbb{R}^d$ es el vector con los valores de las entradas de la muestra con respecto a la cual se quiere calcular la distancia y p es el orden de la distancia.

Este algoritmo requiere calcular la distancia a cada una de las muestras en el conjunto de entrenamiento cada vez que se quiera realizar una predicción, aunque este cálculo suele ser rápido si se realiza una implementación vectorizada del

algoritmo. Una vez que se han calculado las distancias a todas las muestras en el conjunto de entrenamiento, se pueden determinar cuáles son los k vecinos más cercanos y la predicción $\hat{y}$ se calcula promediando las salidas y de dichos vecinos mediante la siguiente ecuación:

$$\hat{y} = \frac{1}{k}\sum_{i=1}^{k} y_i \tag{25}$$

4.2.3.5. Optimización de los modelos

Durante el entrenamiento de los modelos de aprendizaje automático supervisado, se ajusta un conjunto de parámetros con el fin de aprender las relaciones subyacentes entre los datos de entrada y salida. Pero, existen otros parámetros, comúnmente conocidos como hiperparámetros, los cuales son utilizados para definir la estructura de los modelos o controlar el proceso de entrenamiento. Los valores de los hiperparámetros son definidos antes de comenzar el proceso de entrenamiento y, generalmente, no se modifican una vez que este ha comenzado. La correcta elección de los hiperparámetros tiene una gran influencia en la capacidad de aprendizaje y el desempeño del modelo (Yang & Shami, 2020). Sin embargo, debido a que las características de cada problema pueden variar, los hiperparámetros que permiten entrenar eficazmente a un modelo en una determinada situación, pueden no ser idóneos para otra situación diferente.

Una opción para determinar los valores adecuados para los hiperparámetros consiste en probar manualmente diferentes combinaciones, entrenar los modelos y comparar su desempeño. Sin embargo, debido a la ineficiencia de este método, se han desarrollado estrategias más sistemáticas y automatizadas para la optimización de hiperparámetros. Entre estas estrategias las más difundidas son la búsqueda en rejilla y la búsqueda aleatoria. En la búsqueda en rejilla se discretiza el espacio de búsqueda de todos los hiperparámetros y se realiza una búsqueda exhaustiva al evaluar todas las posibles combinaciones. Por otra parte, en la búsqueda aleatoria se evalúa un número determinado de combinaciones al azar de los valores de los hiperparámetros. Estas estrategias presentan el inconveniente de no utilizar la información de combinaciones de hiperparámetros evaluadas previamente, por lo que pueden incurrir en un gasto computacional innecesario al centrar la búsqueda en direcciones con pocas probabilidades de contener la combinación de hiperparámetros más adecuada.

Con el objetivo de explorar de una manera más eficiente el espacio de búsqueda de los hiperparámetros se pueden utilizar estrategias basadas en optimización bayesiana. Uno de los algoritmos de optimización bayesiana más utilizados en la actualidad es *Tree-structured Parzen Estimator* (TPE), ya que ha demostrado gran eficiencia computacional y permite trabajar con distintos tipos de distribuciones de probabilidad en dependencia del tipo de variable de los hiperparámeros (Ozaki et al., 2020). En este algoritmo secuencial se busca modelar la probabilidad del hiperparámetro λ dado que se tengan valores altos (no deseados) o bajos (deseados) de la función de coste f mediante las respectivas distribuciones de probabilidad $g(\lambda)$ y $l(\lambda)$ definidas de la siguiente manera:

$$g(\lambda) = p(\lambda | f > f^*) \tag{26}$$

$$l(\lambda) = p(\lambda | f \leq f^*) \tag{27}$$

donde f^* representa el valor de la función de coste asociado a un cuantil prefijado del número total de observaciones.

En TPE se busca el valor de λ que maximice la función M, la cual presenta cierta similitud con la función de mejora esperada, la cual se puede describir de manera simplificada como la relación entre estas dos distribuciones de probabilidad:

$$M(\lambda) \propto \frac{l(\lambda)}{g(\lambda)} \tag{28}$$

de manera que, donde se obtenga el mayor valor de M se considera que existe una mayor probabilidad de encontrar un valor prometedor de λ. La Figura 4.9 muestra un ejemplo simplificado de la determinación de este valor para una función de coste hipotética que depende de un único hiperparámetro. Teniendo esto en cuenta, se procede a evaluar el valor más prometedor encontrado del hiperparámetro y con los resultados se actualizan los valores de $g(\lambda)$ y $l(\lambda)$ y, seguidamente, se repite el proceso (Madrigal et al., 2019).

Con el fin de optimizar los hiperparámetros de los modelos incluidos en la metodología se decidió aplicar TPE dadas las ventajas que ofrece para trabajar simultáneamente tanto con hiperparámetros continuos como categóricos y su alta eficiencia computacional. En la práctica, al aplicar TPE, en cada iteración se entrena una nueva versión del modelo que se esté optimizando con la combinación de hiperparámetros más prometedora utilizando los datos del subconjunto de entrenamiento y se evalúa el desempeño del modelo al realizar predicciones con los datos del subconjunto de validación. Este procedimiento se ejecuta para cada uno

de los tipos de modelos descritos previamente, de manera que se cuente con una versión con los hiperparámetros optimizados de MLP, SVR, XGBoost y *k*NN. La Tabla 4.1 muestra los rangos explorados mediante TPE para los diferentes tipos de modelo incluidos en la metodología propuesta.

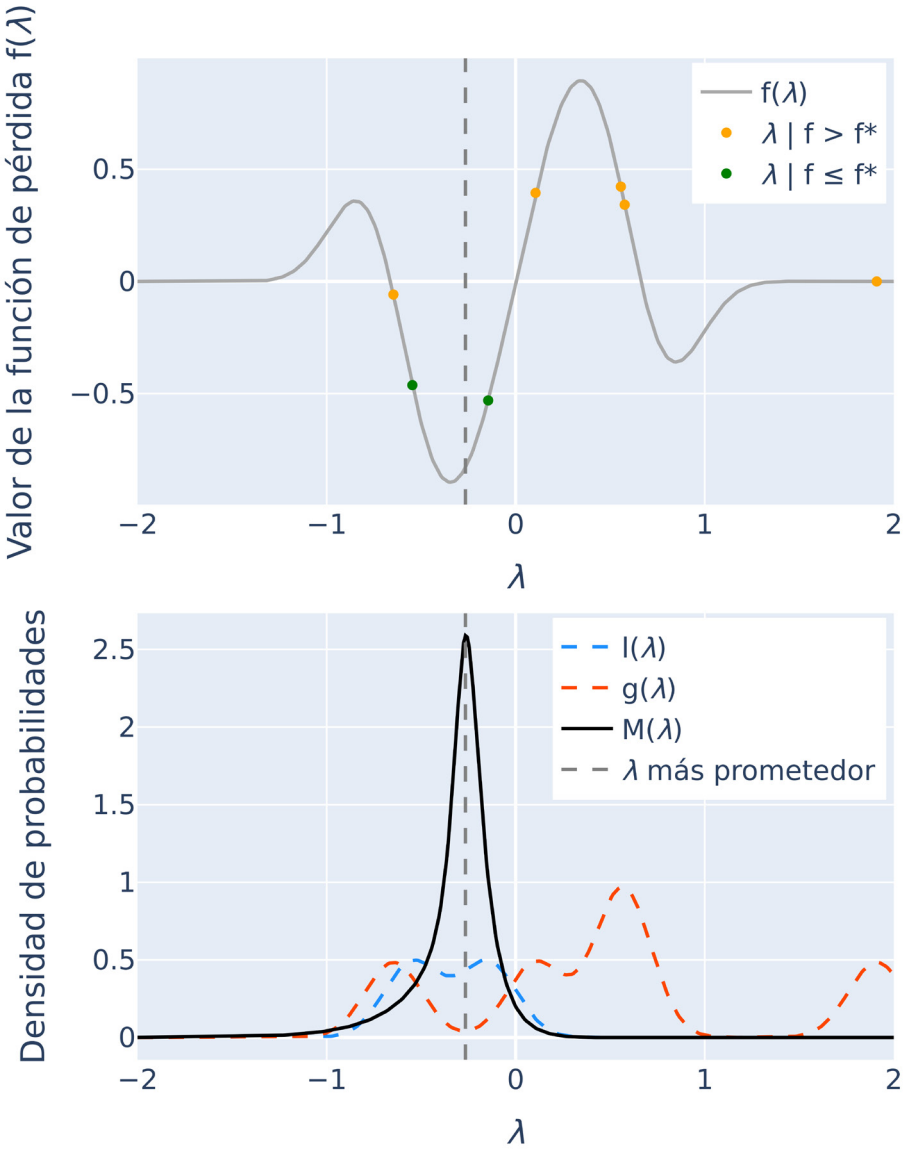

Figura 4.9: Ejemplo simplificado de la determinación del valor más prometedor de un hiperparámetro

Tabla 4.1: Rangos explorados durante la optimización de hiperparámetros

Modelo	Hiperparámetro	Tipo/escala	Rango/valor
MLP	número de capas ocultas	entero/lineal	1-5

	función de activación en la capa de entrada y capas ocultas (σ)	categórico	ReLU, sigmoide, tanh
	número de unidades por capa	entero/lineal	4-128
	tasa de aprendizaje (α)	real/logarítmica	0.00001-0.1
SVR	parámetro de regularización (C)	real/logarítmica	0.001-1000
	anchura del tubo de insensibilidad (ε)	real/logarítmica	0.001-10
XGBoost	número de árboles	entero/lineal	100-500
	profundidad máxima	entero/lineal	3-10
	tasa de aprendizaje	real/logarítmica	0.01-0.1
	tasa de submuestreo	real/lineal	0.5-1
	tasa de submuestreo de columnas por árbol	real/lineal	0.5-1
	reducción mínima de la función de coste para crear una bifurcación (γ)	real/lineal	0-5
*k*NN	número de vecinos (k)	entero/lineal	1-20
	función de ponderación	categórico	uniforme, distancia
	orden de la distancia de Minkowski	entero/lineal	1-2

4.2.3.6. Aprendizaje de conjunto: Combinación de predicciones

En las subsecciones anteriores se describieron los distintos tipos de modelos a evaluar en la metodología propuesta, así como la estrategia para optimizar sus hiperparámetros. Teniendo en cuenta que tras la optimización de hiperparámetros de cada modelo se tienen varios algoritmos para representar una misma salida y que el gasto computacional adicional que supone en el proceso de modelado es mínimo con respecto a la optimización de hiperparámetros, se ha decidido explorar dos estrategias de modelado adicionales. Estas reúnen la información de los modelos creados previamente y la utilizan para intentar minimizar el error en las predicciones. Dichas estrategias de modelado han sido adaptadas de trabajos desarrollados previamente, en los cuales se comprueba que contar con diversos modelos contribuye a la mejora de las predicciones ya que cada modelo distribuye sus errores de manera diferente en distintas regiones del espacio de las variables de entrada (Cruz et al., 2021, 2022). La Figura 4.10 muestra un ejemplo hipotético donde se puede apreciar como un modelo se ajusta mejor a las muestras en las que la variable de entrada tiene menores valores y otro se ajusta mejor a las muestras en las que la variable de entrada tiene mayores valores. Sin embargo, al combinar la información de ambos modelos se logra una mejor generalización sobre todo el espacio de la variable de entrada.

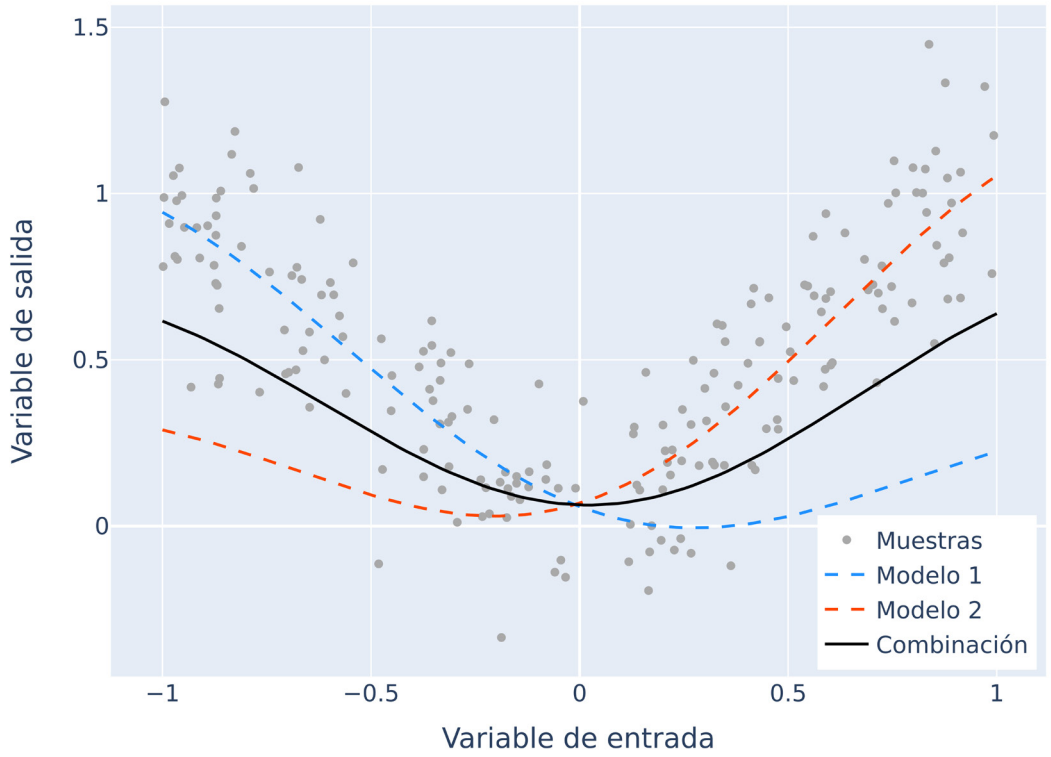

Figura 4.10: Ejemplo de mejora en la generalización al combinar las salidas de dos modelos

La primera de estas estrategias, la combinación de predicciones, se basa en la unificación de las predicciones de todos los modelos individuales. La manera de unificar las predicciones varía en dependencia del tipo de problema que se trate. En problemas de clasificación, la predicción del conjunto se realiza por medio de las políticas de votación: *hard voting* o *soft voting*. La política de *hard voting* consiste en seleccionar la clase que mayor cantidad de modelos predicen, mientras que la de *soft voting* implica combinar las probabilidades predichas para cada clase por todos los modelos y, al final, la clase que mayor probabilidad tenga es seleccionada. En problemas de regresión, para obtener la predicción del conjunto, generalmente, se promedia la salida de los modelos individuales. Otra opción, si se cuenta con modelos cuya salida son distribuciones de probabilidad en lugar de un valor directamente, es calcular el promedio ponderado de las distribuciones de probabilidad y determinar el valor esperado, el cual se considera como la predicción del conjunto.

En el caso de la metodología propuesta, solo se están considerando salidas que representen KPIs numéricos. Por otra parte, todos los modelos incluidos predicen

directamente un valor y no una distribución de probabilidades. Por esta razón, para determinar la predicción del conjunto de modelos que incluye al MLP, SVR, XGBoost y *k*NN, primeramente, se obtienen sus predicciones individuales y a continuación se promedian, tal como se muestra en la Figura 4.11.

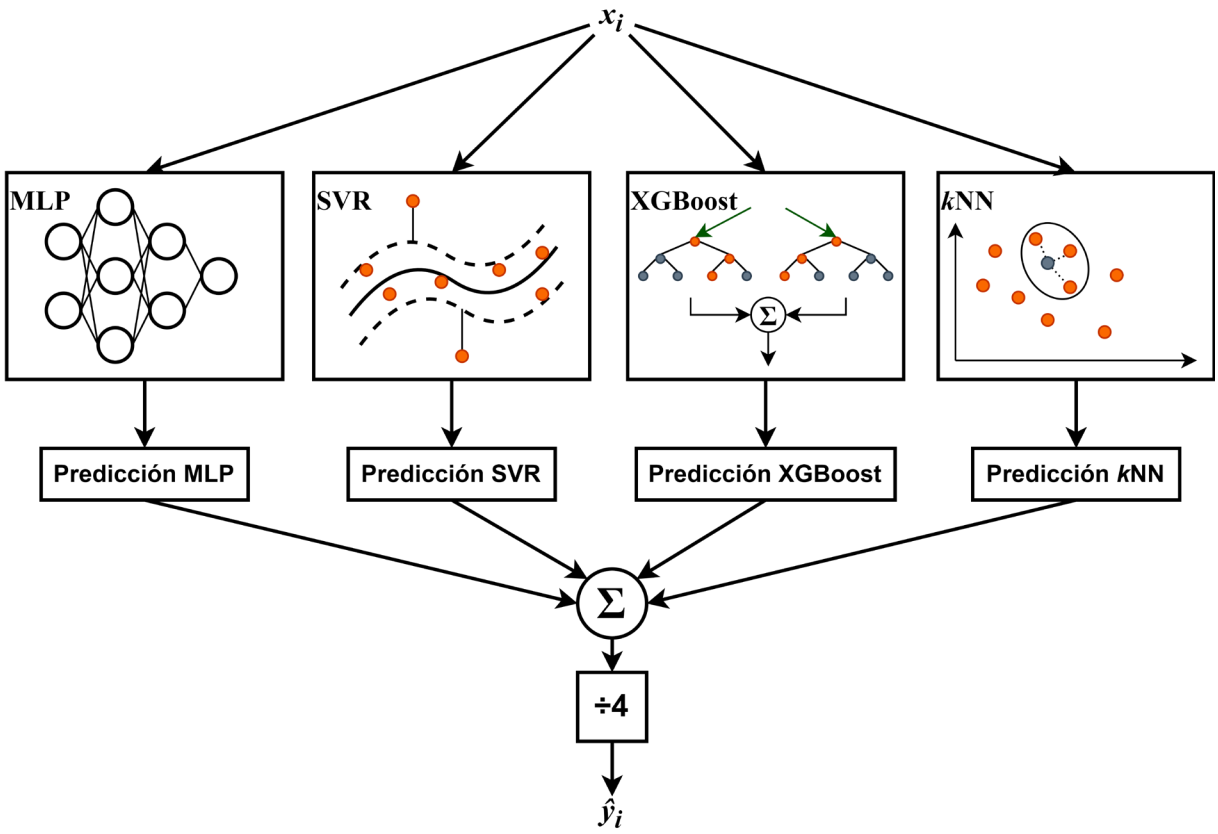

Figura 4.11: Predicción del conjunto mediante la combinación de las salidas

4.2.3.7. Aprendizaje de conjunto: Metamodelo mixto

La segunda estrategia de modelado utilizada en la metodología propuesta que busca sacar ventaja de los modelos construidos previamente es el metamodelo mixto. En esta estrategia, se entrena un modelo adicional, llamado metamodelo, para representar la relación entre las predicciones de los modelos base y los valores esperados. Generalmente, para aprender esta relación, se elige un tipo de modelo sencillo, por ejemplo, una regresión lineal. Para evitar una sobreestimación de la capacidad de generalización del metamodelo, este debe ser entrenado con datos que no hayan sido utilizados previamente ni en el entrenamiento ni en la optimización de los hiperparámetros de los modelos base. En este caso, se utiliza el subconjunto

de datos extra, conocido como conjunto de datos de retención, separado en etapas previas de la metodología. De esta forma, para crear el conjunto de datos de entrenamiento del metamodelo, para cada par de datos (X_{h_i}, y_{h_i}) en el conjunto de retención se realiza la predicción con todos los modelos base obteniéndose el vector $\hat{y}_i = [\hat{y}_{i_1}, \hat{y}_{i_2}, \dots, \hat{y}_{i_n}]$ donde n es el número de modelos base. A continuación, el conjunto de datos de entrenamiento $(\hat{y}, y_h)$ se conforma con todos los pares $(\hat{y}_i, y_{h_i})$ y se procede a entrenar el metamodelo. En el caso de una regresión lineal, el proceso de entrenamiento supone utilizar algún método, como el descenso del gradiente, para actualizar iterativamente los coeficientes de una recta $b = [b_0, b_1, \dots, b_n]$ en la dirección que se minimice una función de coste hasta cumplir con algún criterio de parada. La recta, en este caso, se describe por la siguiente ecuación:

$$\hat{y}_h = b_0 + b_1\hat{y}_1 + b_2\hat{y}_2 + \cdots + b_n\hat{y}_n \tag{29}$$

Por otra parte, la función de coste que generalmente se utiliza en este caso es el error cuadrático medio descrito en la ecuación (10).

Una vez que se han ajustado todos los coeficientes, el metamodelo está listo para ser utilizado. Para realizar predicciones con este, al igual que para crear el conjunto de datos de entrenamiento, primeramente, se deben evaluar todos los modelos base y con sus valores predichos se realiza una evaluación del metamodelo, tal como se muestra en la Figura 4.12.

4.2.4. Criterio de selección de modelos

La metodología propuesta explora varias versiones de diferentes modelos mediante la optimización de hiperparámetros. Esto supone utilizar algún criterio para poder medir cuando una combinación de hiperparámetros conduce a una mejor capacidad de generalización de los modelos. La comparación de las diferentes versiones de un modelo se lleva a cabo utilizando el conjunto de datos de validación. La métrica seleccionada para llevar a cabo la comparación fue el coeficiente de determinación (R^2). R^2 se calcula mediante la siguiente ecuación:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{30}$$

donde n es el número de muestras en el conjunto de datos de validación, y_i y $\hat{y}_i$ son los i-ésimos valores observados y predichos respectivamente, y $\bar{y}$ es el promedio de los valores observados.

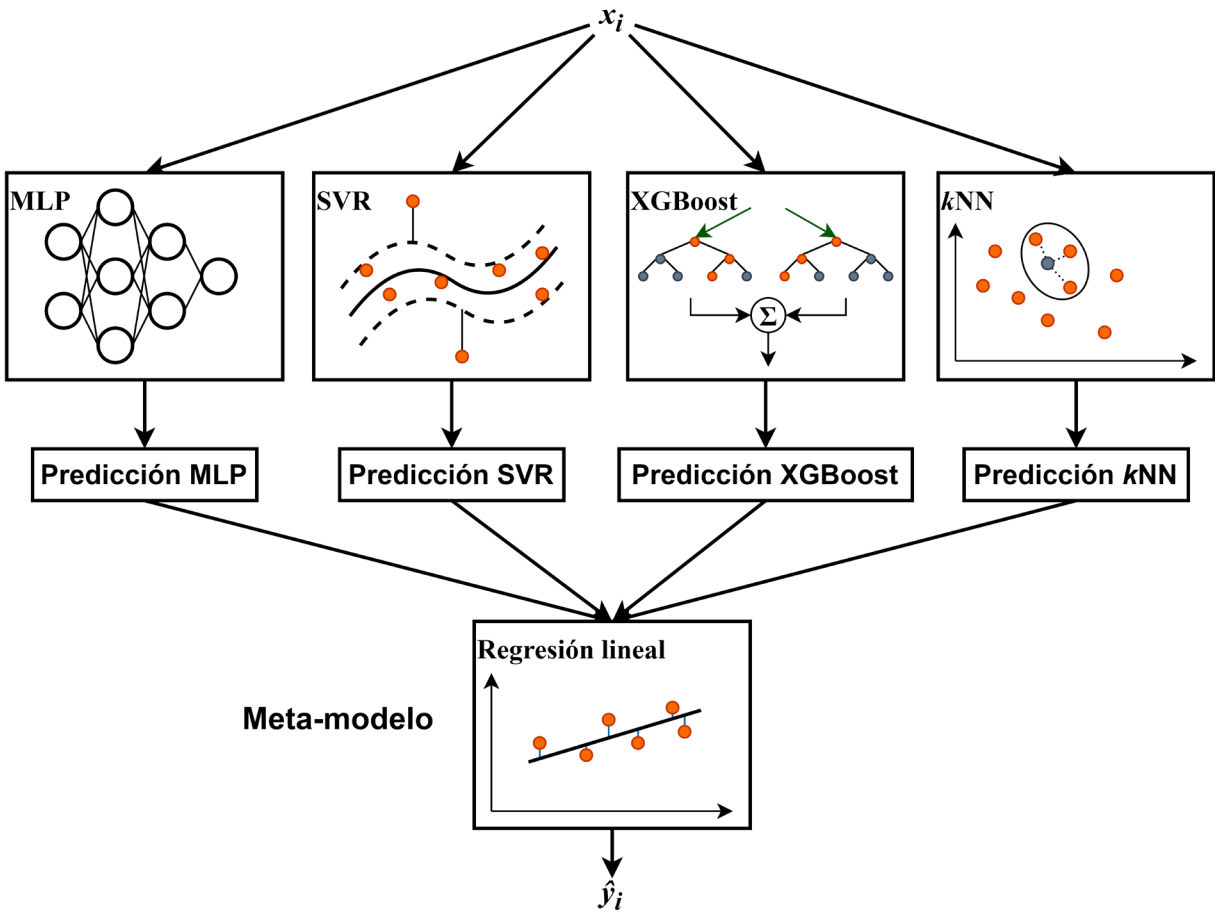

Figura 4.12: Predicción del conjunto mediante un metamodelo

La razón de escoger R^2 es que intuitivamente es más informativa que otras métricas (Chicco et al., 2021). R^2 tiene como límite superior el valor 1, el cual se alcanza si todas las predicciones son idénticas a los valores esperados. Esta propiedad de R^2 hace que sea más fácil de estandarizar y comprender el proceso de selección, ya que es un valor independiente de la escala de los datos y mientras mayor sea, mejor se puede considerar al modelo.

Mediante la comparación de los valores de R^2 obtenidos al evaluar el conjunto de datos de validación se puede seleccionar para cada tipo de modelo aquel cuyos hiperparámetros sean más adecuados. En este punto de la metodología se tienen varios modelos para cada salida, además de las técnicas de aprendizaje de conjunto. El siguiente paso, consiste en seleccionar solo un modelo o una de las técnicas de aprendizaje de conjunto para cada salida. Para esto, es necesario realizar una nueva comparación. En este caso se propone utilizar nuevamente R^2

como métrica, pero esta vez utilizando el conjunto de datos de prueba. Para seleccionar la estrategia de modelado se escoge la que mayor valor de R^2 obtenga. Además, se decidió fijar un umbral mínimo de $R^2 > 0.8$. Si ninguna de las estrategias de modelado logra superar este umbral, se finaliza la ejecución de la metodología y se emite un mensaje indicando que con los datos actuales no se ha podido obtener ningún modelo adecuado para la salida analizada. En caso contrario, se acepta la utilización del modelo como representación de la salida correspondiente del proceso productivo y se continua con la ejecución de la metodología. Esta comprobación es necesaria para que, en caso de que no se logre crear un modelo con un nivel de precisión alto, no se avance a las siguientes fases de la metodología, evitando obtener resultados no deseados.

Por otra parte, si la estrategia seleccionada es un modelo base, se puede prescindir del resto de modelos. Sin embargo, en caso de que la estrategia seleccionada sea la combinación de predicciones o del metamodelo mixto es necesario conservar todos los modelos base, aunque estos no se utilicen de manera individual.

4.2.5. Optimización del proceso productivo

El principal objetivo de la metodología de aprendizaje automático automatizado propuesta es optimizar el proceso productivo analizado, encontrando los valores de las variables de entrada de los modelos (parametrización) adecuados para que el proceso funcione de manera conveniente dadas las restricciones asociadas a determinado régimen de trabajo. Asumiendo que las etapas descritas anteriormente se ejecutaron correctamente y que se aceptó un modelo para representar cada salida ya que este superó el umbral predefinido de R^2 para el conjunto de datos de prueba, se puede iniciar el proceso de optimización de la producción.

Previamente, en las ecuaciones (1-3) se definió el problema de optimización que se debe resolver en esta etapa. A menudo, en muchos procesos productivos, la mejora de un KPI implica la degradación de otro, ya que son objetivos contrapuestos. En estos casos, no existe una solución única al proceso de optimización multiobjetivo, sino que se debe utilizar algún criterio de optimalidad, como el de Pareto, el cual es el adoptado en la metodología, para obtener un conjunto de soluciones que representen el compromiso entre los objetivos. El criterio de optimalidad de Pareto se puede enunciar de la siguiente forma: una solución $x^* \in \mathbb{R}^n$ es óptima de Pareto si y sólo si no existe otra $x \in \mathbb{R}^n$ tal que $f(x)$ domine a $f(x^*)$. Mientras tanto, el

concepto de dominancia de Pareto plantea que para que un vector $u \in \mathbb{R}^n$ domine a otro vector $v \in \mathbb{R}^n$ en un problema de minimización se deben cumplir las siguientes dos condiciones:

$$u_i \leq v_i, \qquad para\ i = 1, 2, \dots, n$$

$$u_i < v_i, \qquad para\ al\ menos\ un\ valor\ de\ i$$

Esto quiere decir que u debe mejorar alguno de los objetivos de v sin empeorar ninguno de los otros objetivos.

En el contexto de problemas de optimización multiobjetivo, existen múltiples enfoques disponibles para su resolución incluyendo algoritmos de enjambre de partículas, entropía cruzada, colonia de hormigas, recocido o temple simulado y algoritmos genéticos. En este trabajo se optó por utilizar el algoritmo *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) el cual ha sido diseñado para resolver problemas con múltiples objetivos adoptando el criterio de optimalidad de Pareto (Deb et al., 2002), aunque también se puede aplicar a problemas de optimización con un solo objetivo. La Figura 4.13 muestra una representación en diagrama de bloques de NSGA-II. Este algoritmo implementa una ordenación rápida no dominada para converger de manera eficiente hacia el frente de Pareto real y, además, emplea el concepto de distancia de aglomeración para mantener la diversidad en las soluciones, pudiendo adaptarse a problemas con frente de Pareto amplio (Yadav et al., 2022). Asimismo, NSGA-II es un algoritmo ampliamente conocido por su escalabilidad en problemas con un alto número de variables, lo cual es conveniente en el caso de la metodología propuesta ya que se debe poder tratar con un amplio rango de variables de decisión. La utilidad de NSGA-II para optimizar procesos industriales ha sido probada en múltiples aplicaciones prácticas (Mukkoti et al., 2020).

4.3. Gemelo digital

Una vez que se haya implementado la metodología diseñada en el módulo *Insight Generator*, esta podrá ser utilizada directamente o como parte de un sistema más complejo. Uno de los principales objetivos de este trabajo consiste en la integración de este módulo a un gemelo digital, tal como se planteó en el Capítulo 3. El gemelo digital que se propone desarrollar contempla tres funcionalidades: visualización, gestión y supervisión. Para la visualización es necesario crear un escenario virtual tridimensional del proceso productivo, el cual permita replicar el flujo del proceso

y que sea parametrizable. Por otra parte, la funcionalidad de gestión se debe encargar de manejar todo el flujo de información relacionado con el gemelo digital. Debido a que la comunicación con *Insight Generator* se realiza a través de Orion, es necesario que el gemelo digital sea capaz de generar e interpretar entidades NGSI-v2. Mientras tanto, la funcionalidad de supervisión, la cual se analizará con mayor profundidad en las siguientes subsecciones, es la que le debe permitir al gemelo digital coevolucionar con el proceso productivo a medida que este va sufriendo cambios.

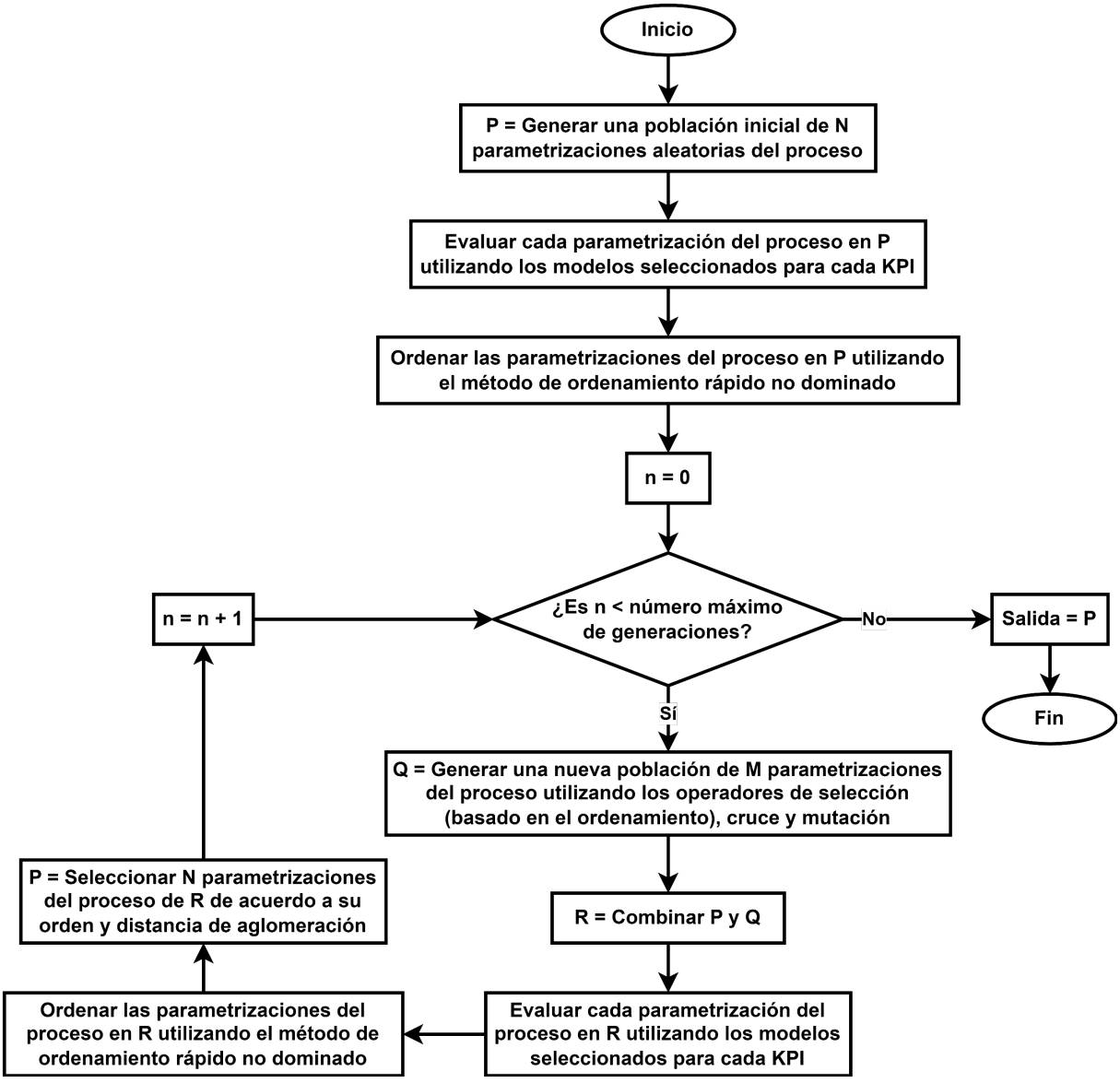

Figura 4.13: Diagrama de bloques de NSGA-II

4.3.1. Supervisión

4.3.1.1. Deriva de datos y de concepto

Una vez que se ha superado exitosamente el proceso de entrenamiento y evaluación de un modelo de aprendizaje automático se puede considerar que este ha sido capaz de aprender las relaciones subyacentes entre las entradas y salidas del proceso representado y que, por tanto, ha adquirido una capacidad de generalización que le permite realizar predicciones acertadas ante casos a los que el modelo no ha sido expuesto con anterioridad. Sin embargo, existen situaciones en las que el desempeño de los modelos puede verse afectado. Esto se debe generalmente a dos fenómenos: la deriva de datos y la deriva de concepto.

La deriva de datos[11] se refiere a la diferencia que existe entre la distribución de los datos de las variables de entrada en el conjunto de datos utilizado para crear el modelo y los nuevos datos provenientes del proceso, una vez que el modelo está en explotación. En el caso de los modelos creados con *Insight Generator* este fenómeno puede ocurrir si en el proceso se utilizan combinaciones de parámetros que no se encuentran presentes en el conjunto de datos con el que se crearon los modelos o si se fijan valores de los parámetros fuera del rango establecido. Si el conjunto de datos utilizado para crear el modelo tiene una amplia cobertura del espacio dimensional de las variables de entrada, se espera que al procesar nuevos datos con otra distribución se obtengan resultados aceptables; sin embargo, no existe ninguna garantía de que esto será así, pudiendo ocurrir una degradación en el desempeño.

Por otra parte, la deriva de concepto se refiere a los cambios que pueden ocurrir en las relaciones subyacentes entre las entradas y salidas del proceso a medida que pasa el tiempo, lo que provoca que, para un mismo valor de las entradas en dos instantes diferentes, la salida sea notablemente distinta. Este fenómeno se puede representar de la siguiente forma:

$$P_{t_1}(y|X) \neq P_{t_2}(y|X) \tag{31}$$

donde X contiene los valores de las variables de entrada, y contiene los valores de la variable de salida y P_{t_1} y P_{t_2} son las distribuciones resultantes en los instantes t_1 y t_2.

[11] Se ha adoptado el término deriva de datos del inglés *data drift*.

De esta forma, si durante la creación de un modelo este aprendió a representar las relaciones entre las entradas y la salida, a partir de datos recolectados en un período de tiempo, y estas relaciones cambian posteriormente, la representación aprendida por el modelo dejaría de ser válida, por lo que las predicciones serían incorrectas. El cambio en las relaciones entre las entradas y la salida puede ocurrir fundamentalmente de tres maneras: repentinamente, de manera gradual o de manera incremental. La Figura 4.14 muestra una representación de las formas en que se produce la deriva de concepto. En este caso el valor de las entradas se mantiene constante, sin embargo, la distribución en los datos de la salida varía con el tiempo.

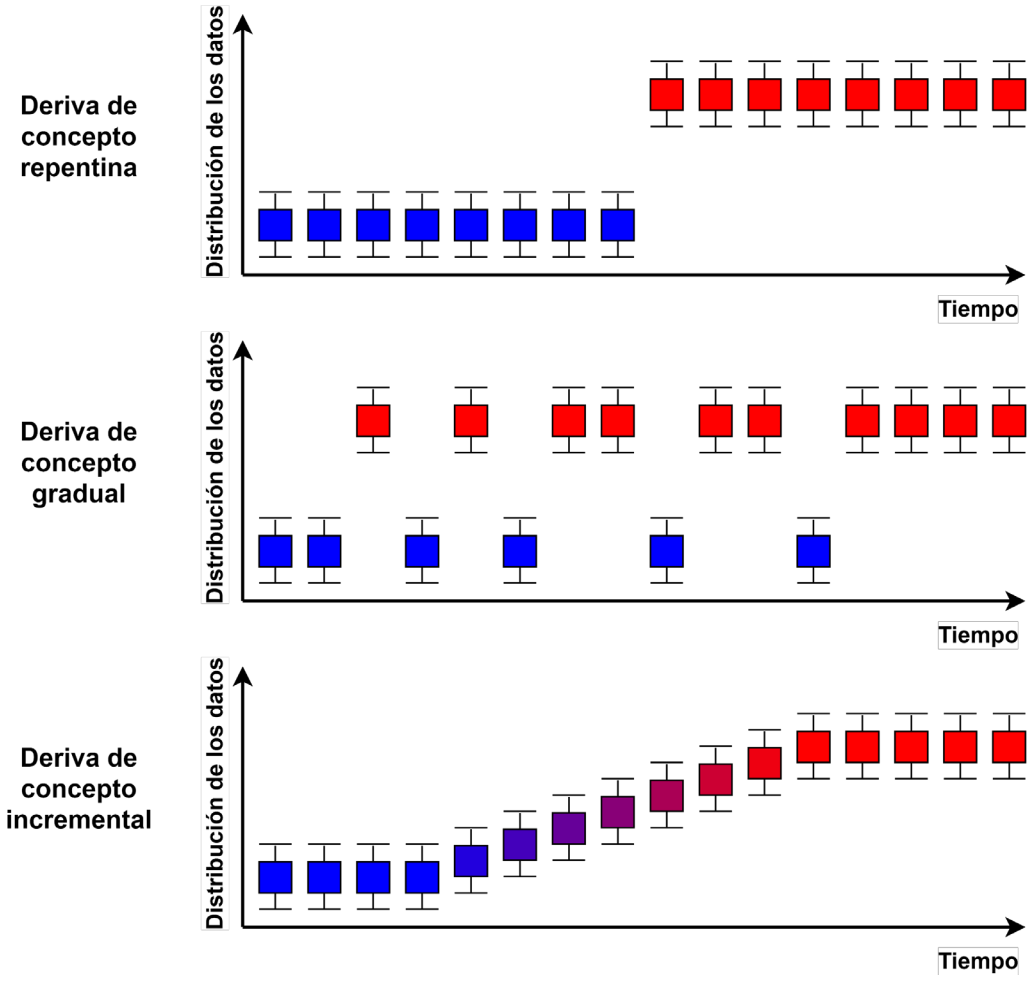

Figura 4.14: Tipos de deriva de concepto

4.3.1.2. Detección de las derivas de datos y de concepto

La detección de derivas de datos o de concepto es fundamental para asegurar que el gemelo digital funcione tal como se espera. Si estos fenómenos no son tomados en cuenta para decidir cuándo reentrenar los modelos, pueden ocurrir cambios en el proceso productivo los cuales nunca serán reflejados por el gemelo digital, conduciendo a predicciones erróneas y, en última instancia, a la toma de decisiones basada en premisas incorrectas.

En el caso de la deriva de datos puede darse la situación de que, a pesar de la ocurrencia de esta, el rendimiento de los modelos no se vea afectado debido a que son capaces de generalizar correctamente ante los nuevos datos. Ante esta posibilidad, se ha decidido no supervisar directamente los datos de entrada, sino que ante los cambios en los datos de entrada se debe supervisar que los valores de salida medidos en el proceso productivo se asemejan a las predicciones de los modelos. De esta forma, si no hay discrepancias significativas entre estos valores no se requeriría ninguna acción y el gemelo digital podría continuar trabajando con los modelos sin necesidad de modificarlos. En el caso contrario, si existen diferencias no despreciables entre las predicciones y los valores de salida medidos, se requeriría reentrenar los modelos con los datos actualizados.

En el caso de la deriva de concepto, esta generalmente se detecta supervisando los datos de salida. Si no se han producido cambios en las entradas se espera que los datos de salida se mantengan relativamente estables. En el caso de que comiencen a notarse cambios en la salida sin que se hayan producido modificaciones en las entradas este es un síntoma de que ha ocurrido una deriva de concepto y, por tanto, los modelos deben reentrenarse con datos actualizados. En este caso, se debe tener en cuenta que en los procesos productivos pueden ocurrir perturbaciones las cuales pueden verse reflejadas momentáneamente en los valores de salida medidos sin que esto implique que ha ocurrido una deriva de concepto. Por tanto, la estrategia de supervisión debe ser capaz de lidiar con esta situación sin desencadenar un reentrenamiento de los modelos.

Para la detección de las derivas tanto de datos como de concepto se decidió utilizar el método *Adaptive Windowing* (ADWIN). Este utiliza una ventana W de longitud variable, la cual se mantiene actualizada con los datos más recientes para buscar cambios en su distribución (Bifet & Gavaldà, 2007). Al recibir un nuevo dato este se agrega a W. Seguidamente comienza un proceso iterativo en el cual W se subdivide en dos ventanas cada una con una longitud mínima n_{min}, las cuales son

utilizadas para determinar si ha ocurrido una deriva: W_0 (con los datos más antiguos) y W_1 (con los datos más actuales). Estas ventanas tienen longitud $n_0 \geq n_{min}$ y $n_1 \geq n_{min}$, respectivamente, la cual varía en cada iteración, tal como se muestra en la Figura 4.15.

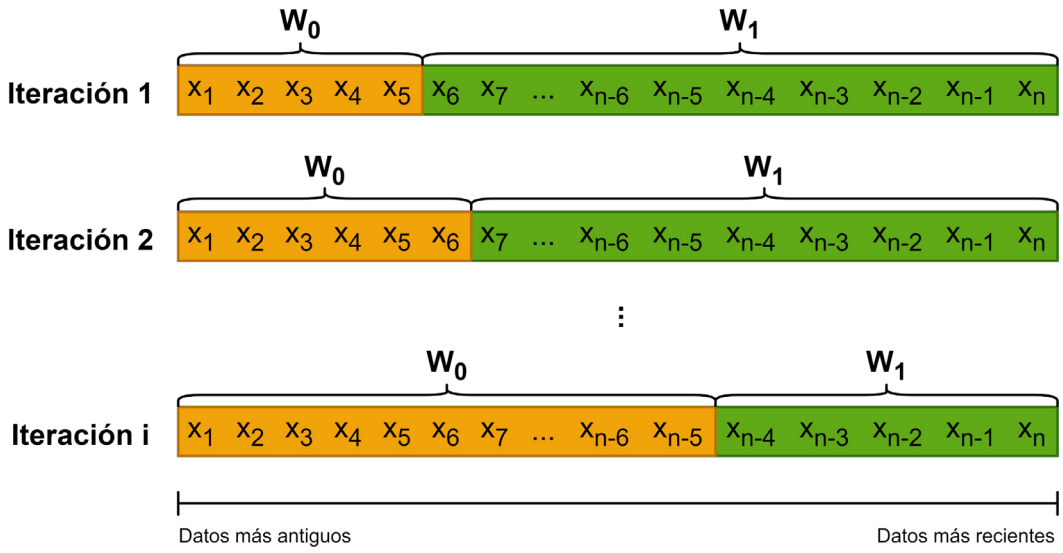

Figura 4.15: Ejemplo de subdivisiones de W analizadas por ADWIN al recibir un nuevo dato para $n_{min} = 5$

ADWIN considera que existe una deriva cuando se cumple la siguiente condición:

$$\left|\mu_{W_0} - \mu_{W_1}\right| > \epsilon \tag{32}$$

donde μ_{W_0} y μ_{W_1} son los promedios de los datos en las ventanas W_0 y W_1, respectivamente, y ϵ es un umbral que se calcula mediante la siguiente ecuación:

$$\epsilon = \sqrt{\frac{2}{m} \cdot \sigma_W^2 \cdot ln\frac{2n}{\delta}} + \frac{2}{3m} \cdot ln\frac{2n}{\delta} \tag{33}$$

donde σ_W^2 es la varianza de los elementos en W, $n = n_0 + n_1$ es la longitud de esta ventana, $\delta \in (0,1)$ es un parámetro que permite controlar la confiabilidad de los resultados del algoritmo y m se calcula de la siguiente forma:

$$m = \frac{1}{\frac{1}{n_0} + \frac{1}{n_1}} \tag{34}$$

Si tras haber culminado todas las iteraciones ADWIN no ha detectado cambios significativos, entonces W se reduce con probabilidad δ y se queda a la espera de recibir nuevos datos. Si, por el contrario, en cualquier momento se detecta algún cambio significativo, este se informa y ADWIN reduce la ventana W a W_1, descartando los datos en W_0 con probabilidad $1 - \delta$. A continuación, se pueden continuar recibiendo datos. ADWIN puede ser implementado de forma tal que se utilicen histogramas exponenciales como estructura de datos básica haciendo que el algoritmo sea más eficiente (Grulich et al., 2018).

ADWIN se puede utilizar directamente para detectar las derivas de concepto en el gemelo digital y, en caso de detectar alguna, se procedería a reentrenar los modelos. Mientras tanto, la propuesta de utilizar ADWIN para detectar las derivas de datos implica realizar una modificación simple. Cada vez que se cambie de régimen de trabajo, se reiniciará el algoritmo para evitar que la variación esperada en la salida debido al cambio de régimen sea detectada como una deriva. Al reiniciar el algoritmo se realizará una predicción del valor de salida para el nuevo régimen y con él se generará un vector que servirá para simular que se ha estado obteniendo durante determinado período esta medición del proceso productivo para, a continuación, comenzar a añadir los datos reales que se vayan midiendo. De esta manera, si ADWIN detecta una discrepancia, significa que los modelos no son capaces de generalizar correctamente para el nuevo régimen de trabajo y se deben reentrenar. Por el contrario, si no se detecta ninguna discrepancia, aunque pueda haber ocurrido una deriva de datos y el proceso esté trabajando en un régimen completamente nuevo, no es necesario llevar a cabo ninguna acción ya que los modelos presentan un buen desempeño.

4.4. Procedimiento de implementación

Las propuestas presentadas en este capítulo, tanto para la metodología de aprendizaje automático automatizado como para las funcionalidades del gemelo digital, requieren la utilización de diversos algoritmos y técnicas. Teniendo en cuenta que el lenguaje de programación Python cuenta con una amplia gama de bibliotecas, las cuales implementan de manera eficiente muchos de los algoritmos necesarios cuyo funcionamiento ha sido ampliamente validado, se optó por usar este lenguaje y varias de sus bibliotecas como base para desarrollar la propuesta en lugar de implementarla desde cero, tal y como se refleja en la Figura 4.16.

Para la gestión y el preprocesamiento de los datos en *Insight Generator* se emplearon las bibliotecas Numpy y scikit-learn. Numpy es la principal biblioteca para la manipulación y operación con datos en vectores, matrices y arreglos multidimensionales en Python (Harris et al., 2020). Por su parte, scikit-learn proporciona herramientas eficientes para el preprocesamiento de datos y la selección de variables (Pedregosa et al., 2011).

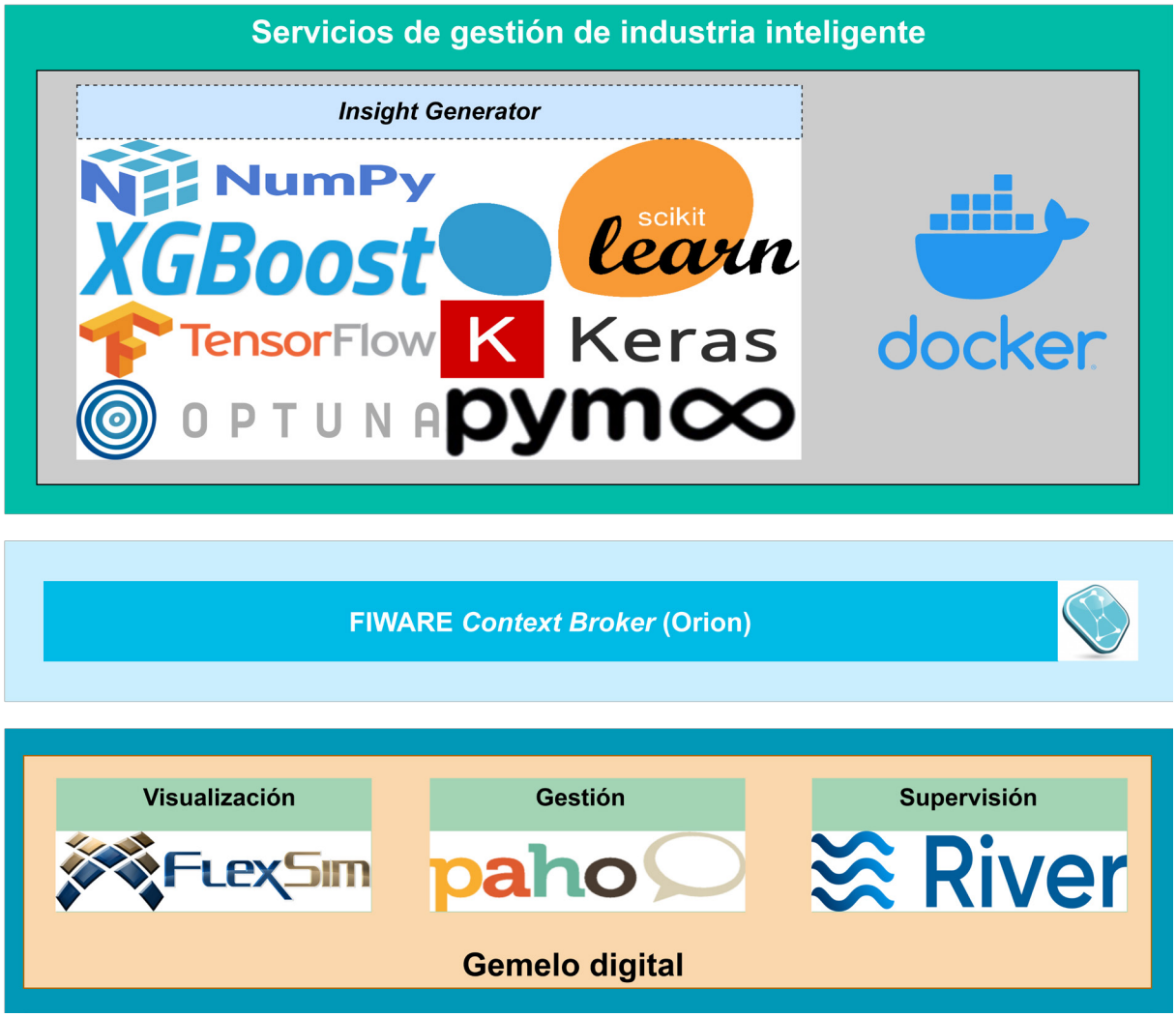

Figura 4.16: Herramientas y bibliotecas utilizadas en la implementación de las propuestas relacionadas con el método de reparametrización óptima y el gemelo digital

En la creación de los modelos también se utilizó scikit-learn, la cual proporciona una interfaz sencilla para el entrenamiento y evaluación de diversos algoritmos de aprendizaje automático, entre ellos SVR y *k*NN. Otra biblioteca utilizada en la

creación de los modelos fue XGBoost debido a su implementación altamente eficiente y flexible del método de refuerzo del gradiente (T. Chen & Guestrin, 2016). Por último, para el trabajo con MLPs se utilizaron las bibliotecas Tensorflow y Keras. Tensorflow es una de las bibliotecas de aprendizaje automático más utilizadas actualmente, principalmente para desarrollar soluciones basadas en redes neuronales. Su principal característica consiste en la representación de los cálculos como grafos computacionales, lo que unido a su soporte multiplataforma y capacidades de paralelismo hacen que su rendimiento sea eficiente, incluso en proyectos a gran escala (Martín Abadi et al., 2015). Por su parte, Keras proporciona una interfaz de alto nivel para la construcción, entrenamiento y evaluación de modelos de aprendizaje profundo. Aunque originalmente se desarrolló de manera independiente, en la actualidad es la interfaz nativa de alto nivel de Tensorflow, simplificando y acelerando el proceso de desarrollo al trabajar con esta biblioteca.

La optimización de hiperparámetros de los modelos incluidos en *Insight Generator* se realizó utilizando la biblioteca Optuna, la cual se especializa en esta tarea. Optuna implementa, entre otros, el método TPE, el cual permite construir el espacio de búsqueda de los hiperparámetros de manera dinámica y, además, emplea estrategias de poda para agilizar su exploración (Akiba et al., 2019). Finalmente, para la obtención de las parametrizaciones óptimas de los procesos productivos en *Insight Generator* se utilizó pymoo. Esta es una biblioteca de Python especializada en la resolución de problemas de optimización multiobjetivo principalmente mediante el uso de algoritmos genéticos (Blank & Deb, 2020).

El módulo *Insight Generator* ha sido encapsulado en un contenedor creado con Docker. Docker permite empaquetar en unidades estandarizadas todos los elementos necesarios para que un software pueda ejecutarse, incluyendo bibliotecas, herramientas del sistema y código, entre otros (Y. Zhang et al., 2022). Gracias a esta capacidad de Docker, el módulo puede funcionar de manera independiente al entorno de ejecución, lo que facilita su portabilidad y simplifica el proceso de transferencia de equipos de desarrollo a plataformas de producción.

Tal como se detalló en la propuesta, la capa intermediaria entre *Insight Generator* y los elementos de las capas inferiores está basada en Orion. Mientras tanto, para la implementación de las distintas funcionalidades del gemelo digital se utilizaron varias herramientas y bibliotecas. En el caso de la visualización se utilizó FlexSim, el cual es un software de simulación tridimensional de procesos y sistemas. FlexSim tiene una interfaz intuitiva, con un amplio número de componentes industriales predefinidos, por lo que recrear un escenario de producción es

relativamente sencillo. Sin embargo, admite un alto nivel de personalización por lo que se pueden crear simulaciones complejas y avanzadas. FlexSim además soporta la integración con otros sistemas para recibir datos y por lo que puede actualizar la simulación con información en tiempo real. En el caso de la funcionalidad de supervisión, esta se implementó utilizando la biblioteca River. River se especializa en el aprendizaje automático en línea, para lo cual utiliza algoritmos que permiten el análisis de flujos de datos, como ADWIN (Montiel et al., 2020). Esta característica es de vital importancia en la mayoría de los escenarios reales de producción ya que los datos se generan gradualmente a medida que pasa el tiempo. Mediante el análisis de los flujos de datos se puede evaluar de forma continua la adecuación del modelo, lo que permite tomar decisiones informadas sobre cuándo realizar alguna modificación. Por último, la funcionalidad de gestión implica, entre otras cosas, comunicarse con los elementos del proceso productivo, para lo cual se utilizó la biblioteca paho-MQTT. Esta biblioteca proporciona la implementación de un cliente del protocolo *Message Queuing Telemetry Transport* (MQTT), el cual se puede conectar a un *broker* para publicar mensajes y suscribirse a diferentes asuntos para recibir los mensajes relacionados en una red de nodos IoT.

4.5. Conclusiones

En este capítulo se presentó la metodología para la reparametrización óptima de los procesos productivos basada en aprendizaje automático automatizado, la cual se implementará en un módulo de la plataforma KITT4SME. Se detallaron las diferentes etapas que forman parte de esta metodología, en la cual inicialmente se lleva a cabo un preprocesamiento de los datos de manera que se eliminen inconsistencias, asegurando la integridad de la información. Seguidamente, se contempla una etapa de análisis sistémico y de variables, la cual se centra en la selección de las variables con mayor capacidad predictiva determinada mediante métodos de filtrado basados en criterios estadísticos. La siguiente etapa se encarga de la creación de los modelos para representar los KPIs deseados del proceso productivo analizado y la optimización de sus hiperparámetros. Los diferentes modelos que se consideran son MLP, SVR, XGBoost y *k*NN, los cuales han sido escogidos para asegurar la diversidad entre los enfoques utilizados. Mientras que, para la optimización de hiperparámetros de todos estos modelos se utiliza TPE, un algoritmo de optimización bayesiana cuya eficacia para esta tarea ha sido ampliamente comprobada. Adicionalmente, a los modelos base considerados

también se examinan dos estrategias de aprendizaje de conjunto, una en la que se promedian los resultados de los modelos base y la otra en la que se utiliza un metamodelo para, a partir de los resultados de los modelos base, realizar las predicciones. Tras evaluar los modelos base y las dos estrategias de aprendizaje de conjunto se selecciona la mejor opción en base a la métrica R^2. En la siguiente etapa se utilizan los modelos seleccionados como parte de un problema de optimización multiobjetivo, de manera que al resolverlo se obtiene un conjunto de parametrizaciones óptimas para el proceso productivo.

Por otra parte, en este capítulo también se describieron las distintas funcionalidades que deberá poseer el gemelo digital que se creará para la línea piloto de GAMHE, profundizándose en la capacidad de supervisión, la cual se basa en el algoritmo ADWIN, el cual fue diseñado para el análisis de flujos de datos. Se presentaron las diferentes situaciones prácticas que pueden afectar el desempeño de los modelos creados y se analizó cómo la funcionalidad de supervisión es la que le facilitará al gemelo digital lidiar con ellas, coevolucionando con el proceso productivo. Por último, se presentaron las diferentes herramientas que se utilizarán para la implementación de cada uno de los aspectos tratados en el capítulo.

5. Metodología para la reconfiguración automática e inteligente

5.1. Introducción

En el capítulo previo se describió y analizó el primer módulo basado en un método de IA propuesto para integrar en la arquitectura KITT4SME, así como su utilización para el desarrollo de sistemas más complejos como el gemelo digital de un proceso productivo. En este capítulo se aborda la metodología del segundo módulo propuesto que será integrado en la arquitectura KITT4SME, que consiste en la reconfiguración automática e inteligente para la corrección de desviaciones en la productividad de las máquinas/robots y dispositivos del sistema de producción: *Reco*. Este problema, común en entornos industriales, muchas veces impide cumplir los objetivos de producción y los tiempos de entrega. La metodología presentada está soportada en un sistema de inferencia borrosa y ha sido concebida de tal manera que pueda ser aplicada de forma genérica a una amplia gama de elementos de las líneas de producción. Además, se presenta el procedimiento de implementación del módulo *Reco*.

5.2. Método propuesto

Una vez que se han parametrizado los diferentes elementos de un sistema productivo se espera que estos trabajen en el régimen previsto. Sin embargo, en la práctica, los equipos pueden verse afectados por perturbaciones externas o, debido al paso del tiempo, sufrir degradaciones que afecten a su dinámica interna. Ante estas situaciones, se producen desviaciones con respecto al comportamiento esperado de los elementos del sistema productivo. Aunque, en un entorno productivo con un alto nivel tecnológico se espera que la mayoría de los equipos cuenten con mecanismos para corregir estas desviaciones, en los procesos productivos con un nivel tecnológico medio/bajo esto no se cumple siempre. En estos últimos, generalmente, se fijan los parámetros y se dejan funcionar los equipos en lazo abierto. Como resultado, en muchas ocasiones se producen desviaciones las cuales no son corregidas y solo son atendidas si se producen situaciones problemáticas evidentes.

Para hacer frente a las situaciones descritas previamente se ha desarrollado una metodología genérica que utiliza un sistema de inferencia borrosa y que permite, a partir de los valores esperados y reales de un indicador de productividad de los elementos de una línea productiva, determinar si es necesario llevar a cabo alguna acción correctiva. Dichas acciones podrán ser presentadas como recomendaciones a los encargados de planta o directamente aplicadas a los elementos del proceso productivo. Una de las ventajas de esta metodología es que puede ser aplicada tanto de manera individual a los equipos como a celdas compuestas por varios dispositivos dentro de un sistema productivo. Además, al poder ser aplicada de manera individual, si algún equipo implementa su propio mecanismo de corrección, esto no impide que el resto de los equipos puedan hacer uso de la metodología.

Las ventajas de los sistemas de inferencia borrosa en relación con la robustez y la explicabilidad de las soluciones, específicamente de los sistemas borrosos tipo Mamdani, ha sido demostrada en la literatura científica en las últimas décadas en múltiples aplicaciones que utilizan esta tecnología (Haber et al., 2005, 2006). Los resultados en cuestiones relevantes para las aplicaciones industriales, como el análisis dinámico, han corroborado la idoneidad y utilidad de estos sistemas (Haber et al., 2003).

En la metodología se decidió utilizar como indicador de la productividad de los equipos el tiempo en el que procesan una unidad del producto. De esta forma, se definen dos entradas: el tiempo nominal en el que se espera que el equipo analizado procese una unidad del producto (T_a) y el tiempo real que le toma al equipo procesar una unidad del producto (R_a). Mientras que el valor de la primera entrada se define al iniciar la producción o al cambiar el régimen de trabajo, el valor de la segunda variable debe ser actualizado cada vez que se procese un producto. Una vez que se tiene esta información, de manera interna la metodología se encarga de establecer los valores de dos variables que serán las que se utilizarán como entradas al sistema de inferencia borrosa. Estas dos variables son la desviación del tiempo de producción en el instante actual (ΔT_a) y la desviación del tiempo de producción en el instante anterior (ΔT_p), las cuales se definen de la siguiente forma:

$$\Delta T_a(k) = \begin{cases} -1, & si\ R_a(k) \geq 2T_a(k) \\ \dfrac{T_a(k) - R_a(k)}{T_a(k)}, & si\ 0 \leq R_a(k) < 2T_a(k) \end{cases} \tag{35}$$

$$\Delta T_p(k) = \begin{cases} 0, & si\ k = 0 \\ \Delta T_a(k-1), & en\ cualquier\ otro\ caso \end{cases} \tag{36}$$

donde k representa el instante de tiempo correspondiente al producto que se encuentra procesando actualmente el equipo. De esta forma, se asegura que las variables de entrada al sistema de inferencia queden normalizadas al rango [-1, 1], lo que permite que la metodología pueda ser aplicada de forma genérica pues no depende de la base de tiempo específica de cada equipo del proceso productivo. Al rango de valores que puede tomar una variable difusa se le conoce como el universo de discurso de la variable.

Por otra parte, en la metodología se contempla una variable de salida: el ritmo de trabajo recomendado para el equipo ($RECO_1$). Para determinar su valor la metodología utiliza un sistema de inferencia borrosa. Esta ha demostrado ser una herramienta poderosa en la toma de decisiones en situaciones complejas y ambiguas gracias a su capacidad para tratar con la incertidumbre o inexactitud en los datos. A diferencia de la lógica binaria donde una variable solo puede tener uno de los dos estados posibles (verdadero o falso), en la Lógica Borrosa, las variables pueden tomar valores en el rango continuo [0, 1]. Gracias a ello, se puede representar el grado de pertenencia, lo cual es de gran utilidad para expresar conocimiento humano subjetivo.

Específicamente, en la metodología de reconfiguración automática e inteligente del sistema productivo se utiliza un sistema de inferencia borroso tipo Mamdani. La efectividad de este tipo de sistemas en aplicaciones de toma de decisiones y control ha sido ampliamente probada. Al trabajar con reglas lingüísticas y términos borrosos para representar de forma natural el conocimiento humano, los sistemas de inferencia borrosos tipo Mamdani facilitan la colaboración en la etapa de creación y la interpretación de las decisiones que se tomen una vez que se comiencen a utilizar. Estos sistemas se dividen en tres etapas principales: borrosificación, inferencia y desborrosificación.

5.2.1. Conjuntos borrosos de las variables de entrada y borrosificación

El primer paso para poder utilizar un sistema de inferencia borroso consiste en la borrosificación de las variables de entrada. En este proceso, se traducen los valores exactos de las entradas en valores borrosos, es decir, a partir de los datos de entrada se realiza un mapeo de estos en términos borrosos. Este mapeo se realiza asignando a los datos de entrada un grado de pertenencia a diferentes conjuntos borrosos representados con términos lingüísticos. En el caso de la metodología

propuesta se definen los siguientes términos lingüísticos para los conjuntos borrosos de las variables de entrada al sistema de inferencia borroso:

ΔT_a: $Demasiado\ retardo, Retardo, Normal, Antelación, Demasiada\ antelación$

ΔT_p: $Retardo, Normal, Antelación$

La Figura 5.1 muestra gráficamente las funciones de pertenencia asociadas a los conjuntos borrosos de ΔT_a, mientras que la Figura 5.2 muestra gráficamente las funciones de pertenencia asociadas a los conjuntos borrosos de ΔT_p. En este caso se han utilizado funciones de pertenencia tanto triangulares como trapezoidales. Utilizando estas funciones se puede determinar el grado de pertenencia a cada uno de los conjuntos definidos para la variable. Por ejemplo, si se tiene $\Delta T_a = 0.07$ se puede determinar que este tiene un grado de pertenencia de 0.3 para el conjunto $Normal$, de 0.7 para el conjunto $Antelación$ y de 0 para los conjuntos $Demasiado\ retardo$, $Retardo$ y $Demasiada\ antelación$, tal como se muestra en la Figura 5.3.

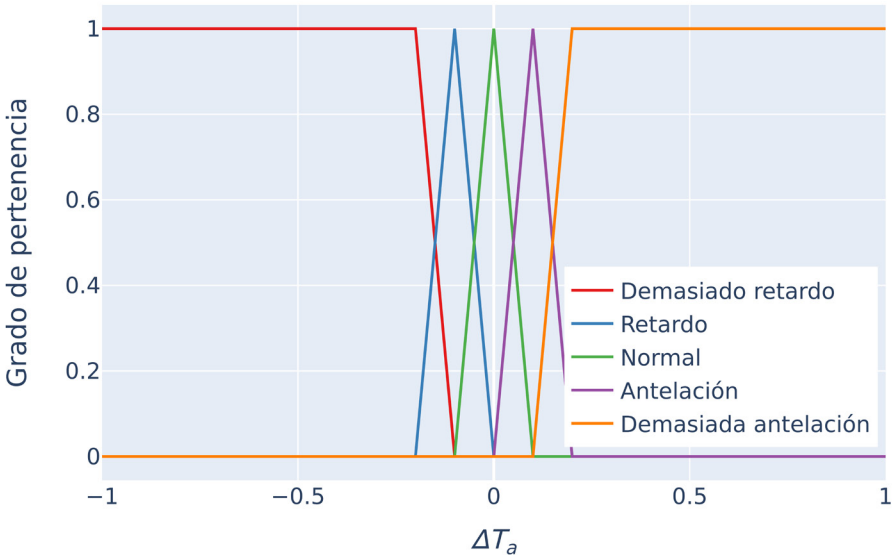

Figura 5.1: Funciones de pertenencia para los conjuntos borrosos de la variable ΔT_a

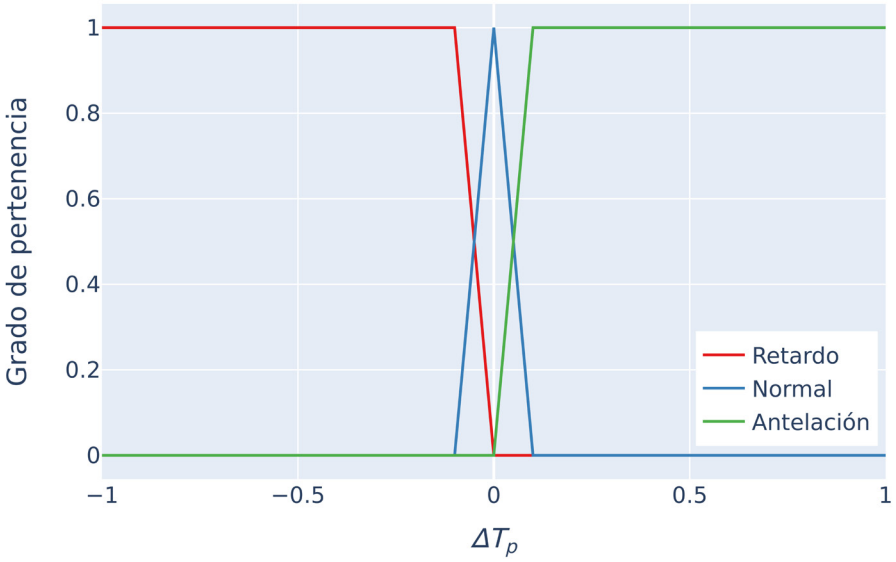

Figura 5.2: Funciones de pertenencia para los conjuntos borrosos de la variable ΔT_p

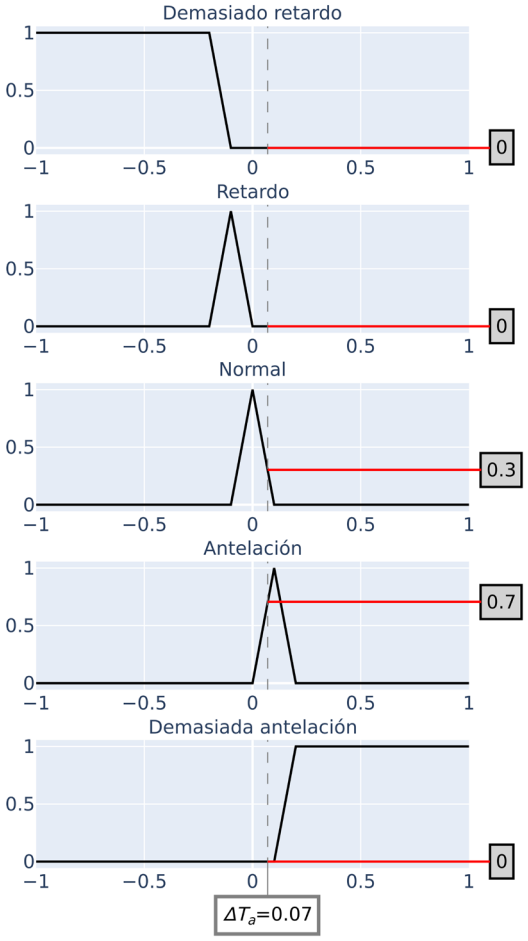

Figura 5.3: Ejemplo de determinación del grado de pertenencia a los diferentes conjuntos borrosos de la variable ΔT_a

5.2.2. Conjuntos borrosos de la variable de salida

De la misma manera que para utilizar un sistema de inferencia borroso se necesita traducir los valores numéricos concretos de las entradas a valores borrosos tomando como referencia los conjuntos borrosos, también se deben definir conjuntos borrosos para las variables de salida mediante las funciones de pertenencia correspondientes. Así, una vez que el sistema infiera una salida difusa, esta información se puede transformar a un valor numérico concreto mediante el proceso de desborrosificación. Los términos lingüísticos definidos para los conjuntos borrosos de la variable de salida son los siguientes:

$RECO_1$: $Parar\ 1, Reducir, Mantener, Aumentar, Parar\ 2$

La Figura 5.4 muestra las funciones de pertenencia definidas para los conjuntos borrosos de $RECO_1$.

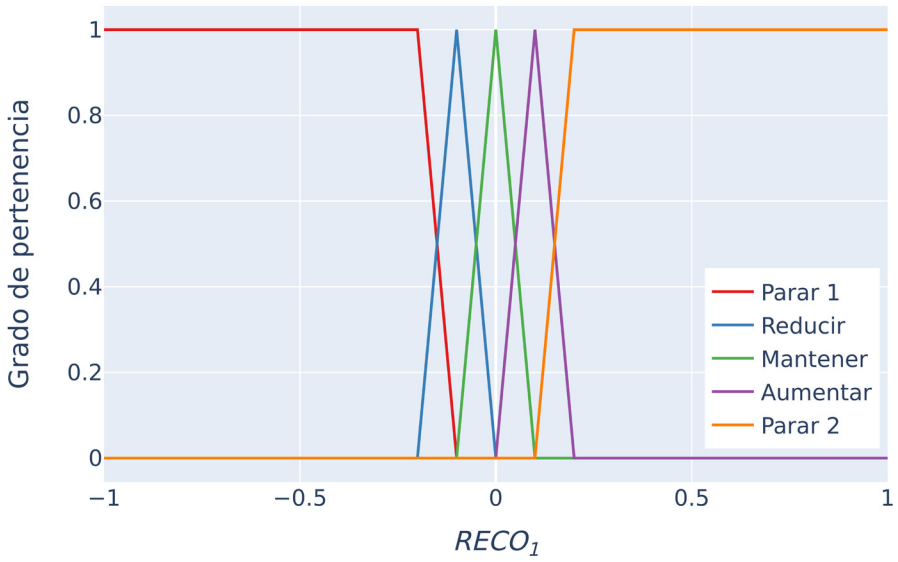

Figura 5.4: Funciones de pertenencia para los conjuntos borrosos de la variable $RECO_1$

5.2.3. Inferencia

Una vez que se han definido las funciones de pertenencia para las entradas y la salida, es necesario establecer la forma de relacionarlas. Para poder procesar la

información en un sistema borroso tipo Mamdani se necesita contar con dos elementos: una base de reglas y un mecanismo de inferencia.

5.2.3.1. Base de reglas

La base de reglas consiste en un conjunto de reglas lingüísticas en la forma "SI (antecedentes) ENTONCES (consecuente)". Cada una de estas reglas representa el vínculo que existe entre una o más variables de entrada con una variable de salida, utilizando términos lingüísticos borrosos. La base de reglas puede ser creada mediante técnicas de verbalización o identificación/entrenamiento. La verbalización supone formular las reglas mediante una descripción verbal del proceso analizado por parte de algún experto. La principal ventaja de este enfoque es que se garantiza la explicabilidad y transparencia del sistema. Por otra parte, el enfoque de identificación/entrenamiento implica la búsqueda de patrones en las relaciones entre entradas y salidas a partir de datos experimentales recogidos durante la operación manual por parte de algún experto del proceso analizado. Esta técnica se centra en la objetividad al basarse en datos medidos directamente o en patrones encontrados tras procesar ficheros de datos con la información recolectada durante la operación manual.

En este caso, la metodología pretende ser aplicable de manera genérica en diferentes procesos productivos. Por tanto, como no se cuenta con datos específicos de ningún proceso y, además, se desea priorizar la explicabilidad, se toma la decisión de crear la base de reglas mediante verbalización. A continuación, se muestran las reglas definidas para la salida $RECO_1$:

1. $SI\ \Delta T_a\ es\ Demasiado\ retardo\ ENTONCES\ RECO_1\ es\ Parar\ 1$
2. $SI\ \Delta T_a\ es\ Retardo\ Y\ \Delta T_p\ es\ Retardo\ ENTONCES\ RECO_1\ es\ Aumentar$
3. $SI\ \Delta T_a\ es\ Retardo\ Y\ \Delta T_p\ es\ Normal\ ENTONCES\ RECO_1\ es\ Mantener$
4. $SI\ \Delta T_a\ es\ Retardo\ Y\ \Delta T_p\ es\ Antelación\ ENTONCES\ RECO_1\ es\ Aumentar$
5. $SI\ \Delta T_a\ es\ Normal\ Y\ \Delta T_p\ es\ Retardo\ ENTONCES\ RECO_1\ es\ Mantener$
6. $SI\ \Delta T_a\ es\ Normal\ Y\ \Delta T_p\ es\ Normal\ ENTONCES\ RECO_1\ es\ Mantener$
7. $SI\ \Delta T_a\ es\ Normal\ Y\ \Delta T_p\ es\ Antelación\ ENTONCES\ RECO_1\ es\ Mantener$
8. $SI\ \Delta T_a\ es\ Antelación\ Y\ \Delta T_p\ es\ Retardo\ ENTONCES\ RECO_1\ es\ Reducir$
9. $SI\ \Delta T_a\ es\ Antelación\ Y\ \Delta T_p\ es\ Normal\ ENTONCES\ RECO_1\ es\ Mantener$
10. $SI\ \Delta T_a\ es\ Antelación\ Y\ \Delta T_p\ es\ Antelación\ ENTONCES\ RECO_1\ es\ Reducir$
11. $SI\ \Delta T_a\ es\ Demasiada\ antelación\ ENTONCES\ RECO_1\ es\ Parar\ 2$

5.2.3.2. Mecanismo de inferencia

El mecanismo de inferencia se utiliza para evaluar y combinar las reglas lingüísticas, teniendo en cuenta las entradas del sistema borrosificadas en la etapa anterior. Primeramente, se debe evaluar el antecedente de cada regla para determinar en qué medida esta es aplicable en función de la coincidencia entre los términos borrosos de las entradas y el antecedente. En presencia de reglas con múltiples antecedentes, estos estarán vinculados mediante operadores lógicos, generalmente "Y" u "O". En el caso de la metodología propuesta, los antecedentes de las reglas solo contienen el operador lógico "Y". Para evaluar este operador en Lógica Borrosa, se emplea una t-norma, siendo las más comunes el producto algebraico y la intersección (mínimo). Esta última ha sido la utilizada para evaluar el valor de verdad (v) del antecedente de la i-ésima regla mediante la siguiente expresión:

$$v_i = \min(\mu_{\Delta T_a i}(\Delta T_a), \mu_{\Delta T_p i}(\Delta T_p)) \tag{37}$$

donde $\mu_{\Delta T_a i}$ representa la función de pertenencia de la variable ΔT_a que aparece en la i-ésima regla y $\mu_{\Delta T_p i}$ representa la función de pertenencia de la variable ΔT_p que aparece también en dicha regla.

La Figura 5.5 muestra un ejemplo del cálculo del valor de verdad del antecedente de una regla cuando se reciben los valores numéricos 0.07 y 0.02 para las variables de entrada ΔT_a y ΔT_p, respectivamente.

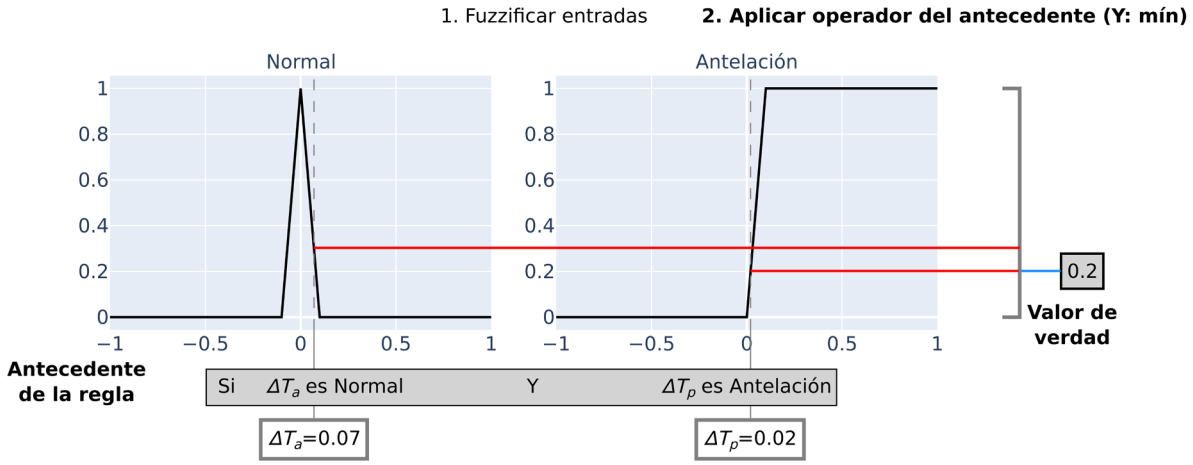

Figura 5.5: Ejemplo del cálculo del valor de verdad del antecedente de una regla

Una vez que se ha calculado v, el siguiente paso es ejecutar la implicación. La implicación es una operación que permite relacionar el resultado de evaluar el antecedente de una regla con el consecuente. Esta operación consiste en obtener una versión modificada del conjunto borroso de la variable lingüística presente en el consecuente en dependencia del valor de verdad determinado previamente. En este caso, la operación de implicación también se rige por la t-norma del mínimo. La siguiente expresión describe como determinar la función de pertenencia del conjunto borroso resultante de la implicación ($\mu'_{ui}(u)$) de la i-ésima regla:

$$\mu'_{ui}(u) = \min(v_i, \mu_{ui}(u)) \tag{38}$$

donde $\mu_{ui}(u)$ representa la función de pertenencia de la variable de salida (u) presente en el consecuente de la i-ésima regla.

La Figura 5.6 muestra un ejemplo de cómo se aplica la implicación a una regla.

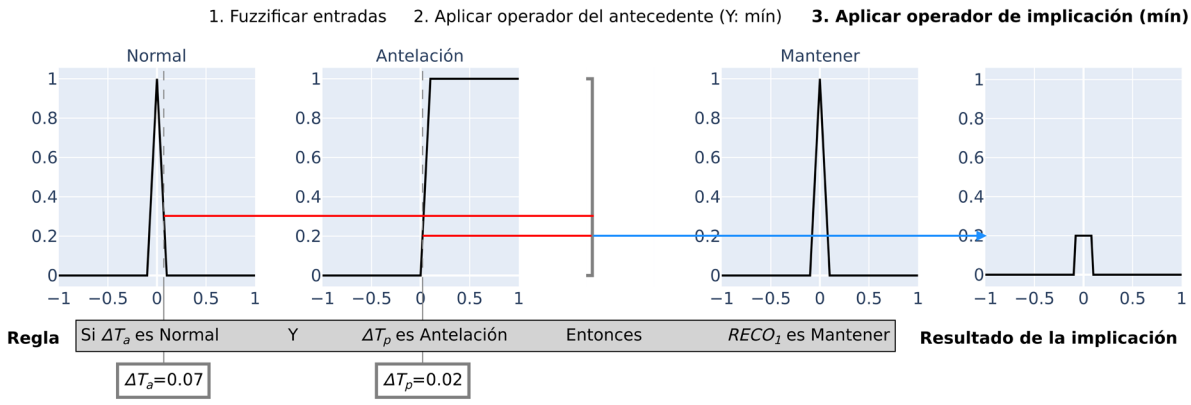

Figura 5.6: Ejemplo de implicación

La última operación en el proceso de inferencia es la agregación de los resultados de la implicación del total de las n reglas presentes en la base de reglas. La agregación en un sistema borroso tipo Mamdani se rige por una t-conorma, siendo las más usuales el máximo o la suma algebraica. Generalmente, esta operación da como resultado un conjunto borroso descrito por una función de pertenencia más compleja que las de los conjuntos borrosos que la originan. En la metodología se ha decidido utilizar la t-conorma del máximo y, por tanto, el resultado de la agregación ($\mu_R(u)$) se puede describir mediante la siguiente ecuación:

$$\mu_R(u) = \max(\mu'_{u1}(u), \mu'_{u2}(u), \dots, \mu'_{un}(u)) \tag{39}$$

5.2.4. Desborrosificación y salidas del sistema

Por último, una vez que se ha generado el conjunto borroso resultante mediante el mecanismo de inferencia, es necesario transformar esta respuesta a un valor numérico concreto. Esta operación se conoce como desborrosificación y la literatura contempla varios métodos comunes para ejecutarla: centroide, bisectriz, media de máximos, primero de máximos y último de máximos, entre otros.

El método del centroide consiste en calcular el promedio ponderado del conjunto borroso generado por el mecanismo de inferencia. Matemáticamente este método se expresa de la siguiente forma:

$$u' = \frac{\int u\mu_R(u)du}{\int \mu_R(u)du} \tag{40}$$

donde u' es el valor numérico concreto calculado para la variable de salida u. El rango de integración en este cálculo se corresponde con el universo de discurso de la variable, es decir, el intervalo de todos los posibles valores que puede tomar esta.

Por su parte, el método de la bisectriz trata de buscar el valor numérico de la variable de salida que separa la función de pertenencia del conjunto borroso resultante en dos partes con la misma área. Mientras tanto, los métodos de la media, primero y último de los máximos se enfocan en la región donde la función de membresía del conjunto borroso resultante toma valores más altos. Si el valor máximo se alcanza en un único punto, estos tres métodos ofrecen el mismo resultado. Sin embargo, si hay más de un punto donde se alcanza el valor máximo se utilizan las siguientes ecuaciones para aplicar los métodos de la media, el primero y el último de los máximos, respectivamente:

$$u' = \frac{\max\left(\arg\sup\left(\mu_R(u)\right)\right) - \min\left(\arg\sup\left(\mu_R(u)\right)\right)}{2} \tag{41}$$

$$u' = \min\left(\arg\sup\left(\mu_R(u)\right)\right) \tag{42}$$

$$u' = \max\left(\arg\sup\left(\mu_R(u)\right)\right) \tag{43}$$

La Figura 5.7 presenta los resultados de aplicar diferentes métodos de desborrosificación al conjunto borroso de la variable $RECO_1$ obtenido luego de efectuar la operación de agregación en el sistema propuesto con los datos que se han utilizado a manera de ejemplo a lo largo de este capítulo.

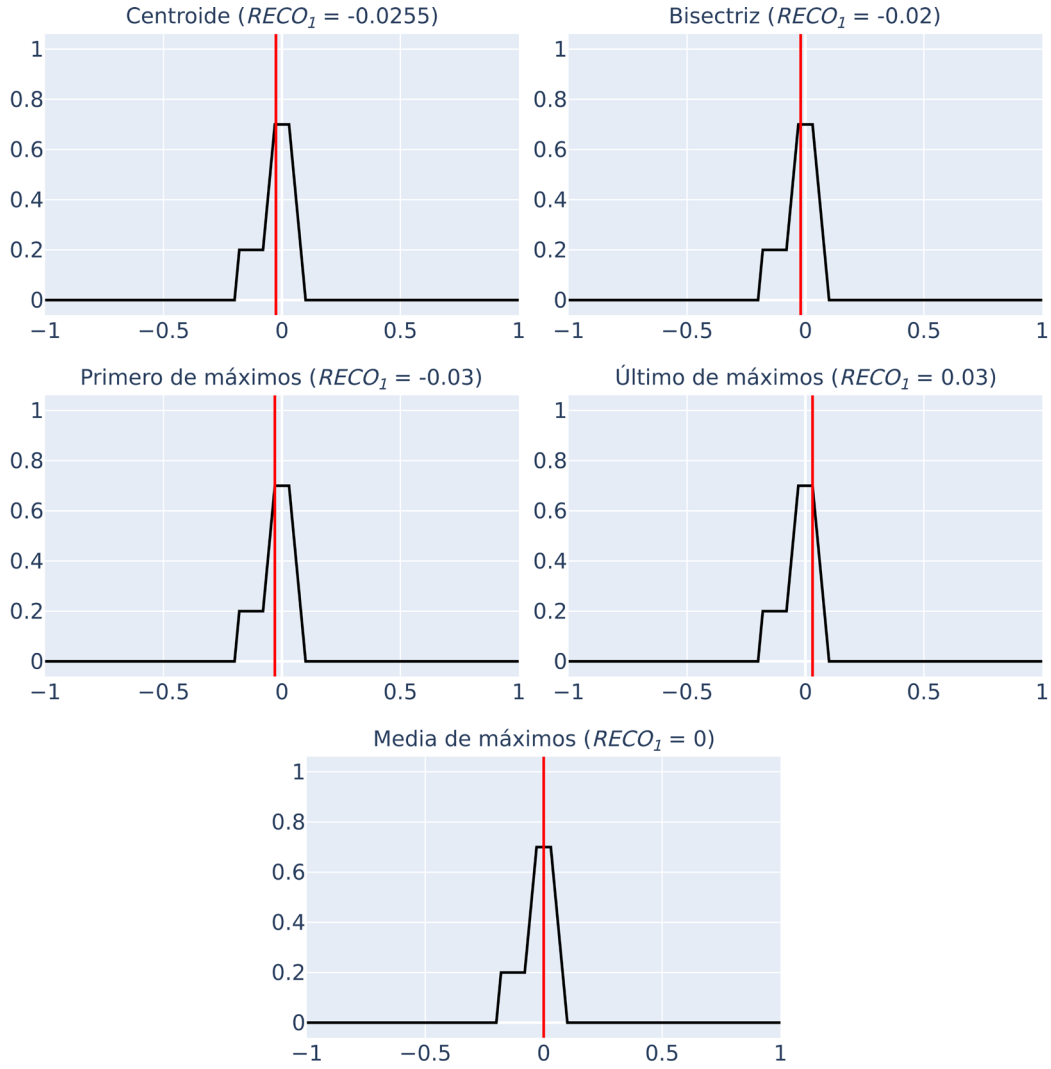

Figura 5.7: Resultados de diferentes métodos de desborrosificación

En la metodología se decidió utilizar el método del centroide para el cálculo de la variable de salida. De los métodos presentados, el del centroide es el más utilizado cuando se trabaja con conjuntos borrosos continuos, ya que se tiene en cuenta toda la información del conjunto generado por el mecanismo de inferencia y, además, es más robusto frente a pequeñas variaciones en la forma de las funciones de pertenencia comparado con los otros métodos.

De la forma en que se definieron los conjuntos borrosos de la variable de salida, los valores de $RECO_1$ generados por el sistema de inferencia serán desborrosificados en el intervalo [-1, 1]. Si el usuario de la metodología prevé utilizarla para crear un sistema de toma de decisiones automatizado que sea capaz de reconfigurar

directamente los equipos productivos, deberá implementar la forma de interpretar el valor de $RECO_1$ para transformarlo a la escala del parámetro que controle la productividad del equipo. La metodología asume que si $RECO_1$ toma un valor de -1, se debe reducir el valor del parámetro de control de la productividad a un mínimo predefinido, mientras que al tomar un valor de 1 se incrementará el valor de dicho parámetro a un máximo predefinido, con todos los valores intermedios siguiendo una escala lineal.

Por otra parte, la metodología también prevé ofrecer los resultados en un segundo formato de tipo nominal. Con esto, se busca facilitar la implementación de sistemas de recomendaciones inteligentes que puedan indicarle al operador del proceso productivo si debe ejecutar alguna acción de manera fácilmente interpretable. Para este fin, se define una variante de la variable de salida denominada $RECO_{1_nom}$. Para facilitar la interpretabilidad, las etiquetas que se asignen a estas variables coincidirán con las variables lingüísticas definidas para $RECO_1$. Las etiquetas serán asignadas teniendo en cuenta la variable lingüística del conjunto borroso cuya función de pertenencia tenga mayor valor al ser evaluada en el resultado desborrosificado previamente. De esta forma, las etiquetas se asignarán según la siguiente ecuación:

$$RECO_{1_nom} = \begin{cases} Parar\ 1, & si\ RECO_1 < -0.15 \\ Reducir, & si\ -0.15 \leq RECO_1 < -0.05 \\ Mantener, & si\ -0.05 \leq RECO_1 \leq 0.05 \\ Aumentar, & si\ 0.05 < RECO_1 \leq 0.15 \\ Parar\ 2, & si\ RECO_1 > 0.15 \end{cases} \quad (44)$$

Se debe notar que la metodología contempla ofrecer el resultado en ambos formatos, tanto el numérico ($RECO_1$), como el nominal ($RECO_{1_nom}$), siendo el usuario el que decida cuál de ellos utilizar en dependencia del tipo de sistema que vaya crear.

5.3. Procedimiento de implementación

Dado que la comunicación con *Reco* se realiza a través Orion, en este caso, también es necesario utilizar entidades NGSI-v2 para interactuar con el módulo. Por otra parte, tal como se muestra en la Figura 5.8, para la implementación de *Reco* se utilizó Simpful, la cual es una biblioteca de Python desarrollada para el razonamiento borroso. Esta biblioteca permite trabajar con sistemas de inferencia tipo Mamdani o Takagi-Sugeno de cualquier orden (Spolaor et al., 2020). Además, Simpful soporta la definición y utilización de reglas arbitrariamente complejas, así

como el uso de funciones de pertenencia altamente personalizables. Una de las características más relevantes de esta biblioteca es que proporciona una interfaz ligera y altamente intuitiva, a través de la cual las reglas pueden ser definidas incluso mediante cadenas de texto en lenguaje natural. El módulo *Reco* también ha sido encapsulado en un contenedor creado con Docker para independizar su funcionamiento del entorno de ejecución. Finalmente, para facilitar las comunicaciones entre los elementos del sistema productivo que se utilizarán para validar la propuesta y la arquitectura de KITT4SME se desarrolló un componente basado en paho-mqtt. Este se encarga de traducir los mensajes de nivel de planta a las entidades NGSI-v2 necesarias y de despachar las respuestas producidas por *Reco*.

Figura 5.8: Herramientas y bibliotecas utilizadas en la implementación de las propuestas relacionadas con el método de rectificación de desviaciones en la producción

5.4. Conclusiones

En este capítulo se presentó la metodología de reconfiguración automática e inteligente la cual está enfocada a la corrección de desviaciones en la productividad de las máquinas/robots y dispositivos del sistema de producción (*Reco*), la cual

servirá de base en la implementación de un módulo de la arquitectura de KITT4SME. Se especificaron los requerimientos para utilizar la metodología y se profundizó en cada una de las etapas del sistema de inferencia borroso de tipo Mamdani. Se definieron los conjuntos borrosos de las variables de entrada y salida y se explicaron los procesos de borrosificación y desborrosificación en el contexto de la metodología. Además, se detalló la base de reglas que describe las relaciones entre las entradas y salidas, y se describió el proceso que se lleva a cabo para generar las respuestas ante las diferentes situaciones que se deben manejar en la metodología. Finalmente, se introdujeron las diferentes herramientas que se emplearán para llevar a cabo la implementación del módulo basado en la metodología presentada, así como para su aplicación en la línea piloto de GAMHE.

6. Resultados y Discusión

6.1. Introducción

En este capítulo se realiza una descripción de la línea piloto de GAMHE, la cual representa un proceso productivo real en una empresa con un nivel de madurez tecnológica medio/bajo. Primeramente, se detalla el flujo del proceso productivo, las variables y los KPIs que se desean optimizar. Además, se presentan los resultados de utilizar el primer módulo desarrollado para la arquitectura KITT4SME para el modelado y optimización de la línea piloto a partir de un conjunto de datos históricos. Se realiza un análisis exhaustivo de los resultados de cada etapa de la metodología de reparametrización óptima de la producción. También se muestra la réplica del escenario productivo desarrollada como parte del gemelo digital y se presentan los resultados de su funcionalidad de supervisión para detectar la degradación oportuna de los modelos. Finalmente, se presentan los resultados de utilizar el segundo módulo desarrollado para la arquitectura KITT4SME en la reconfiguración automática e inteligente para la corrección de desviaciones en la productividad de las máquinas/robots y dispositivos de fabricación al ser evaluado en la línea piloto.

6.2. Descripción de la línea piloto

En la línea piloto de GAMHE se llevan a cabo las etapas finales de un proceso de fabricación de paneles termoaislantes. Estos paneles son utilizados en la industria de la construcción para mejorar el aislamiento térmico de las edificaciones, de manera que se incremente la eficiencia energética al reducir las pérdidas de calor en el invierno y el calor excesivo en verano. Normalmente, los paneles termoaislantes se utilizan como recubrimientos de paredes y techos y son fijados a estos mediante tornillos, adhesivos o sistemas de anclaje oculto. Los paneles termoaislantes se componen generalmente de una capa de material termoplástico que se coloca entre dos láminas de acero, aluminio u otro material. Entre los materiales termoplásticos más comunes utilizados para la fabricación de paneles termoaislantes se encuentran la espuma de poliestireno expandido, el poliuretano y la fibra de vidrio. Estos paneles generalmente se fabrican en forma rectangular o cuadrada, aunque pueden ser cortados y manipulados de manera sencilla para

adaptarlos a las necesidades del proyecto en que se estén utilizando. La Figura 6.1 muestra dos vistas de un panel termoaislante de poliestireno expandido.

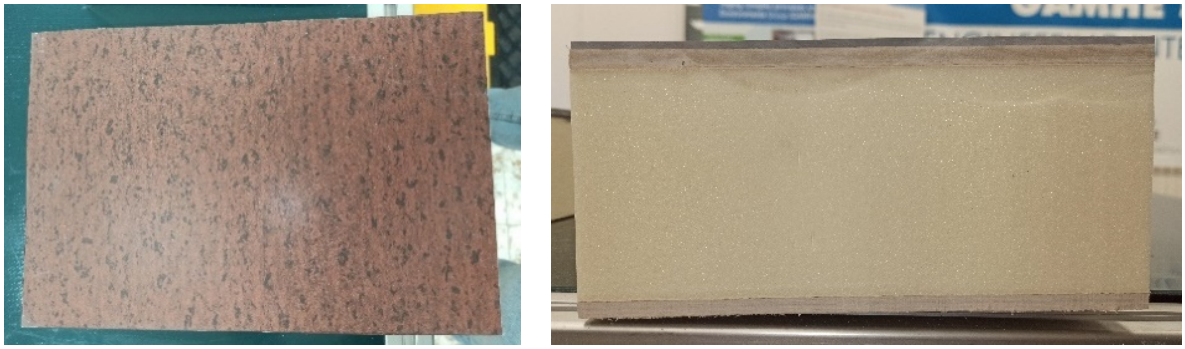

Figura 6.1: Vista superior y lateral de un panel termoaislante

El proceso productivo de los paneles termoaislantes se compone de varias operaciones. Las primeras operaciones consisten en la expansión o conformación del material aislante, su corte de acuerdo con las dimensiones deseadas para el panel, el pegado mediante algún adhesivo de las láminas y el prensado hasta que el pegado haya finalizado. En el caso de la línea piloto de GAMHE, los paneles son recibidos una vez que se han completado estas operaciones. A continuación, se realiza un proceso de perfilado para corregir cualquier irregularidad que pueda presentar el panel y se realiza un proceso de corte para ranurar las cuatro caras laterales de forma paralela a las láminas. Los paneles son ranurados, en este caso, para prepararlos para un sistema de sujeción de anclaje oculto en el cual deben encajar en una estructura metálica que sirve como guía y soporte. Seguidamente, los paneles son transferidos hasta un sistema de inspección de calidad y, por último, son sometidos a una operación de micro grabado donde se plasma en una de las superficies metálicas un patrón predefinido (Beruvides et al., 2017). La Figura 6.2 resume estas operaciones.

Para ejecutar estas operaciones de manera automatizada, la línea piloto de GAMHE cuenta con varios robots y máquinas. En la línea también se encuentra un operador, pero este solo se encarga de posicionar los paneles al principio de la línea, retirarlos al final de la línea y supervisar que el funcionamiento de los distintos componentes es correcto. El primer elemento de la línea es un robot Stäubli modelo RX90 (denominado A_1), el cual se encarga de tomar los paneles desde la posición inicial donde los coloca el operador e insertarlos en el segundo elemento de la línea, un centro de mecanizado Deckel Maho DMC 75V Linear

(denominado A_2). En este centro de mecanizado es donde se llevan a cabo las operaciones de perfilado y ranurado. Una vez que se han completado estas operaciones, el robot Stäubli retira los paneles del centro de mecanizado y los coloca en un sistema compuesto por dos cintas transportadoras (denominado A_3). Las cintas mueven los paneles hasta una posición donde pueden ser retirados por un robot UR5e (denominado A_4), que posteriormente los va posicionando de manera tal que el sistema de inspección de calidad que utiliza una cámara Allied Vision modelo Mako G192C (denominada A_5) pueda capturar imágenes para examinarlos. A continuación, el robot UR5e coloca los paneles en una fresadora con control numérico computarizado (CNC) de 5 ejes KERN Evo (denominada A_6) para llevar a cabo la operación de micro grabado. Finalmente, una vez que se ha concluido esta operación, el robot UR5e retira los paneles de la fresadora y los coloca en una posición donde el operador puede retirarlos. La Figura 6.3 muestra los diferentes elementos de la línea piloto de GAMHE, mientras que la Figura 6.4 representa la secuencia en que estos intervienen en el proceso productivo.

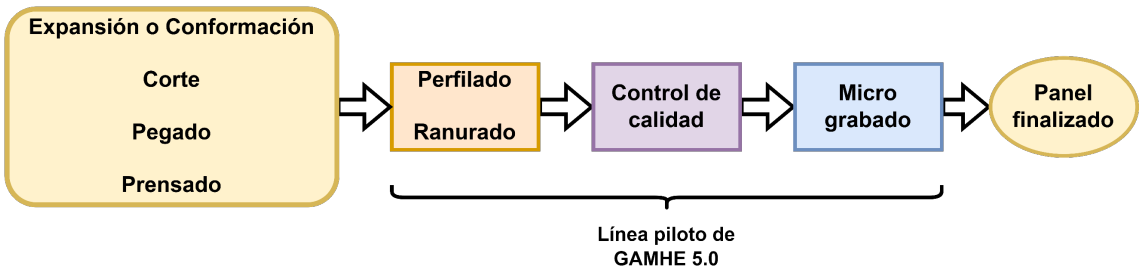

Figura 6.2: Operaciones para la manufactura de los paneles termoaislantes en la línea piloto de GAMHE

Para lograr que los elementos de la línea piloto de GAMHE actúen de manera coordinada se requiere una comunicación constante entre ellos. Debido a la heterogeneidad de los elementos, en la línea piloto se cuenta con una red de nodos IoT para facilitar las comunicaciones. Las comunicaciones en esta red se realizan utilizando el protocolo MQTT, razón por la cual ciertos componentes presentados en el capítulo anterior han sido implementados utilizando la biblioteca paho-mqtt y así, poder integrarlos directamente en la red. Cada uno de los elementos de la línea piloto ha sido asociado a un ordenador de placa única tipo Raspberry Pi que se encarga de procesar los mensajes recibidos y transmitirle la información al elemento pertinente, así como de elaborar los mensajes que el elemento asociado desee enviar.

Figura 6.3: Línea piloto del grupo GAMHE

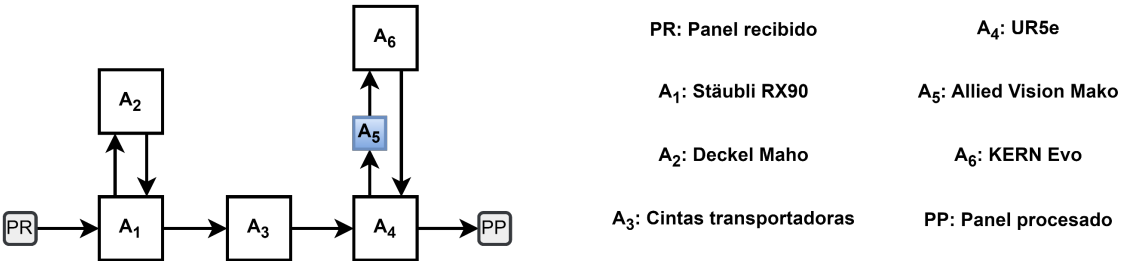

Figura 6.4: Secuencia de actuación de los elementos de la línea piloto de GAMHE

Por otra parte, para la realización de las pruebas de los módulos desarrollados para la arquitectura KITT4SME, así como el resto de los componentes necesarios para su aplicación en la línea piloto se recopilaron datos de varias variables y KPIs del proceso. A pesar de que la línea se encuentra configurada para procesar solo paneles termoaislantes, estos pueden tener diferentes especificaciones. Dado que el proceso de ranurado debe adaptarse a los paneles, una de las variables que se registra es el perímetro de estos (p), el cual coincide con la longitud de corte. También se registraron las siguientes variables de los elementos de la línea: velocidad de trabajo del robot Stäubli (v_s), avance del centro de mecanizado Deckel (a_d), velocidad del cabezal del centro de mecanizado Deckel (v_d), velocidad del motor de las cintas transportadoras (v_c), velocidad de trabajo del robot UR5e (v_u) y avance de la fresadora KERN (a_k). La Tabla 6.1 presenta los rangos de trabajo utilizados durante la creación del conjunto de datos y las pruebas en la línea piloto

de GAMHE. En el caso de las velocidades de trabajo de los robots, estas son variables facilitadas por las interfaces de control, las cuales se utilizan internamente para fijar los valores de velocidad de las articulaciones.

Tabla 6.1: Rangos de trabajo experimentales definidos para las variables en la línea piloto de GAMHE

Variable	Mínimo definido	Máximo definido
p	100 cm	120 cm
v_s	3.3 m/s	7.7 m/s
a_d	27000 mm/min	63000 mm/min
v_d	5400 rpm	12600 rpm
v_c	2160 rpm	5040 rpm
v_u	30%	70%
a_k	4800 mm/min	11200 mm/min

En el caso de los KPIs, se registraron los datos correspondientes al tiempo de producción por unidad (T_p) y al tiempo de inactividad del operador por unidad (T_i). Se considera tiempo de inactividad del operador por unidad al tiempo que transcurre durante el procesamiento de un panel en el cual el operador no realiza acciones físicas relacionadas con la producción. Durante la recolección de datos para las pruebas, todos los parámetros y KPIs fueron registrados de manera automática, excepto T_i. Dado que actualmente en la línea piloto no se cuenta con sensores para medir directamente este KPI, se realizó un estudio en el que se cronometraron los períodos de inactividad en múltiples circunstancias y se determinó la diferencia media (μ_d) con respecto a T_p y la desviación estándar de esta diferencia (σ_d^2). Contando con estos parámetros estadísticos se generaron los valores de T_i incluidos en el conjunto de datos utilizando la siguiente ecuación:

$$T_{i_m} = T_{p_m} - s \tag{45}$$

donde m representa el número de la muestra en el conjunto de datos y $s \sim \mathcal{N}(\mu_d, \sigma_d^2)$.

6.3. *Insight Generator*: Aplicación y resultados

El primer módulo desarrollado para la arquitectura KITT4SME que se evaluó en la línea piloto de GAMHE fue *Insight Generator*. La metodología de aprendizaje automático automatizado propuesta completó su ejecución en aproximadamente 2348 s al procesar un conjunto de datos compuesto por 1250 muestras tomadas de

la línea piloto. De esta duración total, se dedicaron 1473 s a las operaciones que abarcan desde que se reciben los datos hasta la fase de selección de los modelos, mientras que los restantes 875 s se emplearon en optimizar el proceso productivo.

6.3.1. Acondicionamiento de los datos

Una vez que se cargan los datos y se realizan todas las verificaciones y modificaciones iniciales, se lleva a cabo la estandarización. La Figura 6.5 muestra los resultados de la estandarización de los datos de entrada del conjunto de entrenamiento. Se puede observar como antes de la estandarización existe una notable diferencia entre los rangos de las variables. Los valores de a_d superan en varias unidades de magnitud a otros como p, v_s y v_u. Al llevar a cabo la operación de estandarización los datos son escalados a rangos similares, lo que permite evitar sesgos en la posterior selección de las variables a utilizar para generar el modelo de cada variable de salida.

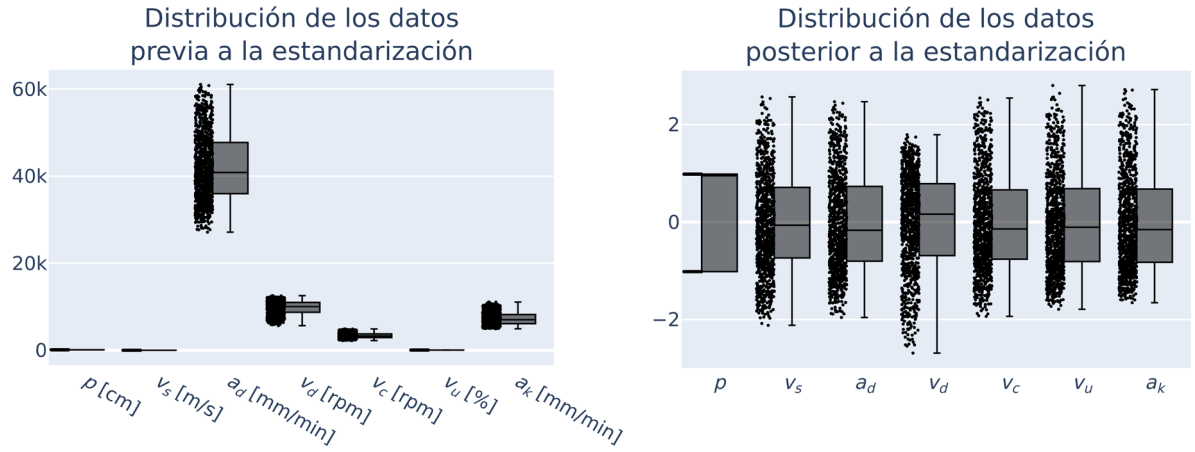

Figura 6.5: Estandarización de las variables de entrada

6.3.2. Selección de variables

El siguiente paso consiste en determinar que variables se utilizarán para construir los modelos de cada KPI. Para efectuar la selección se utilizó el criterio definido en el Capítulo 4, el cual se basa en considerar simultáneamente los valores de los coeficientes de correlación de Pearson y de Spearman. La Figura 6.6 y la Figura 6.7 muestran las matrices de correlación de Pearson y de Spearman para los dos KPIs analizados en la línea piloto, respectivamente. Como resultado, para ambos

KPIs se seleccionaron las variables v_s, a_d, v_u y a_k. Denotar que la inclusión de la variable v_s entre las seleccionadas para ambos KPIs se debe al valor obtenido solamente mediante el coeficiente de correlación de Pearson, ya que el valor del coeficiente de correlación de Spearman no superó el umbral predefinido. Por otra parte, el hecho de que se hayan seleccionado las mismas variables para ambos KPIs se debe a la naturaleza de estos, ya que están estrechamente relacionados. Si el tiempo necesario para procesar una unidad incrementa, suponiendo que el operador mantenga un ritmo de trabajo estable, el tiempo de inactividad física también debe aumentar, y viceversa.

La información de las variables seleccionadas proporciona un nivel elemental de explicabilidad. En este caso, es evidente que las variables relacionadas con la velocidad a la que se realizan las operaciones son las que influyen en ambos KPIs y, además, que la relación entre ellas es inversa. Esta intuición coincide con los resultados obtenidos donde las variables que mayor correlación tienen con ambos KPIs son las velocidades y, además, la correlación es negativa, validando la etapa de selección de variables en la metodología. A pesar de que la variable v_c también tiene influencia en los KPIs, esta no ha sido seleccionada ya que no superó el umbral predefinido. Al analizar el funcionamiento del proceso productivo se puede advertir que el tiempo que están las piezas en las cintas transportadoras es relativamente pequeño comparado con el tiempo que pasan en el resto de los elementos de la línea. Por tanto, una variación en la velocidad de las cintas no representa un gran cambio en las salidas analizadas. También es notable que no se haya seleccionado la variable v_d, la cual no tiene relación directa con el tiempo de mecanizado.

6.3.3. Optimización y selección de modelos

Una vez han sido seleccionadas las variables, se ejecuta el proceso de selección de modelos, el cual se repite para cada KPI. Este proceso se inicia con la optimización de hiperparámetros de los modelos base. Seguidamente, se prueban las dos estrategias de aprendizaje de conjunto y se realiza una comparación entre el desempeño de estas estrategias y el de cada uno de los modelos base, seleccionándose la mejor opción. Las dos siguientes subsecciones muestran los resultados obtenidos en las etapas de optimización de hiperparámetros y de evaluación y selección del modelo para los dos KPIs considerados en la línea piloto.

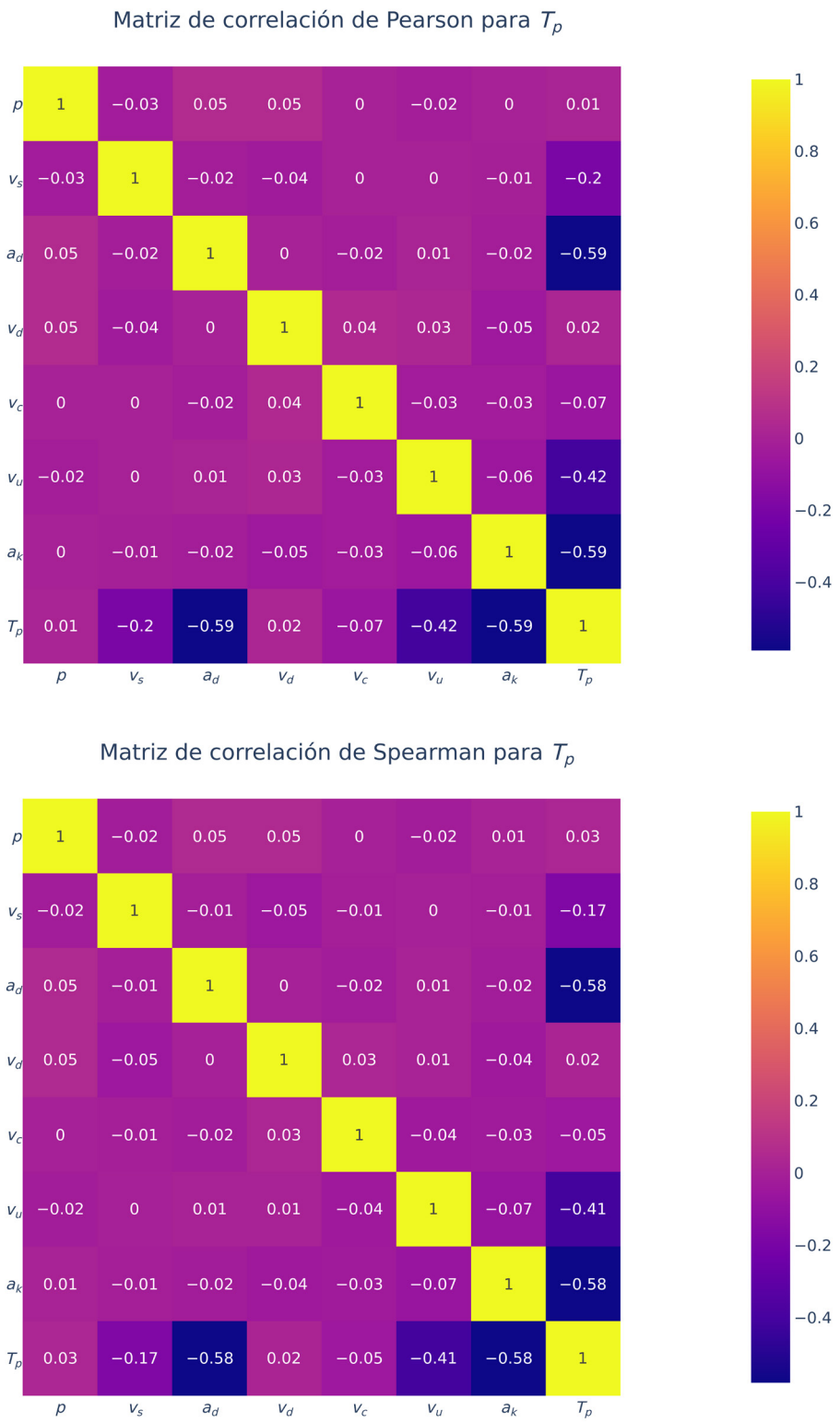

Figura 6.6: Matrices de correlación para T_p

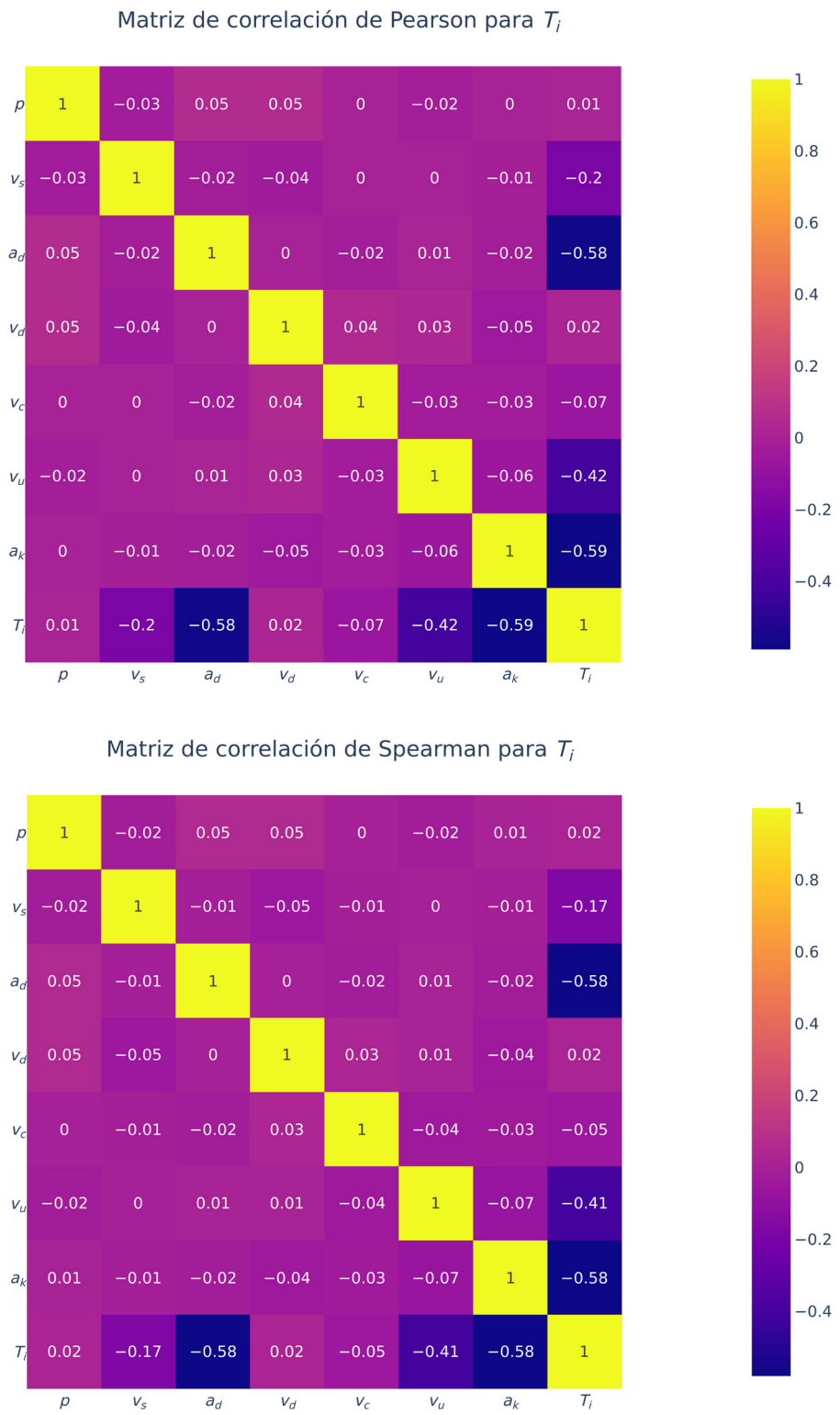

Figura 6.7: Matrices de correlación para T_i

6.3.3.1. Primer indicador: Tiempo de producción por unidad

El primer modelo candidato para representar a T_p cuyos hiperparámetros fueron optimizados fue MLP. La Figura 6.8 muestra los resultados de este proceso[12]. Como se puede apreciar, la selección adecuada de hiperparámetros es de gran relevancia en este modelo, ya que ciertas combinaciones pueden propiciar la creación de modelos con muy bajo desempeño, con valores de R^2 en el rango de -140. En este caso, los hiperparámetros que más influencia demostraron en el desempeño del modelo fueron el número de unidades en la primera capa oculta, la tasa de aprendizaje y la función de activación.

El segundo modelo candidato para representar a T_p al cual se le optimizaron sus hiperparámetros fue SVR. En la Figura 6.9 se puede ver que el hiperparámetro C es el que tiene mayor relevancia para alcanzar un mejor desempeño, con una marcada tendencia a mejorar el valor de R^2 a medida que se centra la búsqueda en el límite superior del rango definido para este hiperparámetro.

A continuación, se optimizaron los hiperparámetros de XGBoost. Los resultados pueden ser apreciados en la Figura 6.10. En la mayor parte del espacio de búsqueda de los hiperparámetros se obtienen buenos valores de desempeño, excepto en dos combinaciones, en una de las cuales, el valor de R^2 llega a tener un valor aproximado de -10. Además, se logra determinar que la mayor influencia en el desempeño es debido al número de árboles y a la tasa de aprendizaje.

Seguidamente, se optimizaron los hiperparámetros de *k*NN. La Figura 6.11 muestra los resultados obtenidos en este caso. Es destacable que para este modelo todas las combinaciones de hiperparámetros resultan tener un buen desempeño, con valores de R^2 superiores a 0.89, por lo que, en este caso, el proceso de optimización de hiperparámetros no tiene tanta relevancia como para los otros modelos.

[12] Destacar que, para asegurar una correcta visualización, la mayoría de los gráficos de contorno y de corte incluidos en esta tesis contienen solo una selección de todos los hiperparámetros a optimizar.

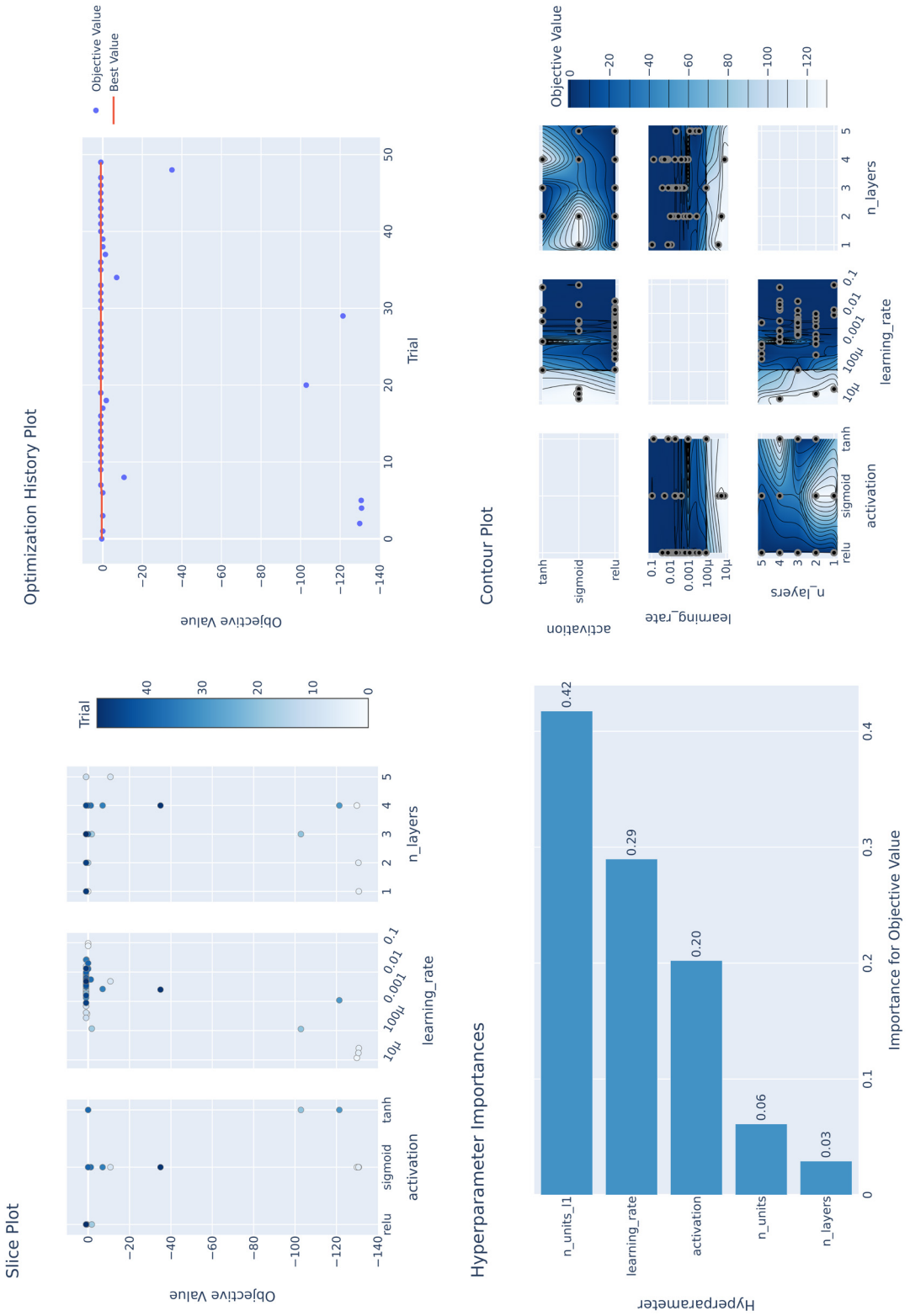

Figura 6.8: Optimización de hiperparámetros de MLP para T_p

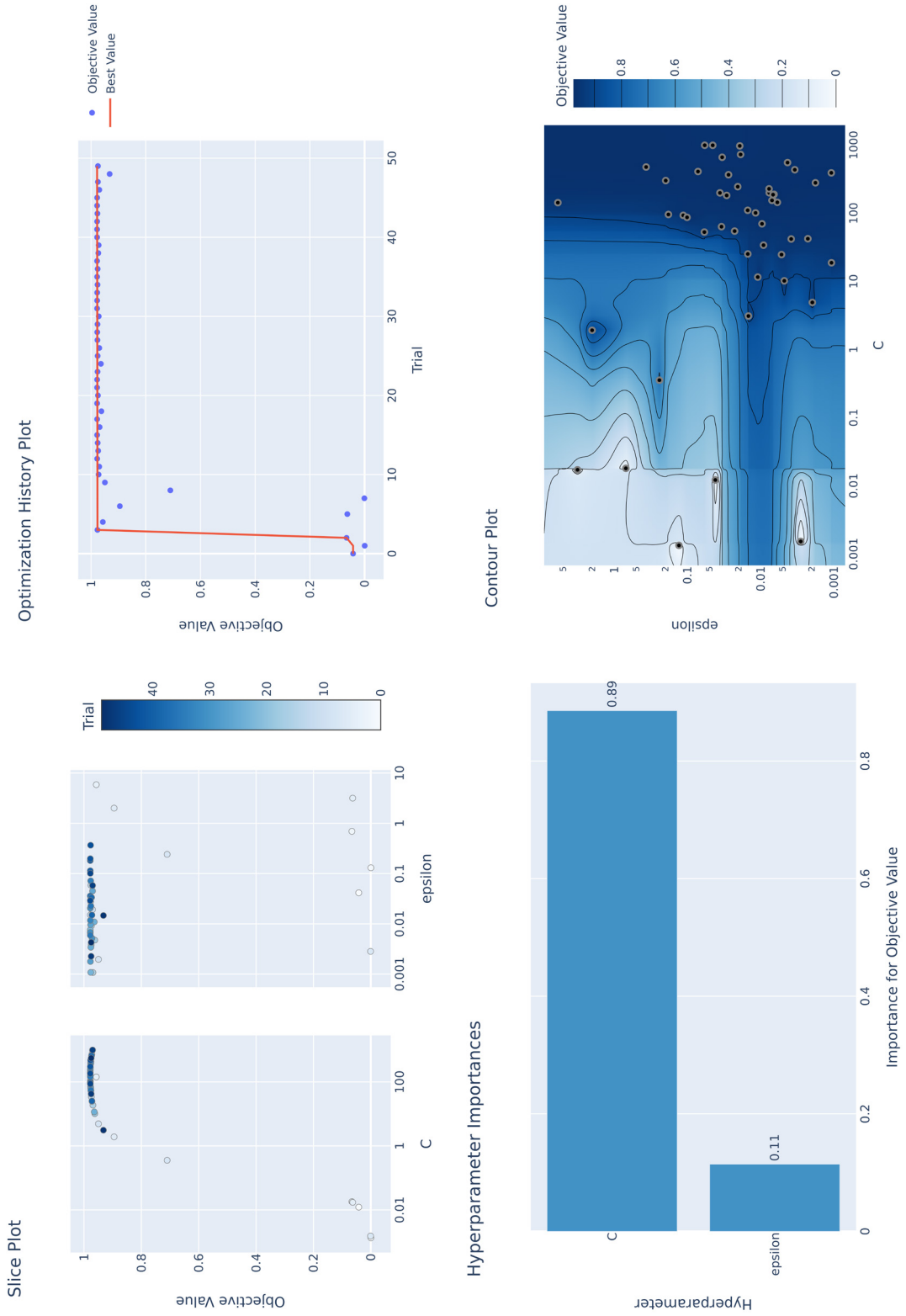

Figura 6.9: Optimización de hiperparámetros de SVR para T_p

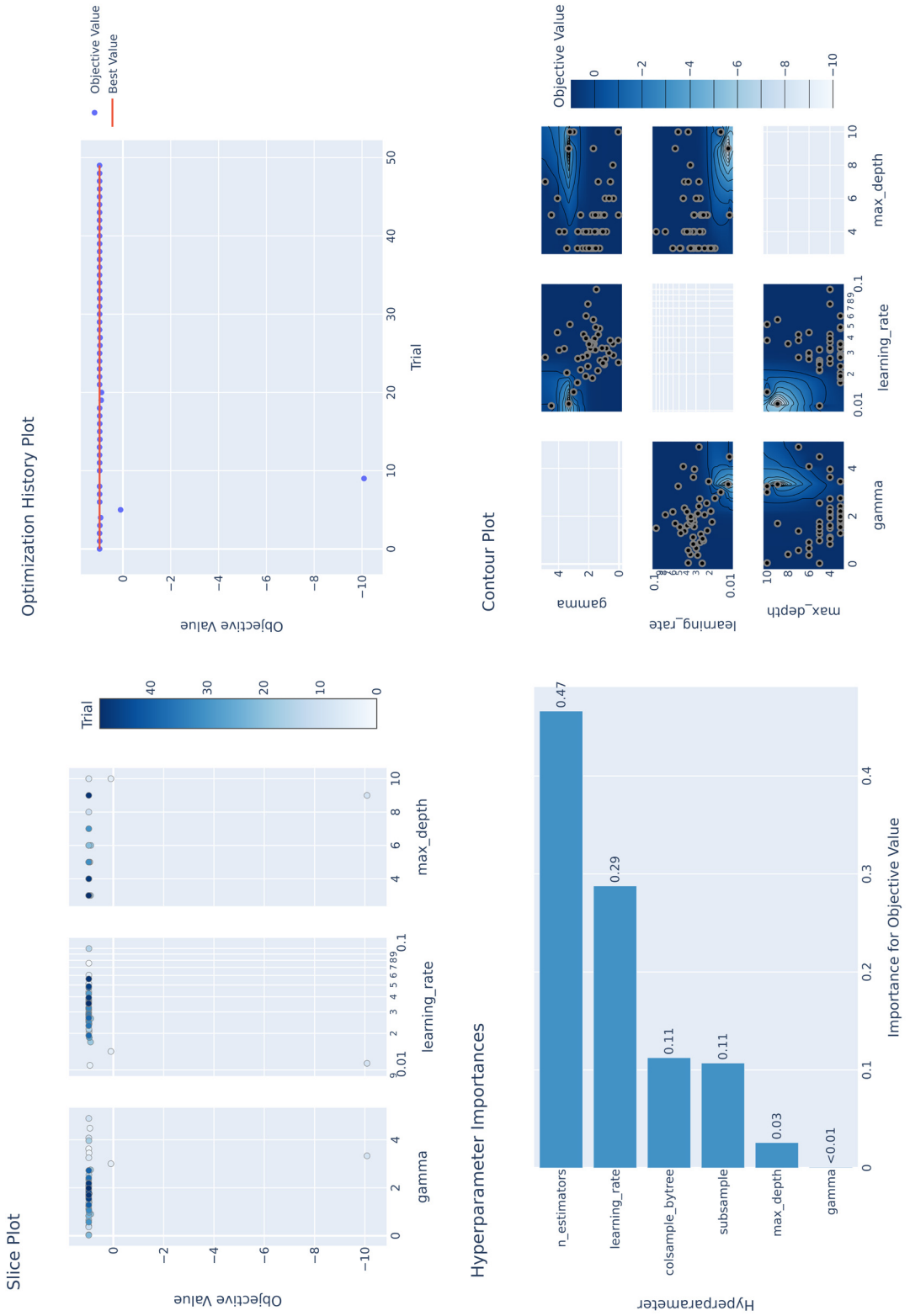

Figura 6.10: Optimización de hiperparámetros de XGBoost para T_p

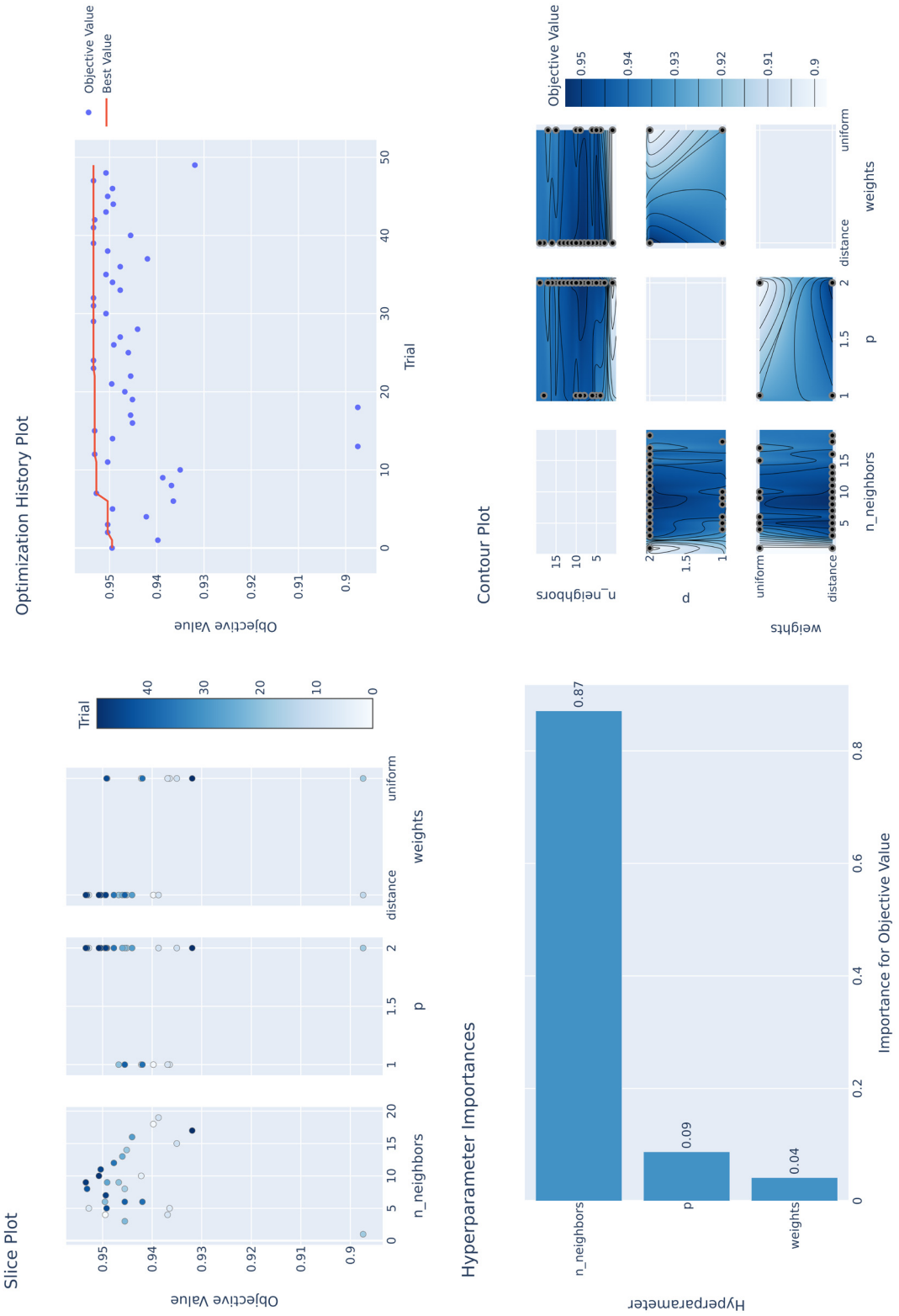

Figura 6.11: Optimización de hiperparámetros de kNN para T_p

Al haberse optimizado los hiperparámetros de los cuatro modelos base se procede a entrenar el metamodelo con el conjunto de datos de retención[13] y, a continuación, se evalúan todas las opciones para seleccionar la que mejor capacidad de generalización demuestre al obtener un mayor valor de R^2 para el conjunto de datos de prueba. La Tabla 6.2 resume los valores de los hiperparámetros de los modelos base tras la optimización, así como datos de la evaluación de todos los modelos. Por otra parte, la Figura 6.12 muestra gráficamente los valores reales de T_p en el conjunto de prueba contra los valores predichos por cada modelo. En estas gráficas se puede ver como las predicciones de todos los modelos tienen un alto nivel de similitud con los valores esperados, donde *k*NN es el modelo que muestra una dispersión ligeramente superior al resto. Tras comparar los resultados, la metodología seleccionó la estrategia de metamodelado para representar a T_p. Esta decisión supone almacenar todos los modelos base además del metamodelo, ya que se necesitan todos estos elementos para realizar inferencias posteriormente.

[13] Nótese que la primera estrategia de aprendizaje de conjunto no requiere ningún entrenamiento extra.

Tabla 6.2: Hiperparámetros seleccionados y resultados de validación para T_p

Modelo	Hiperparámetros	R^2 validación	R^2 prueba	Tiempo de optimización de hiperparámetros	Tiempo de inferencia
MLP	número de capas ocultas = 3, σ = relu, número de unidades por capa = [73, 115, 82, 38], $\propto$ = 0.003500	0.982	0.979	534.133 s	92.341 ms
SVR	C= 95.916, ε = 0.182	0.978	0.970	12.462 s	3.067 ms
XGBoost	número de árboles = 364, profundidad máxima = 3, tasa de aprendizaje = 0.036, tasa de submuestreo = 0.624, tasa de submuestreo de columnas por árbol = 0.527, γ = 1.729	0.980	0.971	75.937 s	3.476 ms
*k*NN	k = 9, función de ponderación = distancia, orden de la distancia de Minkowski = 2	0.953	0.962	**0.659 s**	**0.881 ms**
Conjunto de modelos	-	0.983	0.980	-	107.086 ms
Metamodelo	-	**0.986**	**0.982**	0.016 s (entrenamiento metamodelo)	108.448 ms

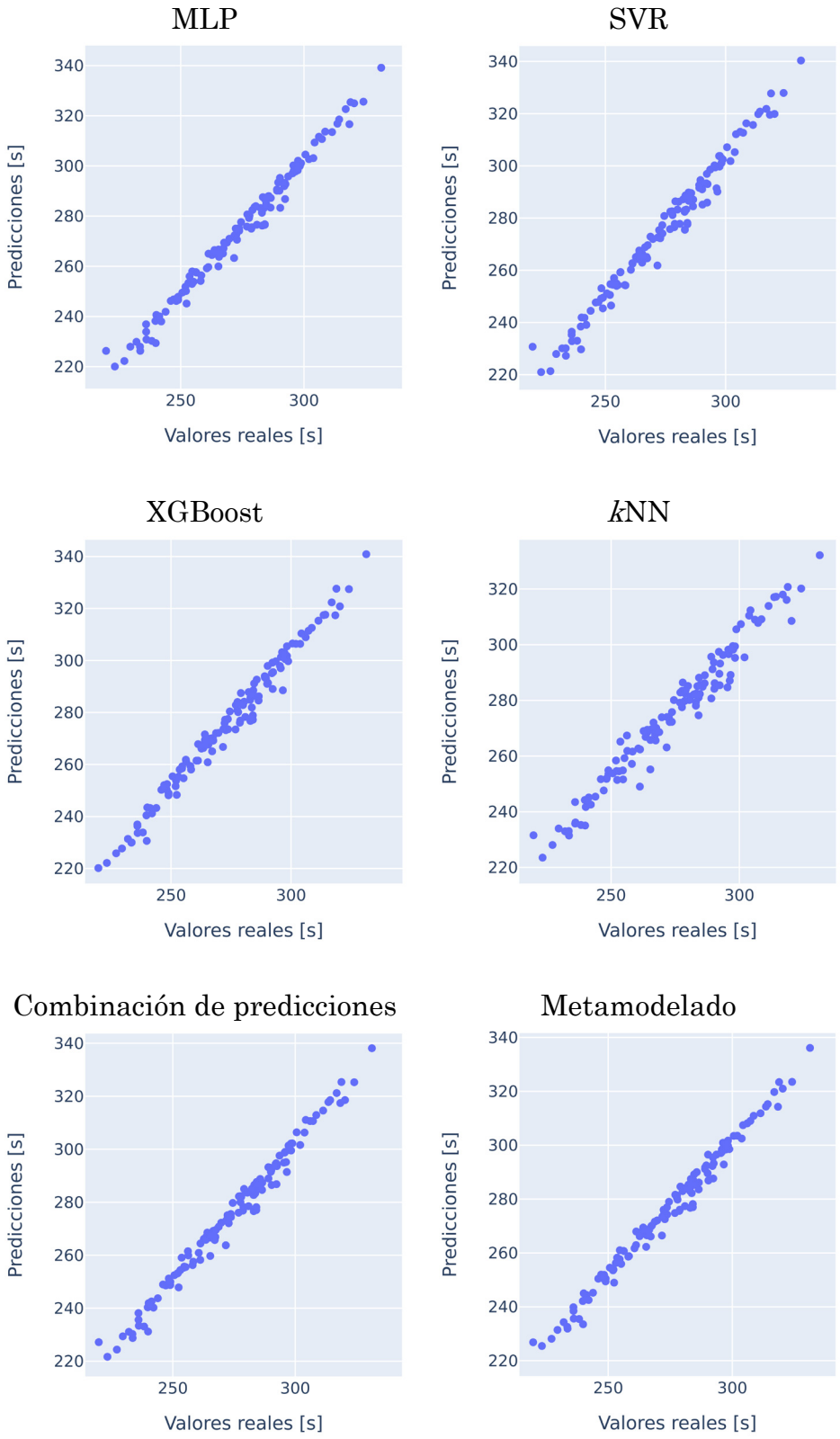

Figura 6.12: Valores reales vs. predicciones de los diferentes modelos para T_p

6.3.3.2. Segundo indicador: Tiempo de inactividad del operador por unidad

Tras haberse seleccionado un modelo para T_p, se debe repetir el proceso para T_i. Al igual que en el caso anterior, el primer modelo candidato cuyos hiperparámetros se optimizaron fue MLP. La Figura 6.13 muestra los resultados de este proceso, confirmando nuevamente la importancia de la optimización de hiperparámetros del MLP para modelar KPIs en la línea piloto pues ciertas combinaciones producen modelos con muy bajo desempeño. A diferencia de la optimización de hiperparámetros para T_p, aquí la función de activación no muestra una influencia significativa en el desempeño de los modelos, siendo más relevantes la tasa de aprendizaje, el número de neuronas en la capa oculta y el número de neuronas en la capa de entrada.

El segundo modelo candidato para representar a T_i al cual se le optimizaron sus hiperparámetros fue SVR. En la Figura 6.14 se puede ver que, al igual que en el caso de T_p, el hiperparámetro C es el que tiene mayor relevancia para alcanzar un mejor desempeño, con una marcada tendencia a mejorar el valor de R^2 a medida que se centra la búsqueda en el límite superior del rango definido para este hiperparámetro.

Seguidamente, se optimizaron los hiperparámetros de XGBoost. Como se puede ver en la Figura 6.15, para todas las combinaciones de hiperparámetros probadas se obtuvieron buenos valores de desempeño, con valores de R^2 superiores a 0.75 en todos los casos y, mayoritariamente, siendo superiores a 0.95. La mayor influencia en el desempeño es debido a la tasa de aprendizaje y al hiperparámetro γ, a diferencia de la optimización de hiperparámetros para XGBoost en el caso de T_p, donde era el número de árboles y la tasa de aprendizaje.

Por último, se optimizaron los hiperparámetros de *k*NN. Los resultados que pueden ser observados en la Figura 6.16, coinciden con los obtenidos en el caso de T_p. En resumen, todas las combinaciones de hiperparámetros resultan tener un buen desempeño, con valores de R^2 superiores a 0.89, teniendo la mayor influencia el hiperparámetro k.

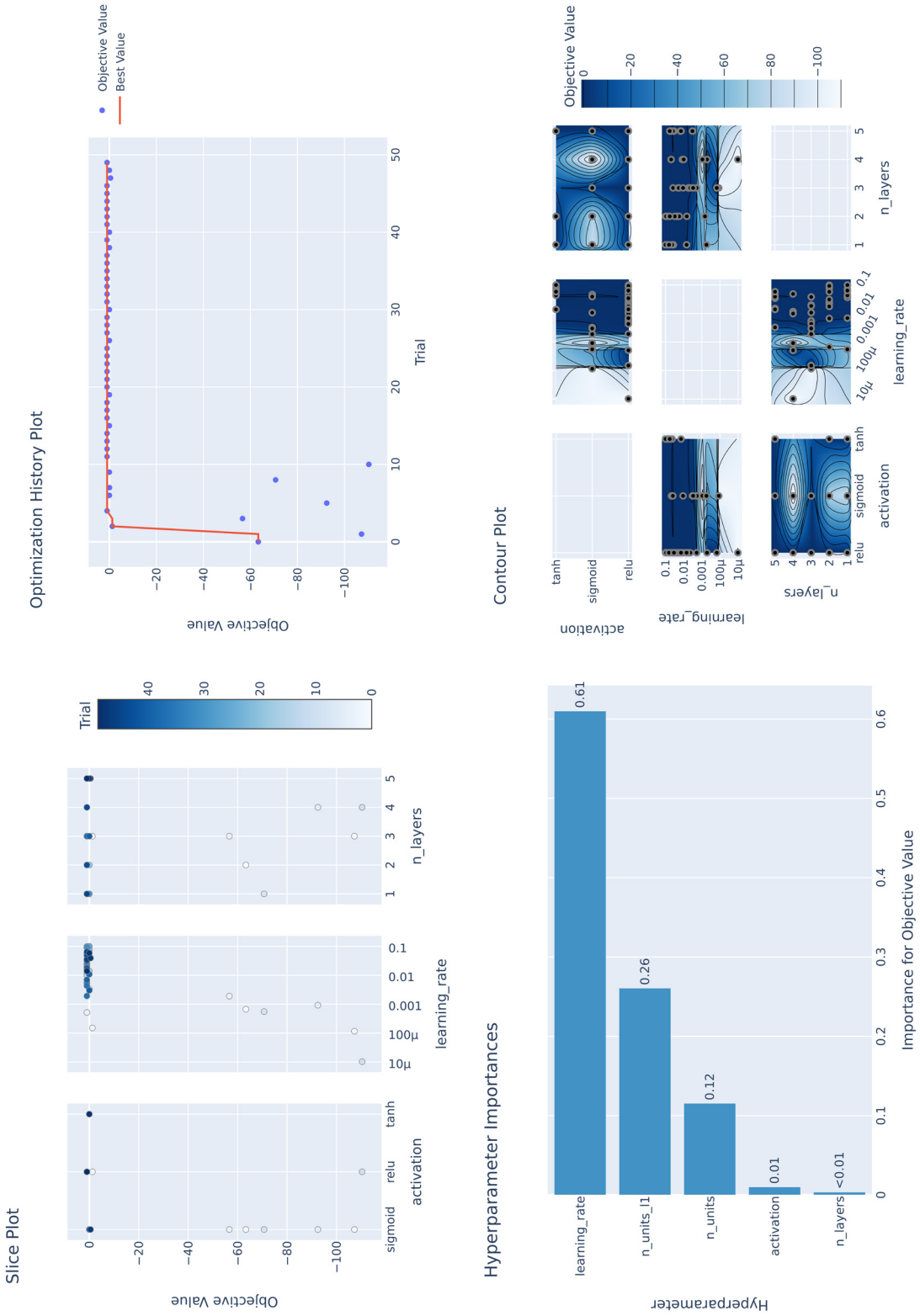

Figura 6.13: Optimización de hiperparámetros de MLP para T_i

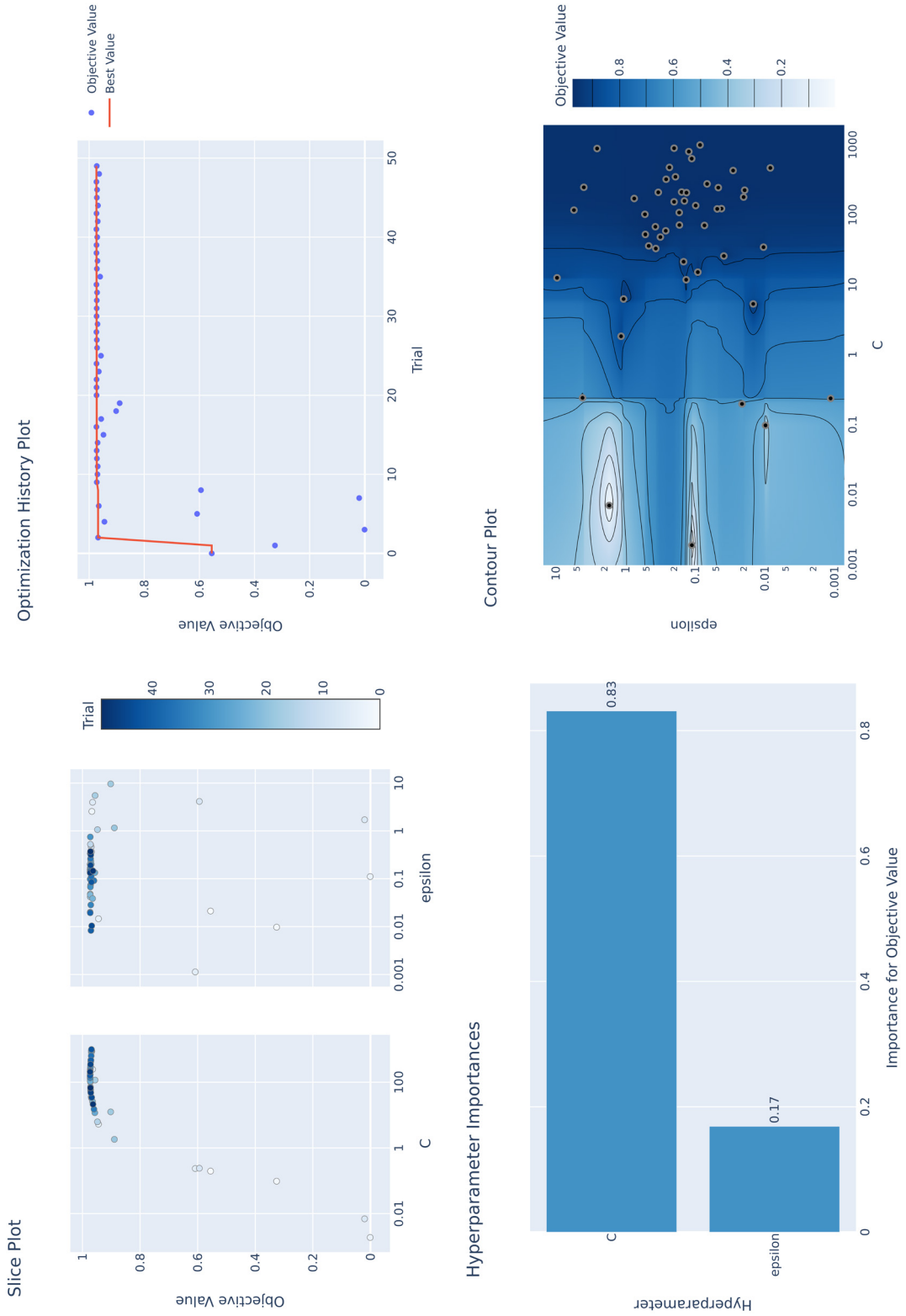

Figura 6.14: Optimización de hiperparámetros de SVR para T_i

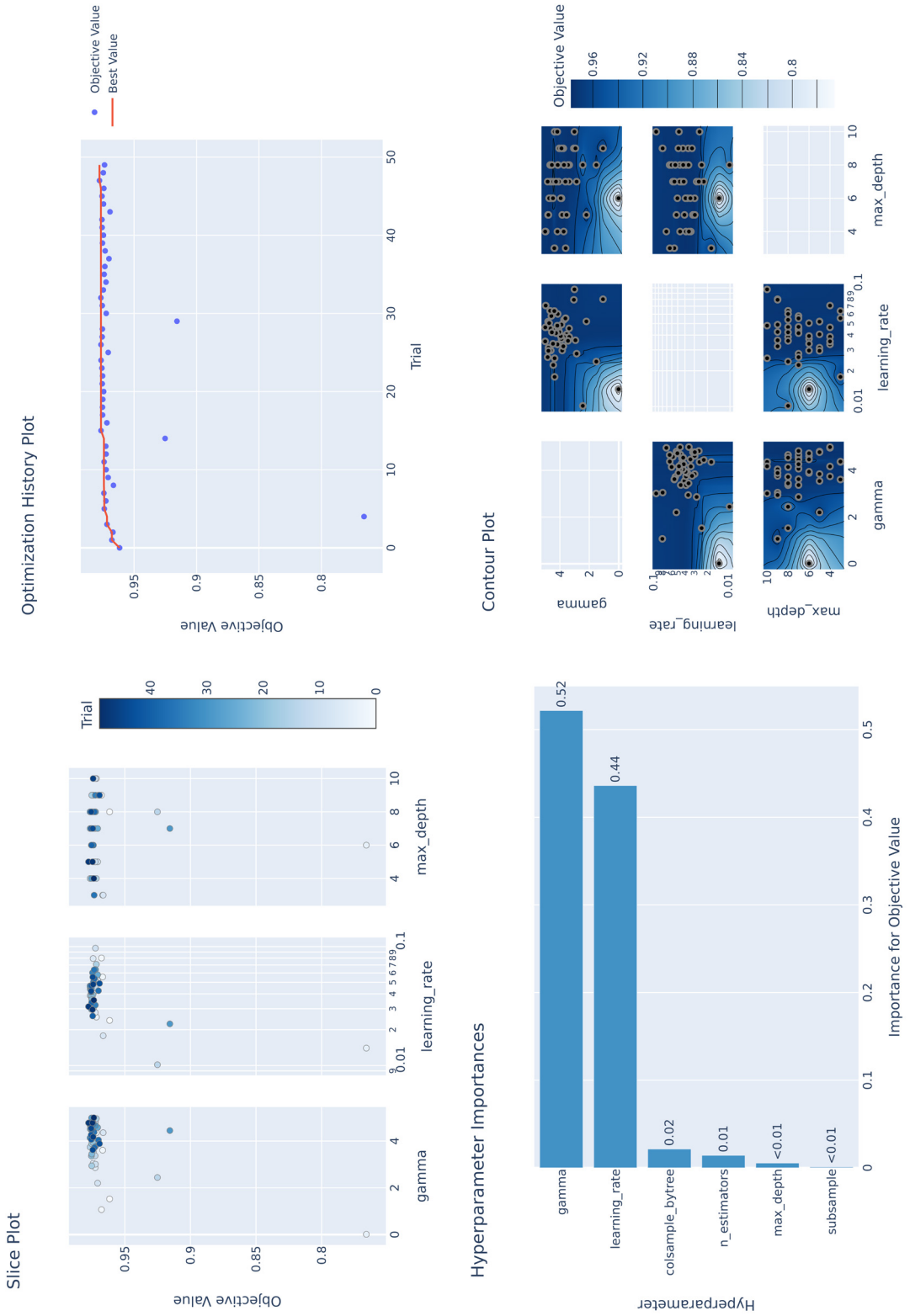

Figura 6.15: Optimización de hiperparámetros de XGBoost para T_i

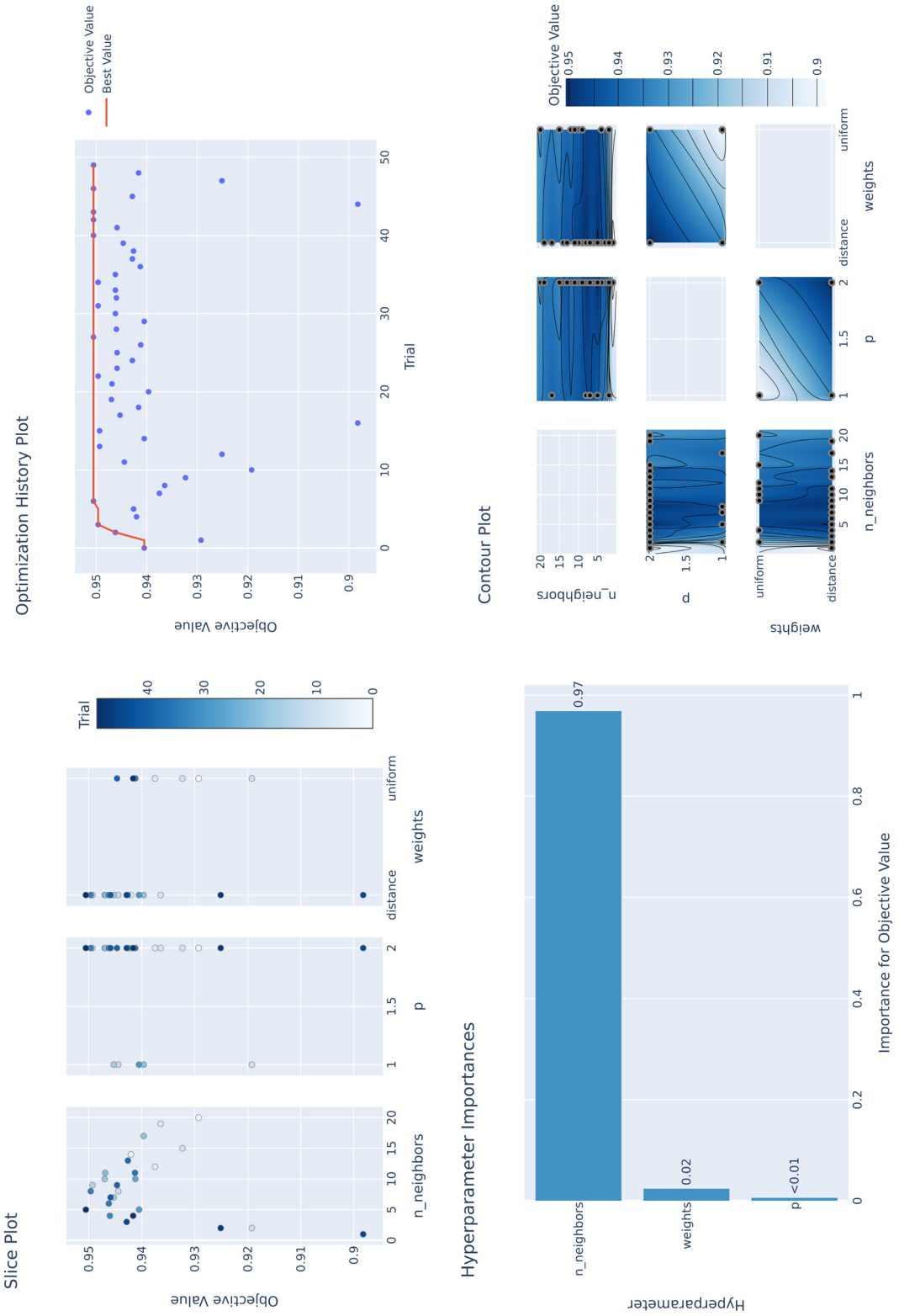

Figura 6.16: Optimización de hiperparámetros de kNN para T_i

A continuación, se entrena el metamodelo con el conjunto de datos de retención y, seguidamente, se procede a la evaluación y comparación para seleccionar la mejor

estrategia de modelado. La Tabla 6.3 contiene los valores de los hiperparámetros de los modelos base tras la optimización, así como datos de la evaluación de todos los modelos, mientras que la Figura 6.17 muestra gráficamente los valores reales de T_i en el conjunto de prueba contra los valores predichos por cada modelo. En estas gráficas se puede ver cómo igualmente para T_i, las predicciones de todos los modelos tienen un alto nivel de similitud con los valores esperados. Tras la comparación, la metodología seleccionó la estrategia de metamodelado también para representar a T_i. Es notable que esta estrategia haya sido la seleccionada para representar ambos KPIs, lo que confirma el acierto en la decisión de incluirla como parte de la metodología.

Tabla 6.3: Hiperparámetros seleccionados y resultados de validación para T_i

Modelo	**Hiperparámetros**	**R^2 validación**	**R^2 prueba**	**Tiempo de optimización de hiperparámetros**	**Tiempo de inferencia**
MLP	número de capas ocultas = 5, σ = relu, número de unidades por capa = [62, 53, 61, 95, 54, 5], $\propto$ = 0.034	0.974	0.968	539.614 s	103.078 ms
SVR	C= 157.350, ε = 0.142	0.974	0.968	7.832 s	2.995 ms
XGBoost	número de árboles = 487, profundidad máxima = 5, tasa de aprendizaje = 0.031, tasa de submuestreo = 0.648, tasa de submuestreo de columnas por árbol = 0.583, γ = 4.776	0.978	0.967	68.155 s	4.494 ms
*k*NN	k = 5, función de ponderación = distancia, orden de la distancia de Minkowski = 2	0.951	0.954	**0.494 s**	**1.220 ms**
Conjunto de modelos	-	0.979	0.975	-	129.488 ms
Metamodelo	-	**0.982**	**0.976**	0.001 s (entrenamiento metamodelo)	139.541 ms

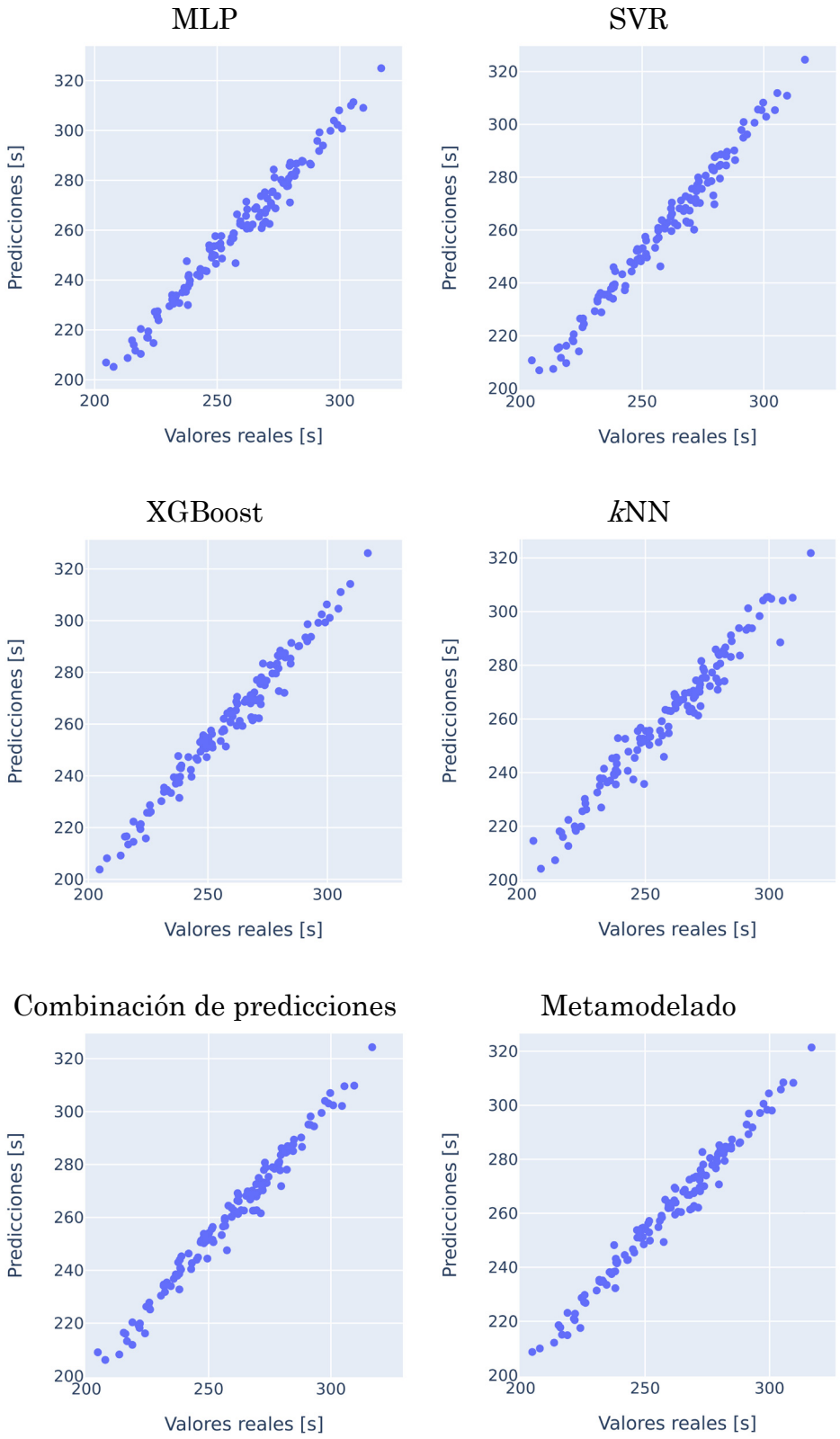

Figura 6.17: Valores reales vs. predicciones de los diferentes modelos para T_i

6.3.4. Optimización del proceso

Tras la selección de los modelos para los dos KPIs considerados en la línea piloto, se procede a la fase de optimización del proceso productivo. Inicialmente, se planteó llevar a cabo la optimización sin ninguna restricción extra a las fijadas durante el registro de los datos utilizados para crear el conjunto de datos, es decir, que las variables de entrada pudieran tomar valores en todo el rango explorado durante la captura de datos. De esta forma, los rangos de valores permitidos en la etapa de optimización son los que se definieron en la Tabla 6.1. Dado que internamente para la creación de los modelos se trabaja con valores normalizados, antes de iniciar la etapa de optimización estos intervalos también son normalizados utilizando los valores de media y desviación estándar calculados previamente. Durante la optimización, se trabaja con los valores de los parámetros normalizados y, cuando se va a generar la respuesta del método, se revierte la operación de normalización.

En la optimización se busca el conjunto de Pareto óptimo considerando sólo las variables de entrada seleccionadas para al menos un KPI. La búsqueda se realiza mediante el algoritmo NSGA-II con una población de 100 individuos y durante 5000 generaciones. La Figura 6.18 muestra el frente de Pareto obtenido tras finalizar la ejecución de la optimización multiobjetivo, en la cual se observa un amplio rango de soluciones distribuidas de manera continua. Además, se incluyen los resultados obtenidos al comprobar en la línea piloto tres parametrizaciones seleccionadas del conjunto de Pareto, dos de las cuales corresponden con los extremos del frente de Pareto y la tercera se ubica cerca del centro. La similitud de los resultados al realizar las pruebas en la línea piloto con respecto a los valores estimados por *Insigth Generator* permiten corroborar el adecuado funcionamiento del módulo. Por su parte, la Figura 6.19 muestra los valores de las variables de entrada seleccionadas que conforman el conjunto de Pareto obtenido[14]. De esta gráfica se puede inferir que de ciertos rangos de valores de las variables se obtienen soluciones subóptimas, como los rangos [3.8, 5.0] m/s y [6.8, 7.7] m/s en el caso de la variable v_s, el rango [33000, 49000] mm/min en el caso de la variable a_d, el rango [62, 70] % para la variable v_u y el rango [8400, 10400] mm/min para la variable a_k. Además, se pueden apreciar determinados patrones en las combinaciones de los valores de los parámetros. Por ejemplo, para los valores de la variable v_s en el rango [5.0, 6.7] m/s todas las combinaciones incluyen valores cercanos a los límites

[14] En esta figura y otras similares se incluye además el valor de un KPI para facilitar la comprensión de la influencia de los valores de los parámetros en las salidas

superiores de las variables a_d y a_k. Por otra parte, las combinaciones que incluyen valores en el rango [27000, 33000] mm/min de la variable a_d fuerzan que las variables v_s, v_u y a_k tomen valores cercanos a sus respectivos límites inferiores.

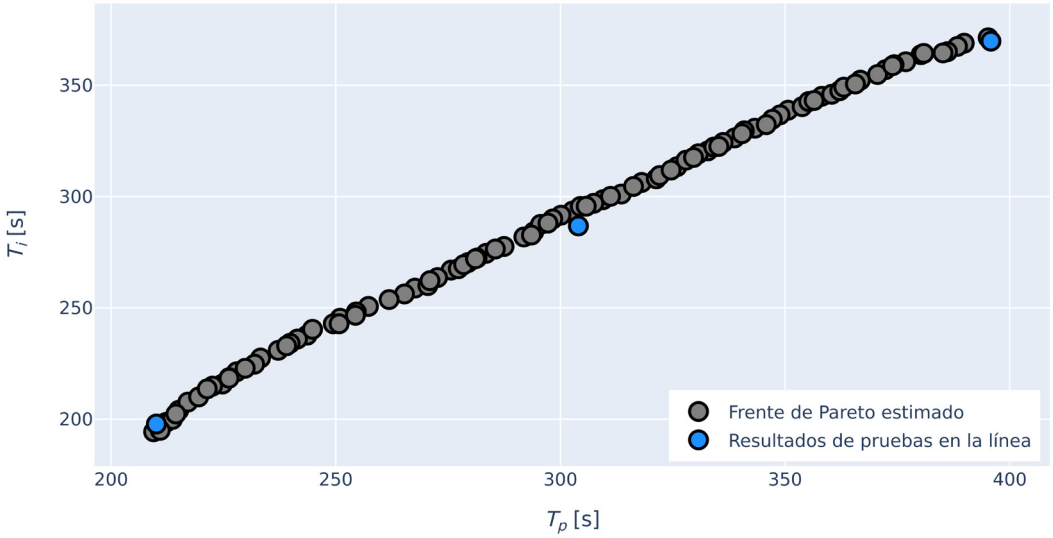

Figura 6.18: Frontera de Pareto estimada por la metodología sin restricciones adicionales

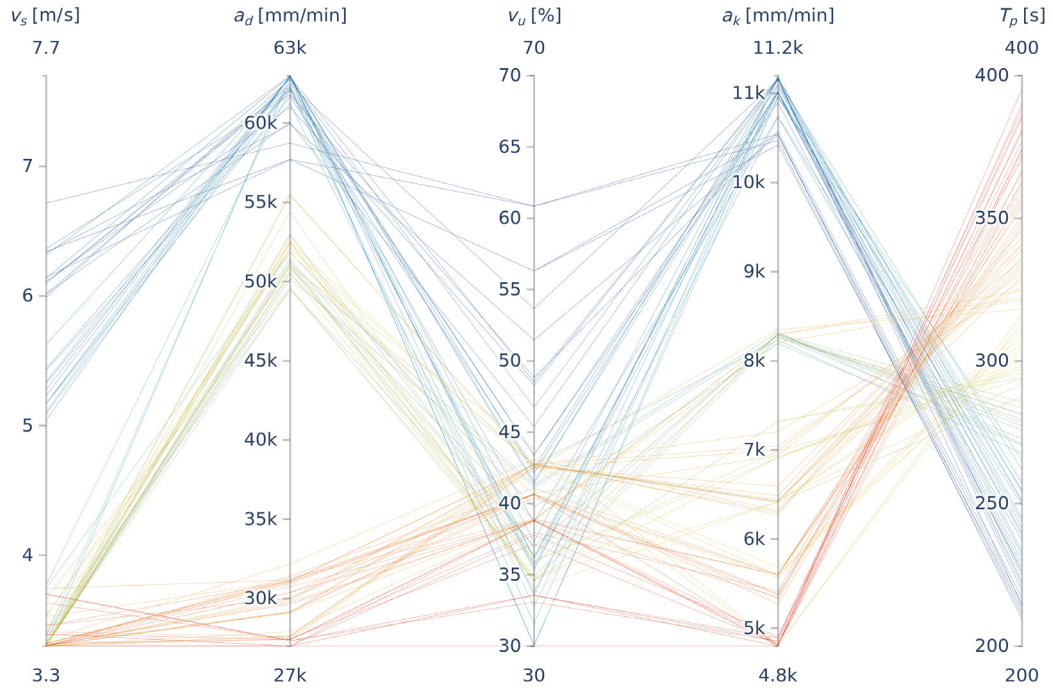

Figura 6.19: Conjunto de Pareto para las variables seleccionadas sin restricciones adicionales

A continuación, se decidió ejecutar nuevamente el proceso de optimización, pero esta vez introduciendo una restricción adicional. Esta prueba se realizó para simular escenarios de producción donde se debe cumplir con algún criterio de calidad y para lograrlo se fijan ciertos parámetros de los equipos de acuerdo a alguna norma o a la experticia de los operadores. La restricción adicional definida en este caso fue establecer el valor de a_d en el mínimo posible dentro del rango definido previamente, es decir, 27000 mm/min. La Figura 6.20 muestra el frente de Pareto obtenido, donde se puede observar como la amplitud del frente de Pareto se redujo en los dos KPIs hacia las zonas donde se obtienen mayores valores. En este caso, también se puede apreciar la similitud entre los valores resultantes del proceso de optimización y los resultados obtenidos al evaluar tres parametrizaciones en la línea. La reducción del frente de Pareto hacia las zonas con mayores valores de los KPIs también se confirma en la Figura 6.21, donde se muestran los valores de las variables de entrada seleccionadas que conforman el conjunto de Pareto obtenido y su efecto en T_p. Estos resultados coinciden con la intuición de que al reducir el valor del parámetro de control de la velocidad de trabajo de algún elemento de la línea aumentará el tiempo necesario para producir un panel y, consecuentemente, también el tiempo de inactividad del operador.

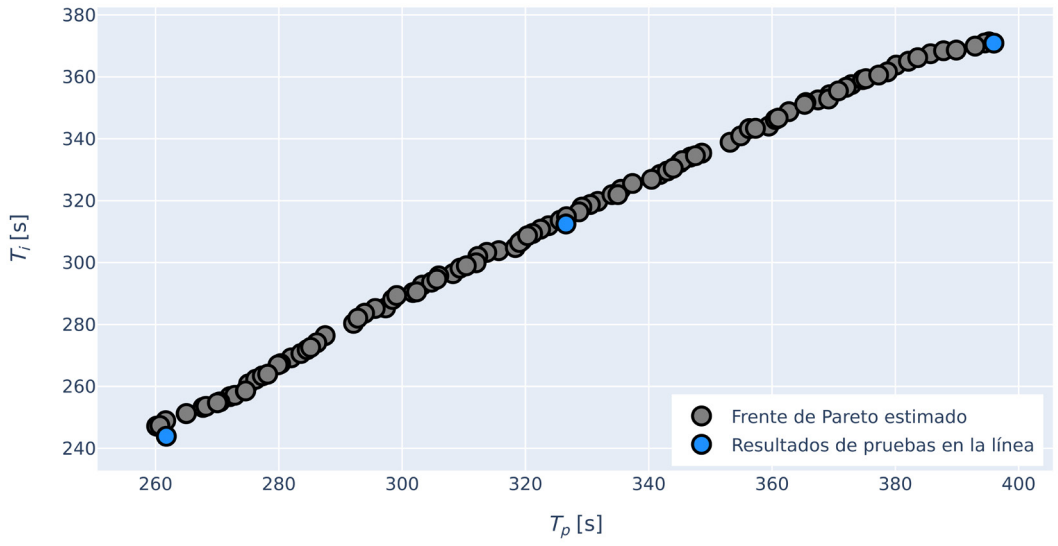

Figura 6.20: Frontera de Pareto estimada por la metodología con la restricción a_d = 27000 mm/min

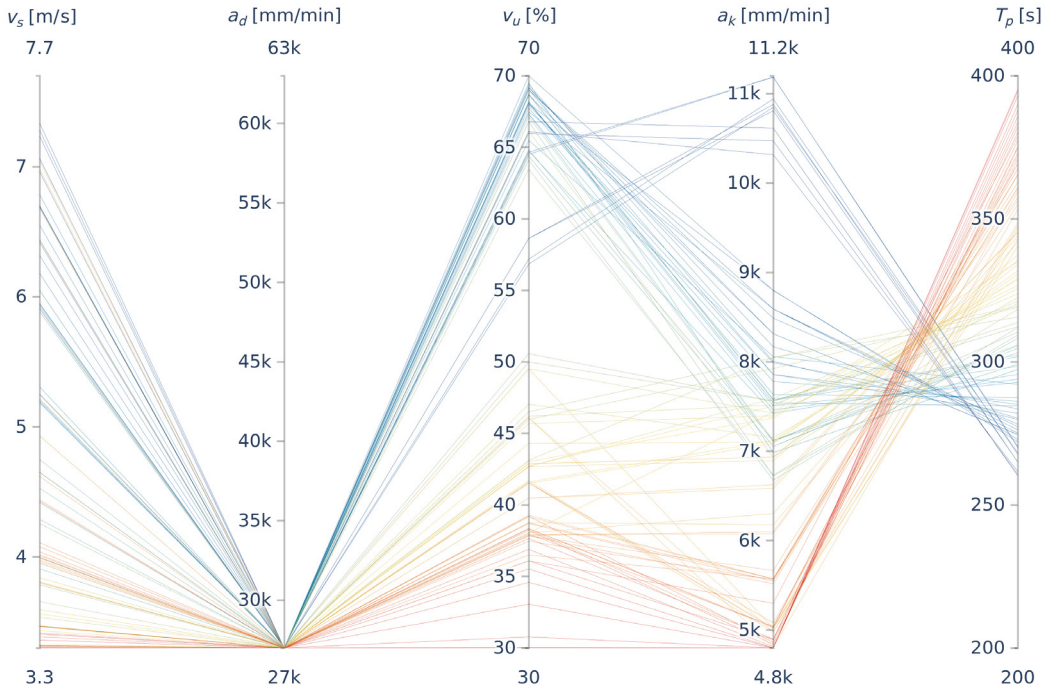

Figura 6.21: Conjunto de Pareto para las variables seleccionadas con la restricción a_d = 27000 mm/min

Por último, se decidió volver a ejecutar el proceso de optimización, pero esta vez introduciendo dos restricciones adicionales. Las restricciones definidas en este caso fueron establecer los valores de v_s y v_u en los máximos posibles dentro de sus respectivos rangos definidos previamente, es decir, $v_s = 7.7$ m/s y $v_u = 70\%$. La Figura 6.22 muestra el frente de Pareto obtenido en este caso, así como los resultados de tres evaluaciones en la línea piloto, corroborándose nuevamente la similitud entre estos valores y los resultados del proceso de optimización. La amplitud del frente de Pareto obtenido nuevamente se reduce, pero en este caso con una tendencia hacia zonas más cercanas a donde los dos KPIs toman menores valores, lo cual se evidencia en la Figura 6.23. Estos resultados también eran previsibles, ya que al parametrizar dos equipos para que trabajen a mayor velocidad, el tiempo de producción de un panel se reducirá, lo que conlleva que el tiempo de inactividad del operador también disminuya.

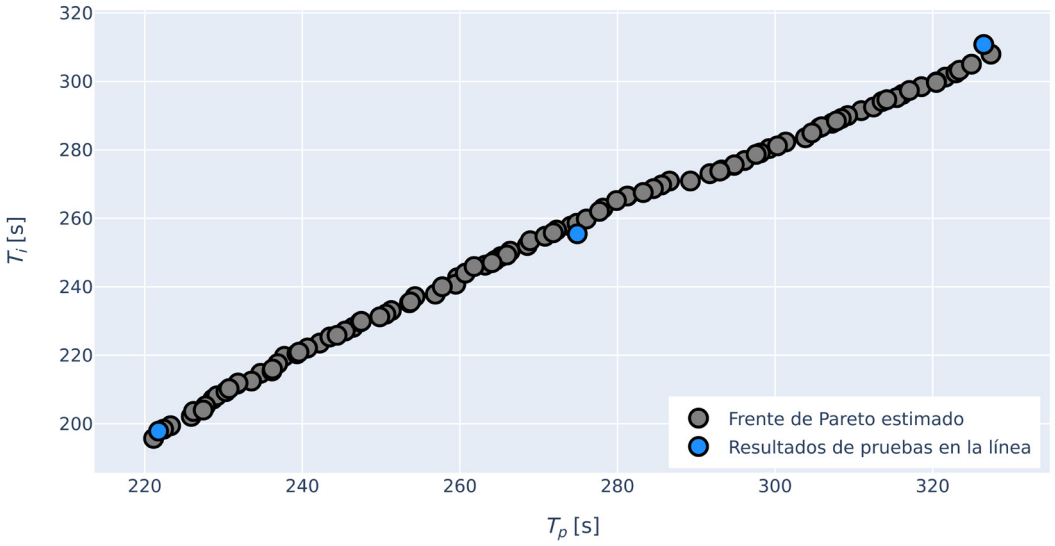

Figura 6.22: Frontera de Pareto estimada por la metodología con las restricciones v_s = 7.7 m/s y v_u = 70%

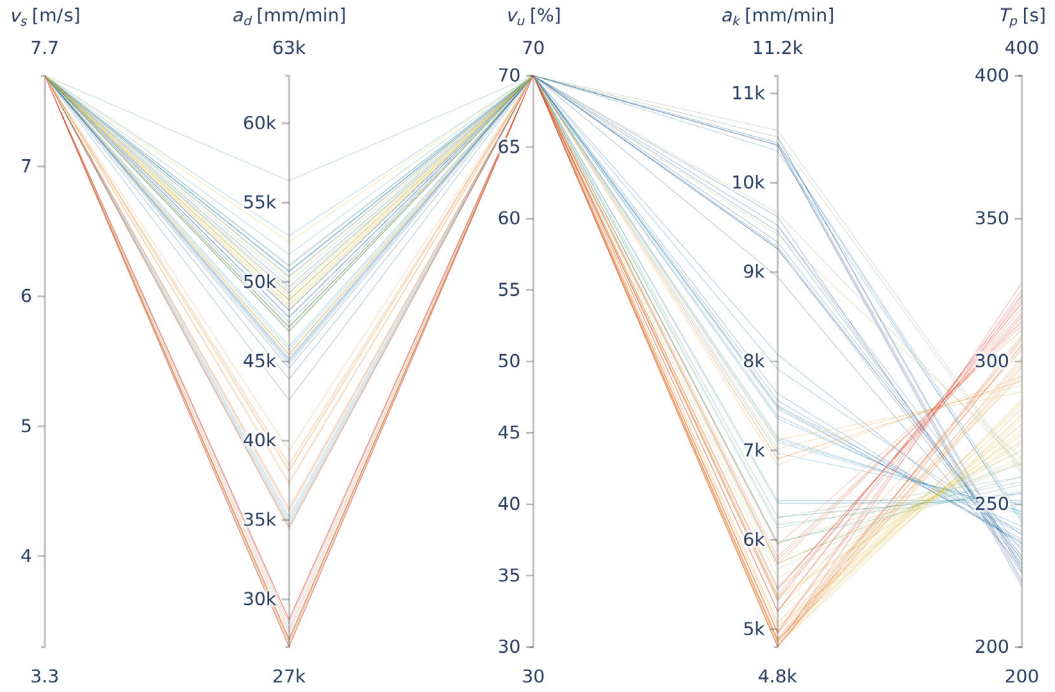

Figura 6.23: Conjunto de Pareto para las variables seleccionadas con las restricciones v_s = 7.7 m/s y v_u = 70%

6.4. Supervisión automática del sistema ciberfísico. Resultados

Además de evaluar el funcionamiento del módulo de aprendizaje automático automatizado de la plataforma KITT4SME, para garantizar que el gemelo digital sea capaz de coevolucionar con el proceso productivo, es necesario que la funcionalidad de supervisión se ejecute correctamente. Esta funcionalidad es de vital importancia, ya que en la línea piloto pueden ocurrir derivas de datos o de concepto, provocando que el desempeño de los modelos disminuya y, por tanto, los resultados proporcionados inicialmente por *Insight Generator* dejen de ser válidos. Si esto ocurriese se deberían reentrenar los modelos y obtenerse una nueva parametrización para el proceso productivo. Sin embargo, también es importante que la funcionalidad de supervisión no sea altamente sensible, ya que esto la haría muy propensa a las perturbaciones y el ruido, desencadenando innecesariamente el reentrenamiento de los modelos, el cual es un proceso de alto coste computacional.

Para la comprobación de esta funcionalidad se realizaron varias pruebas donde se establecieron los valores para los parámetros extraídos del conjunto de Pareto obtenido sin restricciones adicionales, los cuales se muestran en la Tabla 6.4.

Tabla 6.4: Parametrización utilizada en las pruebas de supervisión

Variable	Valor
p [cm]	100
v_s [m/s]	5.17
a_d [mm/min]	62870.07
v_d [rpm]	6550
v_c [rpm]	3150
v_u [%]	30.02
a_k [mm/min]	11063.61

Una vez que el sistema se encontraba trabajando bajo esta parametrización, se inyectaron datos para simular diferentes derivas de concepto en T_p. La primera prueba consistió en simular una deriva de concepto repentina tras procesarse el panel número 100. A los valores de T_p medidos a partir de dicho panel se le añadieron 4.39 s de manera sostenida. Estos datos fueron procesados tal como se describió en el Capítulo 4, utilizando ADWIN. En este caso, en el cual el cambio ocurrió de manera abrupta, el algoritmo requirió que se procesaran 13 paneles

antes de detectar la deriva, tal como se puede apreciar en la Figura 6.24. Este resultado permite corroborar que ADWIN admite cierto margen de insensibilidad antes de detectar la deriva, tal como se desea. En el momento de detección de la deriva se inició el reentrenamiento de los modelos y la ejecución del proceso de optimización, obteniéndose los resultados de la nueva parametrización para ser aplicados tras el procesamiento de 9 paneles más.

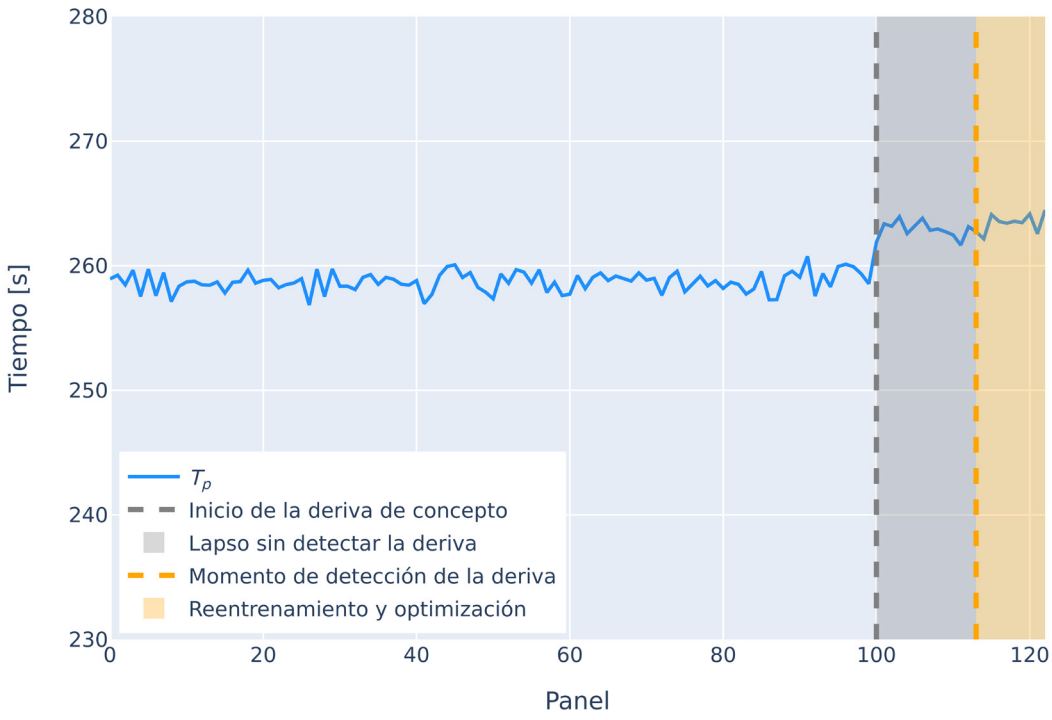

Figura 6.24: Detección de una deriva de concepto repentina

Los resultados de las pruebas descritas en esta sección fueron descartados directamente sin ser aplicados en la línea piloto ya que la reparametrización se realizó en base a una deriva de concepto simulada, no real. Tras cada una de las pruebas los modelos fueron restaurados a su estado inicial. Las operaciones llevadas a cabo por *Insight Generator* solo se ejecutaron con el fin de determinar el tiempo que transcurriría durante el reentrenamiento y la optimización en una situación real y el número de paneles que se producirían en ese intervalo. En la primera prueba, desde que se inició la deriva hasta que se obtuvo la reparametrización se produjeron un total de 22 paneles.

Para la segunda prueba se simuló una deriva de concepto incremental a partir del panel número 100. En este caso, a los valores de T_p medidos a partir de dicho panel

se le adicionaron 0.16 s de manera incremental por cada panel procesado, es decir, 0.16 s a la primera medición, 0.32 s a la segunda, etc. Frente a este tipo de deriva, donde la dinámica es menos abrupta que en el caso anterior, ADWIN requirió que se procesaran 23 paneles antes de poder detectarla, tal como se muestra en la Figura 6.25. Al igual que en la prueba anterior, durante el tiempo necesario para el reentrenamiento de los modelos y la ejecución del proceso de optimización se produjeron 9 paneles más, por lo que en total desde que se inició la deriva hasta que se generaron los nuevos parámetros se produjeron 32 paneles.

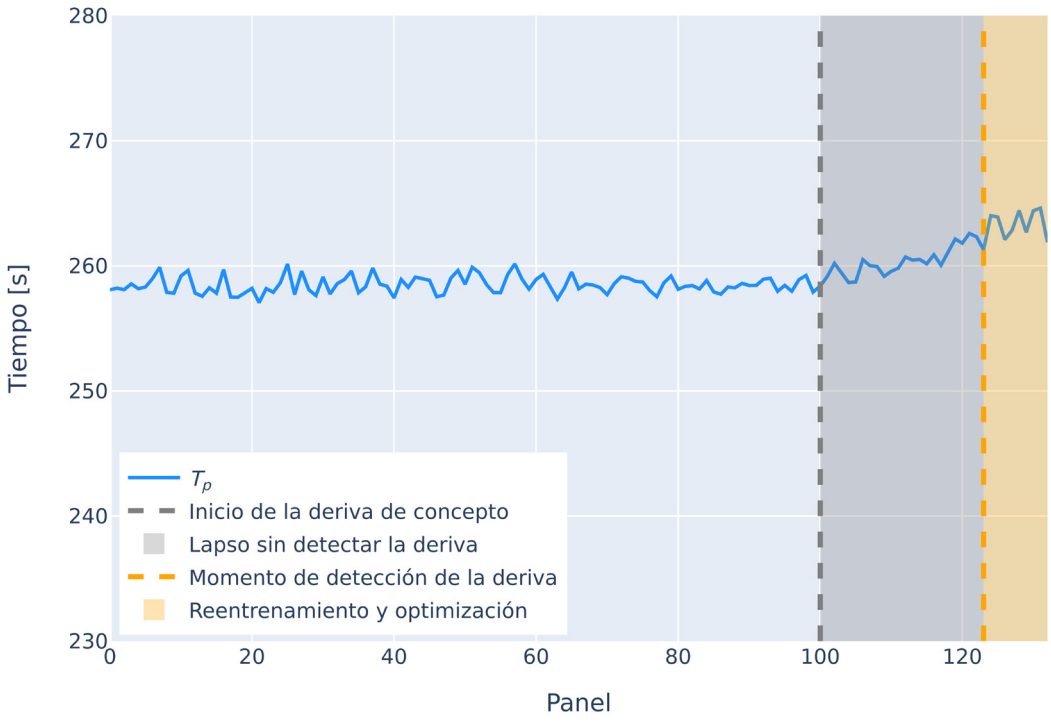

Figura 6.25: Detección de una deriva de concepto incremental

Por último, se realizó una tercera prueba en la que se simuló una deriva de concepto gradual a partir del panel número 100. Para esto, a los valores de T_p medidos a partir de dicho panel se le adicionaron 4.39 s de manera intermitente, hasta que a partir del panel número 140 se adicionaron los 4.39 s de manera sostenida. En esta prueba se procesaron 51 paneles antes de que ADWIN detectara la deriva, tal como se muestra en la Figura 6.26. El hecho de que en este caso se hayan procesado más del doble de paneles desde el comienzo de la deriva hasta su detección que en la segunda prueba y más del triple que en la primera se debe a la naturaleza intermitente de la deriva, la cual provoca que la diferencia de los

promedios de las ventanas analizadas se mantenga inferior al umbral de detección durante más tiempo. Por otra parte, durante el reentrenamiento y la ejecución de la optimización nuevamente se produjeron 9 paneles, para un total de 60 paneles desde el comienzo de la deriva hasta la obtención de la nueva parametrización.

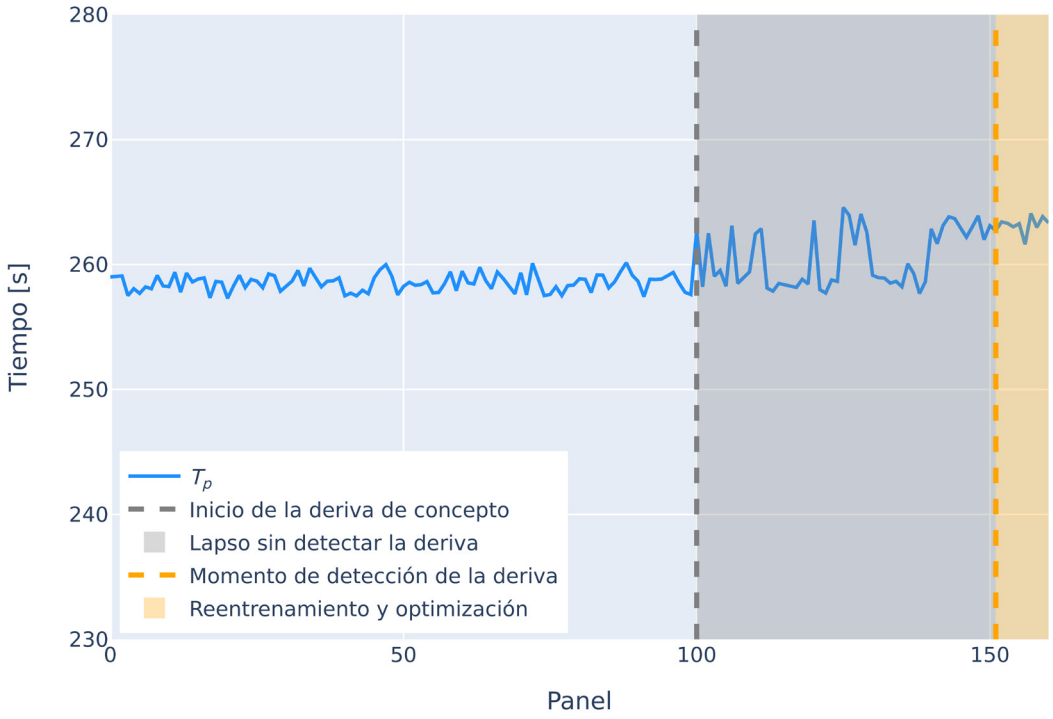

Figura 6.26: Detección de una deriva de concepto gradual

6.5. Interfaz gráfica del gemelo digital

Uno de los componentes propuestos para el desarrollo del gemelo digital consiste en la interfaz de visualización en FlexSim. Conociéndose los diferentes equipos que forman parte del proceso de producción y la secuencia en que se ejecutan las operaciones se puede realizar una réplica tridimensional del escenario utilizando elementos predefinidos en esta herramienta. En este caso, los dos robots presentes en la línea fueron representados mediante ejecutores de tareas de tipo robot de 6 ejes o grados de libertad, los cuales se utilizan para la manipulación de objetos. El operador, fue representado mediante un ejecutor de tareas tipo operador, el cual además de manipular objetos puede desplazarse e imitar el comportamiento humano. Mientras que el resto de los elementos de la línea fueron representados mediante recursos fijos de tipo procesador. Estos recursos se usaron para replicar

el flujo de proceso. Debido a que los dos KPIs analizados en la tesis se miden a nivel de proceso y no de equipo, durante las simulaciones los tiempos de procesamiento y transferencia desde un elemento a otro no se predicen de manera individual. En este caso, sus tiempos de actuación se calculan como un por ciento del valor de T_p predicho. El porcentaje correspondiente a cada elemento se determinó configurando todos los equipos para trabajar en su capacidad media y se calculó que fracción del tiempo necesario para producir un panel correspondía a cada uno. La Figura 6.27 muestra una captura de la simulación 3D de la línea piloto de GAMHE creada en FlexSim.

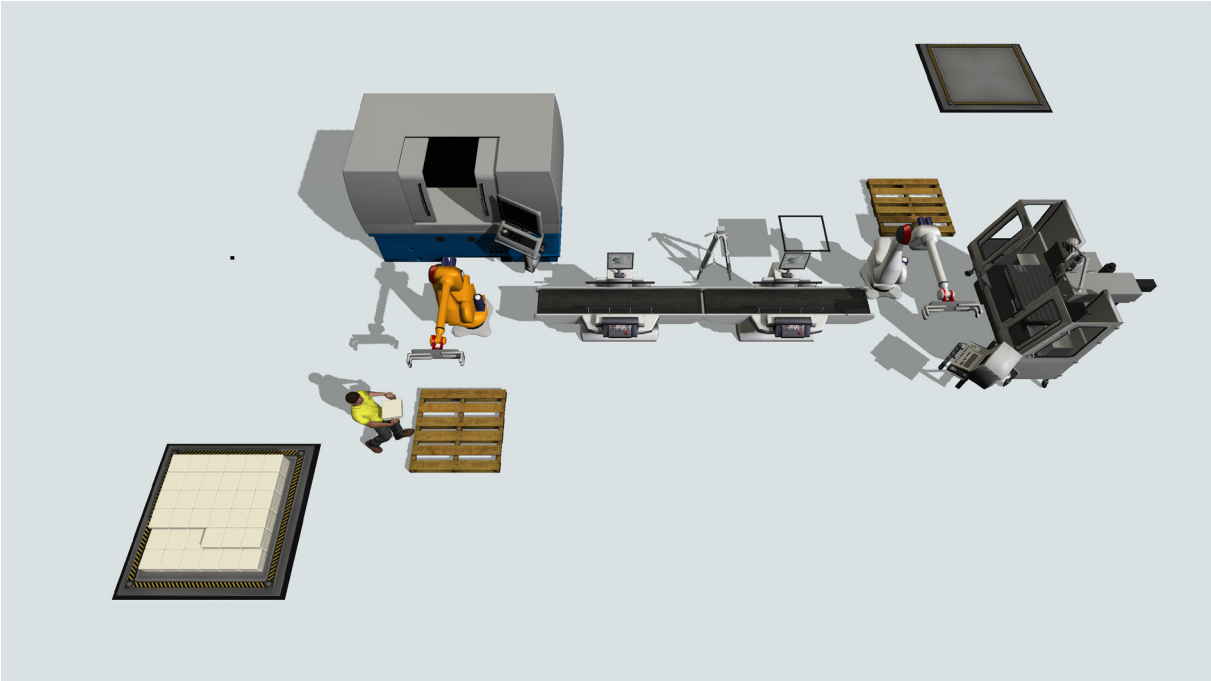

Figura 6.27: Captura de la interfaz gráfica del gemelo digital de la línea piloto creada en FlexSim

En la interfaz visual del gemelo digital no se calcula el tiempo para cada elemento de manera individual ya que esto requeriría tener un modelo para cada elemento. Aunque este enfoque es realizable si se define un KPI para el tiempo de procesamiento de cada elemento de la línea, esta una solución mucho menos eficiente computacionalmente, ya que no habría que entrenar ni evaluar los modelos para representar a T_p, pero en su lugar se requeriría realizar estas operaciones en 7 ocasiones (una por cada elemento con KPI individual y otra para T_i). De esta manera, si se asume que el tiempo de ejecución de la etapa de creación

de los modelos, optimización de hiperparámetros y selección, crece de manera lineal con el número de KPIs, entonces de esta forma tomaría más del tripe del tiempo que con el enfoque utilizado en la tesis. De igual manera, la etapa de optimización multiobjetivo pasaría de tener 2 funciones objetivo a tener 7, complejizando mucho más su solución y, por ende, propiciando que la convergencia sea potencialmente mucho más lenta. Se debe aclarar que el enfoque adoptado en la tesis es adecuado debido a las características del proceso productivo, en el cual no se comienza a procesar un panel hasta que no se haya terminado con el anterior, por tanto, aunque en la interfaz visual no se calcule el tiempo para cada elemento individualmente, esto no tiene ninguna repercusión en el flujo de trabajo del proceso.

6.6. *Reco*: Aplicación y resultados

Tras haber concluido la evaluación experimental del primer módulo desarrollado para la arquitectura KITT4SME utilizado en la línea piloto y de la funcionalidad de supervisión, se prosiguió con la evaluación experimental del segundo módulo propuesto para integrar en la arquitectura KITT4SME: *Reco*. Debido a ciertas características del proceso productivo que se lleva a cabo en la línea como son el tipo de material del que están compuestas las piezas que se procesan (poliestireno extruido), cuyo corte se lleva a cabo con relativa facilidad o la baja incidencia de perturbaciones en el proceso, la productividad tiende a mantener una alta estabilidad. Por esta razón, para poder evaluar la capacidad de mantener la productividad, corrigiendo las desviaciones mediante el uso del módulo desarrollado, fue necesario simular las desviaciones. Para realizar las pruebas se vincularon todos los equipos de la línea piloto al servicio ofrecido por el módulo *Reco*. La Tabla 6.5 muestra la parametrización utilizada al inicio de cada prueba, y especifica que parámetro de cada equipo se manipulará en caso de que *Reco* determine que es necesario actuar para corregir la desviación en la productividad. Además, se presentan los tiempos nominales en los que cada equipo debe procesar una unidad del producto bajo el régimen de trabajo establecido con la parametrización inicial especificada. Los tiempos nominales presentados son los resultantes de promediar tres mediciones del tiempo de procesamiento de una unidad del producto bajo la misma parametrización.

Tabla 6.5: Parametrización utilizada en las pruebas de corrección de desviaciones en la productividad

Equipo	Variable	Valor inicial	Variable vinculada a *Reco*	T_a
-	p	100 cm	No	-
Stäubli RX90 (A_1)	v_s	5.5 m/s	Sí	29.3 s
Deckel Maho (A_2)	a_d	45000 mm/min	Sí	73.7 s
	v_d	9000 rpm	No	
Cintas transportadoras (A_3)	v_c	3600 rpm	Sí	15.8 s
UR5 (A_4)	v_u	50%	Sí	54.9 s
KERN Evo (A_6)	a_k	8000 mm/min	Sí	82.5 s
Total	-	-	-	256.2 s

Se debe recalcar que el servicio de corrección de desviaciones en la arquitectura KITT4SME no utiliza los valores de los parámetros, este solo requiere los tiempos nominales y reales de procesamiento de un producto para cada equipo analizado. Además, las salidas deben ser interpretadas para adaptarlas al proceso productivo. En este caso, se han utilizado los dos formatos de salida para automatizar la toma de decisiones. La salida $RECO_{1_nom}$ se utiliza para determinar si se debe realizar alguna modificación en el equipo. Si $RECO_{1_nom}$ es $Parar\ 1$ o $Parar\ 2$ se detiene el proceso productivo; si $RECO_{1_nom}$ es $Mantener$ no se realiza ninguna modificación y si $RECO_{1_nom}$ es $Reducir$ o $Aumentar$, se modifica el parámetro de control del equipo en dependencia del valor de $RECO_1$. En caso de que se requiera modificar algún equipo, el valor de $RECO_1$ no se puede utilizar directamente ya que está normalizado, por lo que debe ser procesado para determinar la magnitud de la modificación. El enfoque escogido para adaptar esta salida a los equipos de la línea piloto fue el de escalar el valor proporcionado por $RECO_1$ al rango comprendido entre los valores mínimos y máximos establecidos en la Tabla 6.1. De esta forma, si se están analizando las cintas transportadoras y $RECO_1 = -1$, la velocidad del motor se establecerá a 2160 rpm; y si, por el contrario $RECO_1 = 1$, la velocidad del motor se establecerá a 5040 rpm.

Se debe notar que, a pesar de que los robots puedan variar las velocidades de cada una de sus articulaciones de manera independiente, si en las pruebas realizadas se requiere modificar el comportamiento de algún robot, todas las velocidades de sus articulaciones se modifican simultáneamente utilizando los parámetros v_s y v_u

para A_1 y A_4, respectivamente. Esto quiere decir que, por ejemplo, si se requiere disminuir el tiempo de procesamiento de una pieza por parte del robot Stäubli RX90, lo cual implica aumentar la velocidad de las articulaciones, cada una de estas velocidades será incrementada en la misma proporción dentro de su respectivo rango.

Para realizar las pruebas de *Reco* se simularon derivas de concepto de manera similar a como se hizo en las pruebas de la funcionalidad de supervisión, solo que, en este caso, al registrarse los tiempos de procesamiento de cada elemento de la línea de manera individual, las derivas se añadieron directamente a los tiempos de procesamiento de los elementos y no al tiempo total de procesamiento para una unidad del producto de la línea piloto. Además, en cada una de las pruebas se simularon perturbaciones instantáneas adicionales para determinar si *Reco* era capaz de filtrarlas. De esta forma, tras la parametrización del proceso productivo y su puesta en marcha, se realizaron cuatro pruebas para evaluar la capacidad de *Reco* de mantener la productividad mediante la corrección de las desviaciones.

En la primera prueba, se le aplicó una perturbación instantánea al tiempo de procesamiento de las cintas transportadoras para el noveno panel producido. Esta consistió en adicionarle 1.26 s al valor del tiempo medido para el procesamiento de ese panel, lo que representa un 8% del valor de tiempo promedio para este elemento al inicio de la prueba. Como se puede ver en la Figura 6.28, tras procesar este panel el equipo continuó trabajando en condiciones estables. En este caso, *Reco* no actuó ya que la desviación ocurrió solamente para el procesamiento de un panel. Además, en esta prueba se inyectó una desviación del tiempo de procesamiento al centro de mecanizado Deckel Maho a manera de deriva de concepto repentina a partir del panel número 100. La deriva consistió en adicionarle 5.9 s al tiempo medido para este elemento de la línea piloto, lo cual también representa un 8% del valor de tiempo promedio al inicio de la prueba. Tras procesar los datos del segundo panel una vez inyectada la deriva, el tiempo de procesamiento de este equipo regresó a los valores previos a la desviación. En este caso, esto se logró gracias a la actuación de *Reco*, que permitió corregir la desviación en la productividad del centro de mecanizado aumentando el valor de a_d a 48892 mm/min.

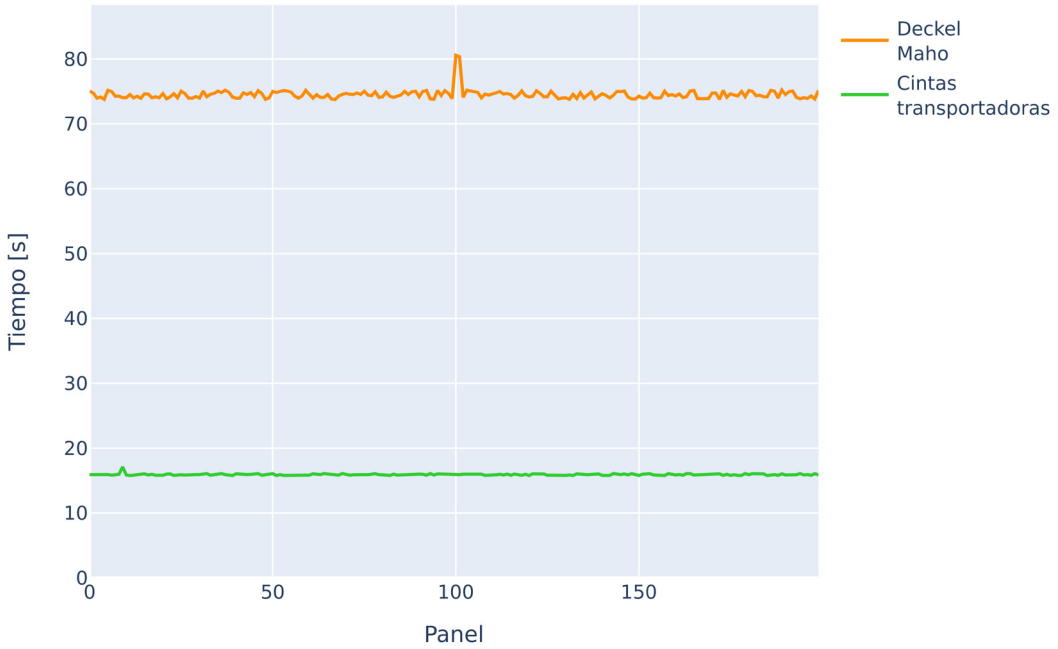

Figura 6.28: Comportamiento frente a una perturbación instantánea en las cintas transportadoras y una deriva de concepto repentina en el centro de mecanizado

En la segunda prueba la perturbación instantánea se le aplicó a la fresadora CNC KERN Evo restándole 4.12 s al tiempo de procesamiento para el panel número 15, lo que representa un 5% menos del valor de tiempo promedio para este elemento al inicio de la prueba. Debido a las características de esta perturbación instantánea, no fue necesario que *Reco* actuara para que la fresadora regresara a su estado de operación previo tras procesar este panel, como puede observarse en la Figura 6.29. En esta prueba también se ensayó una deriva incremental al tiempo de procesamiento del robot UR5e, adicionándole a este tiempo 0.16 s de manera acumulativa por cada panel procesado después del panel número 100. *Reco* intentó corregir esta desviación en varias ocasiones, hasta que, en cierto punto se decidió detener el proceso ya que el UR5e presentó retardos considerables en la manipulación de los paneles. La primera corrección ocurrió tras haberse procesado 18 paneles después del comienzo de la inyección de desviación en la productividad, para lo cual se incrementó el valor de v_u a 52.5%, logrando restaurar la productividad a valores previos a la deriva. Debido a que el valor de desviación inyectado continuó aumentando, *Reco* intentó corregirla nuevamente tras haber procesado 13 paneles después de la primera corrección, incrementando el valor de v_u a 55.3% en esta ocasión. Este comportamiento se repitió varias veces, teniendo lugar la tercera corrección tras haberse procesado 20 paneles después de la

segunda corrección e incrementando v_u a 58.9%; la cuarta corrección tras haberse procesado 16 paneles después de la tercera corrección e incrementando v_u a 62.0%; la quinta corrección tras haberse procesado 14 paneles después de la cuarta corrección e incrementando v_u a 65.2%; la sexta corrección tras haberse procesado 12 paneles después de la quinta corrección e incrementando v_u a 68.8% y la séptima corrección tras haberse procesado 15 paneles después de la sexta corrección e incrementando v_u a 70.0%. A partir de este punto la desviación continuó aumentando sin que se pudiera seguir aumentando el valor de v_u, ya que se había alcanzado el límite superior del rango definido, hasta que después de haberse procesado 19 paneles en estas condiciones la salida de *Reco* indicó que debía realizarse una parada.

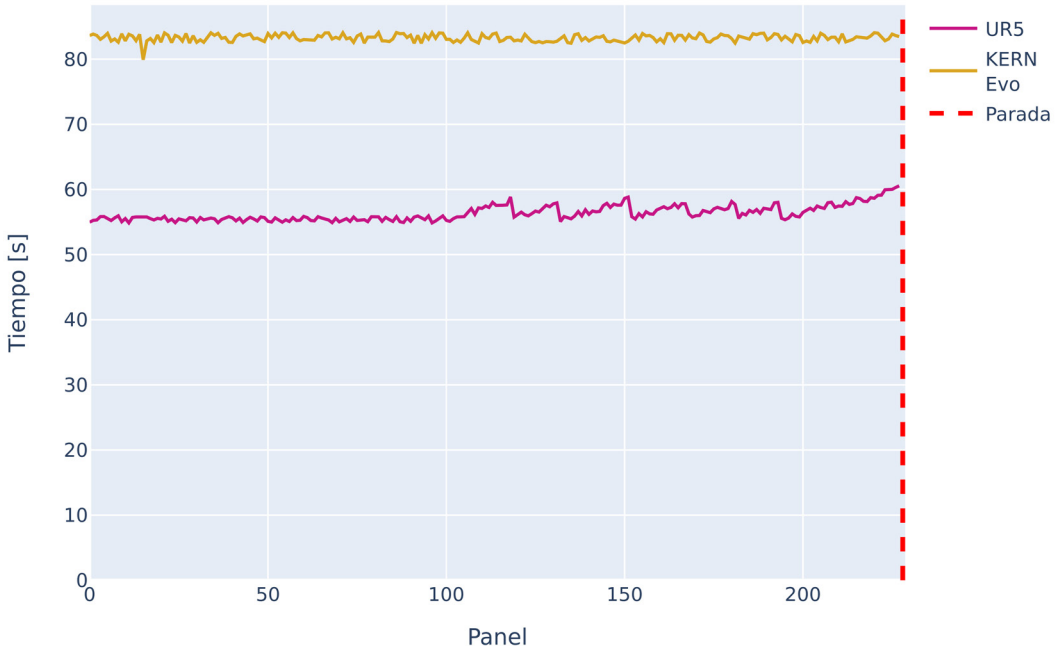

Figura 6.29: Comportamiento frente a una perturbación instantánea en la fresadora CNC y una deriva de concepto incremental en el robot UR5e

En la tercera prueba se aplicaron dos perturbaciones instantáneas, una al robot Stäubli RX90 en la que se le adicionó 1.75 s al tiempo medido para el panel número 29, lo que representa un aumento del 6% del tiempo promedio para este elemento, y la otra a la fresadora CNC KERN Evo restándole 5.77 s al tiempo de procesamiento para el panel número 63, lo que representa un 7% menos del valor de tiempo promedio para este elemento al inicio de la prueba. Ante estas dos perturbaciones *Reco* no tuvo que intervenir y los dos equipos regresaron a su

régimen de operación previo inmediatamente, tal como se puede apreciar en la Figura 6.30. Por otra parte, en esta prueba se ensayó inyectar una deriva gradual al robot UR5e a partir del panel número 100. De este modo se añadieron 4.39 s a las mediciones del tiempo de procesamiento de este robot de manera intermitente, lo que representa un aumento del 8% con respecto a su tiempo promedio. A partir del panel número 140 se continuó añadiendo esta cantidad al tiempo de procesamiento medido, pero de manera permanentemente. Ante esta deriva, *Reco* actuó en varias ocasiones. La primera fue tras haberse procesado 11 paneles después del inicio de la deriva, donde se incrementó v_u a 54.4%. Sin embargo, para el panel siguiente después de haberse aplicado la primera corrección la deriva fue nula, lo que unido al efecto de la primera corrección provocó que el panel se procesara más rápido de lo esperado y *Reco* tuviera que volver a actuar, reduciendo v_u a 50.4%. Luego, tras haberse procesado 14 paneles después de aplicada la segunda corrección *Reco* volvió a actuar por tercera vez incrementando el valor de v_u a 54.3%. Nuevamente, debido a la intermitencia de la desviación, después de haberse procesado 5 paneles *Reco* aplicó una cuarta corrección disminuyendo el valor de v_u a 50.8%. Por último, tras procesarse 10 paneles después de la cuarta corrección, cuando la deriva se aplicó permanentemente, se incrementó v_u a 54.2%, logrando corregirse la desviación en la productividad de este robot.

Finalmente, en la cuarta prueba se introdujeron dos perturbaciones instantáneas y se ensayó la ocurrencia de varias derivas simultáneas. La primera perturbación instantánea se le aplicó al robot Stäubli RX90 restándole 1.75 s al tiempo medido para el panel número 86, lo que representa una disminución del 6% del tiempo promedio para este elemento, mientras que la segunda perturbación se le aplicó al tiempo de procesamiento de las cintas transportadoras sumándole 0.79 s al valor medido para el panel número 156, lo que representa un 5% más del valor de tiempo promedio para este elemento al inicio de la prueba. Al igual que en las pruebas anteriores, dada la magnitud y brevedad de estas perturbaciones *Reco* no necesitó actuar y los equipos continuaron operando con normalidad tras haber rebasado el efecto de la perturbación. Estos resultados pueden ser apreciados en la Figura 6.31.

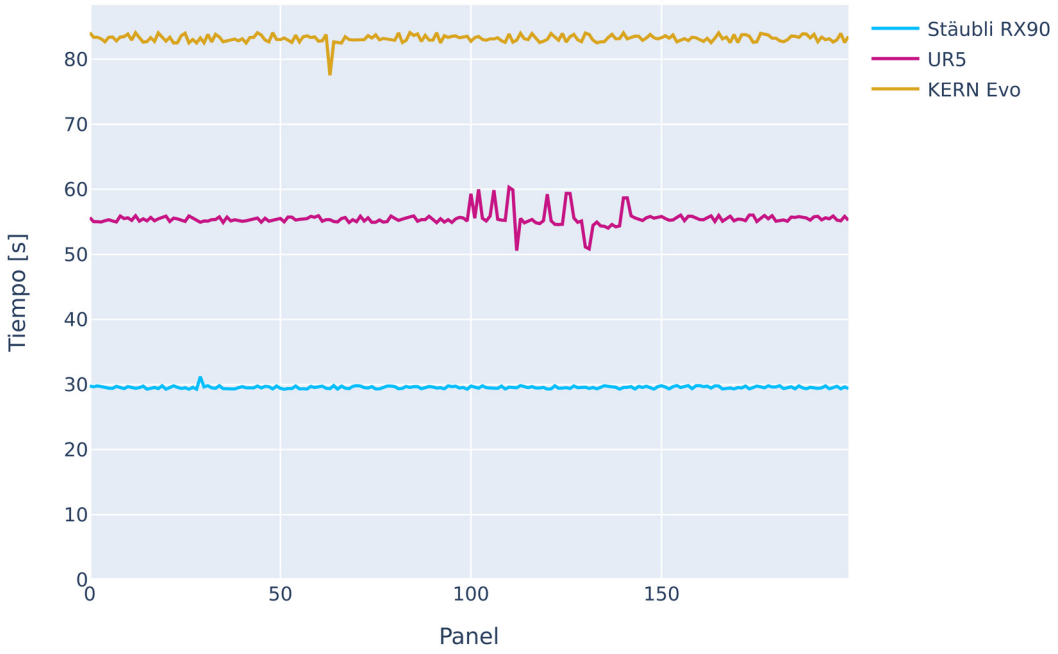

Figura 6.30: Comportamiento frente a perturbaciones instantáneas en el robot Stäubli RX90 y la fresadora CNC KERN Evo, y una deriva de concepto gradual en el robot UR5e

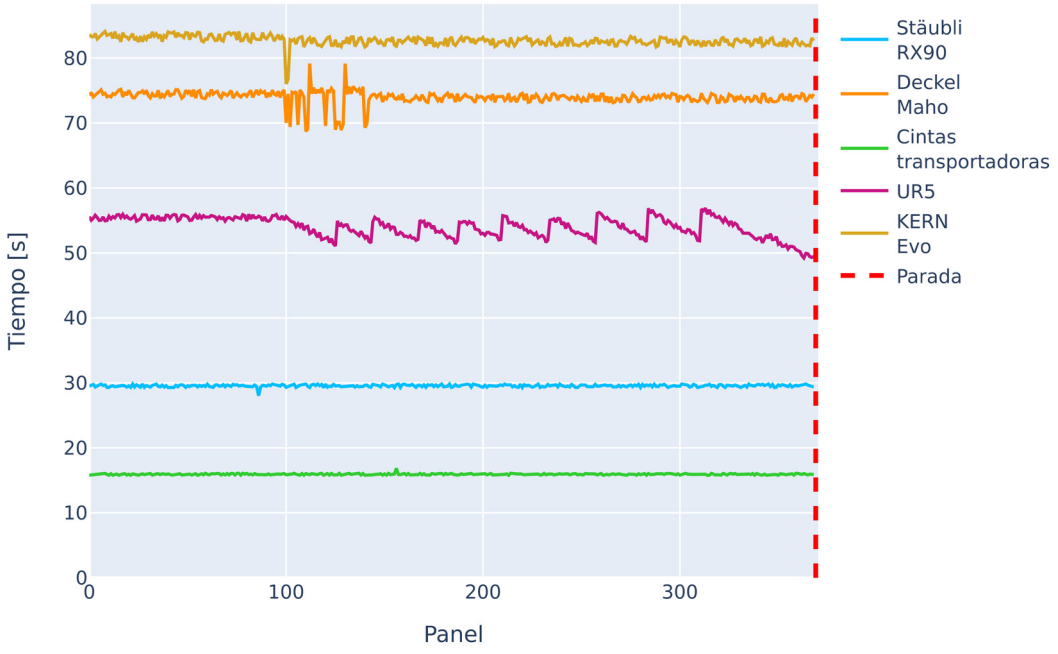

Figura 6.31: Comportamiento frente a perturbaciones instantáneas en el robot Stäubli RX90 y las cintas transportadoras y derivas de concepto repentina en la fresadora CNC KERN Evo, gradual en el centro de mecanizado Deckel Maho e incremental en el robot UR5e

Por otra parte, se inyectaron tres derivas de concepto simultáneamente a partir del panel número 100. La primera fue una deriva de concepto gradual, que se le inyectó al centro de mecanizado restándole 5.16 s a las mediciones del tiempo de procesamiento de este robot de manera intermitente, lo que representa una disminución del 7% con respecto a su tiempo promedio. A partir del panel número 140 se continuó restando esta cantidad al tiempo de procesamiento medido, pero de manera permanentemente. Ante esta deriva, *Reco* actuó en varias ocasiones. La primera fue tras haberse procesado 11 paneles después del inicio de la deriva, donde se redujo a_d a 42111 mm/min. Sin embargo, para el panel siguiente después de haberse aplicado la primera corrección la deriva fue nula, lo que unido al efecto de la primera corrección provocó que el panel se procesara más lento de lo esperado y *Reco* tuviera que volver a actuar, aumentando a_d a 44829 mm/min. Luego, tras haberse procesado 17 paneles después de aplicada la segunda corrección *Reco* volvió a actuar por tercera vez reduciendo el valor de a_d a 44282 mm/min. Nuevamente, debido a la intermitencia de la desviación, tras procesarse el siguiente panel *Reco* aplicó una cuarta corrección aumentando el valor de a_d a 45000 mm/min. Por último, tras procesarse 11 paneles después de la cuarta corrección, cuando la deriva se aplicó permanentemente, se redujo a_d a 42579 mm/min, logrando corregirse la desviación en la productividad del centro de mecanizado.

La segunda deriva de concepto inyectada fue repentina, y se le aplicó a la fresadora CNC KERN Evo. En este caso la deriva consistió en restarle 6.6 s al tiempo medido para este elemento, lo que representa una disminución del 8% de su valor de tiempo promedio al iniciar la prueba. *Reco* actuó tras procesar los datos del segundo panel después de inyectada la deriva, logrando que el tiempo de procesamiento de este equipo regresara a los valores previos a la desviación. Esto se logró reduciendo el valor de a_k a 7529 mm/min.

Mientras tanto, la tercera deriva de concepto inyectada en esta prueba fue de tipo incremental, lo cual se llevó a cabo restándole al tiempo de procesamiento del robot UR5e 0.16 s de manera acumulativa por cada panel procesado después del panel número 100. Al igual que en la anterior prueba donde se ensayó este tipo de deriva, *Reco* intentó corregir la desviación en varias ocasiones, hasta que, en cierto punto decidió realizar una parada ya que el UR5e tenía adelantos considerables en la manipulación de los paneles. Las correcciones fueron las siguientes: la primera corrección ocurrió tras haberse procesado 25 paneles después del inicio de la deriva y se redujo v_u a 47.0%; la segunda ocurrió tras haberse procesado 18 paneles

después de la primera corrección y se redujo v_u a 44.4%; la tercera corrección se produjo tras haberse procesado 24 paneles después de la segunda corrección y se redujo v_u a 42.1%; la cuarta corrección ocurrió tras haberse procesado 20 paneles después de la tercera corrección y se redujo v_u a 39.9%; la quinta corrección se produjo tras haberse procesado 22 paneles después de la cuarta corrección y se redujo v_u a 37.8%; la sexta corrección sucedió tras haberse procesado 24 paneles después de la quinta corrección y se redujo v_u a 35.9%; la séptima corrección tuvo lugar tras haberse procesado 24 paneles después de la sexta corrección y se redujo v_u a 33.9%; la octava corrección ocurrió tras haberse procesado 26 paneles después de la séptima corrección y se redujo v_u a 32.1%; la novena corrección tuvo lugar tras haberse procesado 27 paneles después de la octava corrección y se redujo v_u a 30.4% y la décima corrección sucedió tras haberse procesado 33 paneles después de la novena corrección y se redujo v_u a 30.0 %. A partir de este punto la desviación continuó aumentando a pesar de que se había establecido el valor de v_u en el límite inferior del rango definido, hasta que después de haberse procesado 27 paneles en estas condiciones la salida de *Reco* indicó que debía realizarse una parada ya que a pesar de las correcciones el robot UR5e seguía manipulando los paneles significativamente más rápido de lo esperado.

6.7. Conclusiones

En este capítulo se describió la línea piloto de Industria 4.0 del grupo GAMHE, la cual se compone de 6 tipos de elementos productivos heterogéneos y ha sido preparada para simular una parte de un proceso de fabricación de paneles termoaislantes. Además, se detallaron las condiciones de operación establecidas para realizar las diferentes pruebas llevadas a cabo.

La primera prueba consistió en la evaluación del módulo *Insight Generator*, el cual se utilizó para intentar obtener parametrizaciones óptimas para la línea piloto a partir de un conjunto de datos históricos de esta que contenía información sobre varias variables del proceso productivo y los dos KPIs que se querían optimizar, orientados a reducir el tiempo de producción mientras que se intenta maximizar el período de inactividad del operador. En esta prueba se analizaron los resultados de cada uno de los pasos descritos en la metodología de aprendizaje automático automatizado. La estandarización permitió transformar a rangos similares los valores de las variables de entrada, para de esta forma intentar evitar que ocurra algún sesgo al procesar los datos debido a la diferencia entre las escalas. En la etapa de selección de variables para ambos KPIs se seleccionaron cuatro variables

de las siete presentes en el conjunto de datos, coincidiendo en su mayoría con las que se podían señalar de manera intuitiva como las más relevantes para predecir los KPIs. Hubo una variable de las excluidas, la cual se relaciona directamente con el tiempo de operación de las cintas transportadoras; sin embargo, en la prueba no se seleccionó ya que se determinó que no tenía una influencia relevante en los KPIs del proceso productivo en general. Seguidamente, se crearon los modelos para cada KPI y se entrenaron. Se ejecutó la optimización de hiperparámetros, determinándose para cada modelo cuales eran los que tenían mayor importancia y como influenciaban el proceso de optimización. Posteriormente, se evaluaron las dos estrategias de aprendizaje de conjunto propuestas. Luego, se compararon los resultados de todos los modelos para seleccionar el de mejor desempeño para cada KPI. Para ambos KPIs se seleccionó la estrategia de metamodelado. En el caso de T_p esta estrategia logró un valor de $R^2 = 0.982$ al evaluar el conjunto de prueba, mostrando una alta capacidad de generalización. Mientras que para T_i también se obtuvo un resultado destacable, alcanzándose un valor de $R^2 = 0.976$. La importancia de estos resultados radica en que se obtuvieron en base a una metodología genérica y no en una hecha a medida para la línea piloto. La siguiente etapa consistió en obtener parametrizaciones óptimas para la línea piloto utilizando los modelos seleccionados, mediante la ejecución de un proceso de optimización multiobjetivo basado en un algoritmo genético. Este proceso de optimización se ejecutó varias veces, especificando diferentes restricciones, obteniéndose conjuntos de Pareto adaptados a estas. Al evaluar valores de los conjuntos de Pareto en la línea piloto se pudo comprobar la similitud entre los resultados obtenidos para los KPIs y las fronteras de Pareto predichas.

En este capítulo también se mostró la recreación del escenario de producción en la interfaz de visualización del gemelo digital y se realizaron pruebas para evaluar el comportamiento del componente de supervisión. En dichas pruebas se ensayaron diferentes tipos de derivas de concepto. Primeramente, se analizó la respuesta ante una deriva repentina, la cual fue detectada tras haberse procesado 13 paneles. En el segundo caso, se ensayó una deriva incremental, requiriendo que se procesaran 23 paneles para detectarla. Por último, se analizó una deriva gradual, la cual se detectó tras haberse procesado 51 paneles. Dada la naturaleza diversa de estas derivas, el componente de supervisión se comportó de forma adecuada, ya que entre sus propósitos se incluía detectar la deriva de manera temprana pero no inmediata, para evitar que fuese susceptible a perturbaciones momentáneas que pueden ocurrir en el proceso productivo.

Por otra parte, también se llevaron a cabo pruebas para evaluar la capacidad de corregir desviaciones en la productividad de los equipos de la línea piloto utilizando el módulo *Reco*. Específicamente, se evaluaron cuatro situaciones diferentes en las que se combinaron perturbaciones instantáneas con derivas de concepto. De esta forma, se pudo comprobar la robustez de la metodología frente a fluctuaciones momentáneas, mientras que, para el caso de las derivas sí generó respuestas que permitieron mitigar sus efectos. Las derivas repentinas simuladas, que consistieron en variaciones de hasta el 8% del tiempo de procesamiento esperado, fueron corregidas rápidamente sin contratiempos. Los efectos de las derivas graduales se corrigieron en varias ocasiones hasta que se alcanzó el punto de estabilidad. Finalmente, en el caso de las derivas incrementales, sus efectos también se corrigieron en varias ocasiones; sin embargo, como la magnitud de estas derivas continuó aumentando se llegó a un límite donde las acciones derivadas de las respuestas del módulo no eran capaces de mitigarlas. Una de las implicaciones más importantes de estas pruebas es que se logró cumplir con los objetivos aun cuando los equipos sometidos a las derivas presentaban un alto grado de homogeneidad. Estos resultados fueron posibles debido al diseño genérico de la metodología que le permitió adaptarse a las características de cada uno de los casos analizados.

7. Conclusiones

La creciente necesidad de soluciones inspiradas en el paradigma de la Industria 4.0 que faciliten la democratización de la IA en el entorno industrial requiere el desarrollo de herramientas y plataformas que permitan crear aplicaciones basadas en IA de manera más simple y asequible a los diferentes actores, incluso aquellos que no posean *know-how* en estos temas. En este sentido la tesis se enfocó en el desarrollo de dos metodologías para ser implementadas como módulos en la plataforma KITT4SME con los objetivos de determinar parametrizaciones óptimas de procesos productivos a partir de datos históricos y de corregir desviaciones en la productividad de los equipos de fabricación. Estas metodologías fueron diseñadas de manera tal que fuesen independientes de la aplicación, de manera tal que pudieran adaptarse a un amplio rango de situaciones.

Estos módulos fueron utilizados en aplicaciones desarrolladas para la línea piloto de GAMHE, lo cual permitió llevar a cabo numerosas pruebas en un escenario similar a un entorno de fabricación real, obteniéndose resultados muy positivos. El primer módulo, *Insight Generator*, fue utilizado como capa de inteligencia de un gemelo digital, creando modelos capaces de representar ciertos KPIs de la línea piloto y determinar parametrizaciones óptimas para estos. Al formar parte de un gemelo digital, el cual debe coevolucionar con el proceso productivo se pudo determinar cuándo se requería reentrenar los modelos frente a la ocurrencia de situaciones complejas como derivas de concepto de diferente naturaleza. El segundo módulo, *Reco*, fue utilizado para asegurar la estabilidad de la producción en la línea piloto al mitigar las desviaciones de cada elemento de esta, pero, además, permitió detener el proceso ante situaciones que no fue capaz de resolver. Estas pruebas permitieron comprobar la fiabilidad y robustez de los módulos desarrollados.

7.1. Contribuciones

Esta tesis incluye contribuciones significativas al estado del arte en el campo de la supervisión y reconfiguración automática e inteligente de sistemas ciberfísicos, con una materialización pragmática mediante la implementación de módulos en la arquitectura y plataforma KITT4SME y su aplicación a la línea piloto de GAMHE,

como vía para corroborar y evaluar las metodologías propuestas. A continuación, se resumen las aportaciones:

- Un método para la generación de recomendaciones inteligentes mediante la reparametrización óptima de los procesos productivos. Se desarrolló una metodología de aprendizaje automático automatizado capaz de generar parametrizaciones óptimas de los procesos productivos. La principal particularidad de esta metodología es que permite tratar conjuntamente todo el proceso desde que se reciben los datos hasta que se generan las parametrizaciones, simplificando la creación de soluciones basadas en IA. Esta metodología ha sido estructurada internamente en varias etapas, en las que se ha tenido en cuenta la utilización de métodos fiables y efectivos para cada tarea. La primera etapa se encarga del preprocesamiento de los datos, eliminando posibles inconsistencias y garantizando la integridad de la información. Posteriormente, se lleva a cabo una etapa de análisis de variables, donde se seleccionan las que poseen mayor capacidad predictiva utilizando un criterio basado en los coeficientes de correlación de Pearson y Spearman. La etapa siguiente se enfoca en la creación de los modelos para representar los KPIs deseados del proceso productivo analizado, así como en la optimización de sus hiperparámetros. En este sentido, se consideraron cuatro tipos de modelos diferentes: MLP, SVR, XGBoost y *k*NN, garantizando la diversidad de enfoques utilizados. La optimización de hiperparámetros se lleva a cabo mediante TPE. Además de los modelos base mencionados, se exploran dos estrategias de aprendizaje de conjunto. Una vez que se tienen todos los modelos base y las dos estrategias de aprendizaje de conjunto estos son evaluados, se realiza una comparación y se selecciona la opción que mayor valor de R^2 haya obtenido. En la última etapa, se utiliza el algoritmo NSGA-II para resolver un problema de optimización multiobjetivo, en la que las funciones objetivo son los modelos seleccionados para cada KPI, y como resultado se obtienen un conjunto de parametrizaciones óptimas para el proceso productivo.
- Diseño e implementación de un gemelo digital de la línea piloto de GAMHE. Se diseñó y desarrolló un gemelo digital de la línea piloto de GAMHE, el cual

cuenta con una capa de inteligencia para crear los modelos que representan al proceso productivo, realizar predicciones y optimizar su funcionamiento. El gemelo digital también cuenta con una interfaz de visualización que permite apreciar de manera gráfica el comportamiento esperado de la línea. Igualmente, cuenta con una capacidad de supervisión del desempeño de los modelos que permite detectar si estos continúan siendo adecuados para representar el proceso productivo o si, por el contrario, se necesita actualizarlos ya que el proceso ha sufrido cambios.

- Un método para la reconfiguración automática e inteligente de un sistema ciberfísico para la corrección de desviaciones en la productividad de los equipos de fabricación. Se desarrolló una metodología basada en un sistema de inferencia borroso tipo Mamdani, capaz de generar respuestas que pueden ser utilizadas para mitigar las fluctuaciones en la productividad de los equipos de fabricación que pueden ser provocadas por diversas razones, como degradación de los equipos o acciones externas, entre otras. La metodología se diseñó de manera que pudiera ser aplicada a diferentes tipos de equipos, quedando en manos del usuario la interpretación de la respuesta y su adaptación a la situación particular. De esta forma, se pueden desarrollar sistemas de toma de decisión automatizados basándose en los resultados de la metodología o sistemas de recomendaciones inteligentes, donde se le brinda información al operador y es este quien debe decidir cómo proceder.
- Integración y validación: Las dos metodologías presentadas en esta tesis fueron implementadas e integradas en la plataforma KITT4SME como los módulos *Insight Generator* y *Reco*. Se trata de una plataforma de referencia en Europa para la introducción de métodos de IA en los sistemas manufactureros de las empresas. Por otra parte, el gemelo digital de la línea piloto utilizó el módulo *Insight Generator* como capa de inteligencia. Para comprobar la validez de todos los elementos desarrollados se llevaron a cabo extensas pruebas en la línea piloto de GAMHE, la cual recrea un escenario de producción real.

7.2. Desafíos

Las contribuciones realizadas en esta tesis y los resultados positivos alcanzados han puesto de manifiesto el largo camino en el diseño e implementación de soluciones basadas en IA y gemelos digitales y su aplicación efectiva y eficaz en la industria manufacturera. A continuación, se exponen desafíos que abordar a corto y medio plazo en el ámbito de este trabajo:

- Las dos metodologías presentadas, al ser genéricas, pueden ser generalizables a un amplio rango de procesos productivos, simplificando su utilización de manera tal que incluso personal con poca experiencia en el área de la IA puedan utilizarlas. Si bien los resultados obtenidos con estas son muy prometedores se debe tener en cuenta que, si se desarrolla una solución a medida para el proceso productivo analizado, se pueden obtener mejores resultados. En este sentido en la tesis se buscó un compromiso entre, por un lado, simplificar la utilización de los módulos y reducir drásticamente los tiempos de desarrollo para los potenciales usuarios de dichos módulos, mientras que, por el otro, existe la posibilidad, desde el punto de vista científico, que los resultados obtenidos sean mejorables si se desarrolla una solución a medida para la aplicación.
- La metodología de aprendizaje automático automatizado propuesta no incluye una etapa de clasificación automática lo que abre las puertas a nuevos diseños que tengan en cuenta algoritmos más avanzados.
- La metodología de aprendizaje automático automatizado ha sido diseñada para procesar datos estructurados. Si bien otras fuentes de datos, como los audios, imágenes o vídeos, pueden aportar información de gran valor para predecir los valores de los KPIs, por el momento no es posible utilizarlos.

7.3. Futuras líneas de investigación y desarrollo

Las contribuciones y resultados alcanzados en esta tesis alientan la continuación del trabajo en varias direcciones. A continuación, se listan las futuras líneas de investigación que se pretenden explorar:

- Incluir modelos de clasificación en el módulo *Insight Generator*, de manera que para los KPIs con valores continuos se utilicen algoritmos de regresión y para los KPIs con valores discretos se utilicen los de clasificación. Esto permitiría contar con un enfoque más adecuado para KPIs como la aceptación de los productos por parte del cliente, los cuales generalmente son categóricos y se miden mediante encuestas.
- Analizar, diseñar y aplicar algoritmos de optimización multiobjetivo para el módulo *Insight Generator*. A pesar de que NSGA-II es un algoritmo de probada eficacia, existen otras técnicas que podrían aportar valor añadido al módulo. Por ejemplo, el algoritmo *Reference Point-based Non-dominated Sorting Genetic Algorithm II* (RNSGA-II) busca el conjunto de Pareto óptimo, pero enfocándose en las regiones cercanas a puntos de referencia que se provean. Esto es útil cuando el usuario conoce de antemano la dinámica del proceso productivo y tiene definido claramente sus objetivos, obteniéndose un conjunto de Pareto adaptado a sus especificaciones, en lugar de un conjunto que priorice la amplitud del frente.
- Analizar, diseñar y aplicar técnicas de meta aprendizaje para acelerar la etapa de creación de los modelos y optimización de hiperparámetros. Entre las opciones a valorar en este caso se encuentran técnicas como la transferencia de aprendizaje, en la que se toma un modelo entrenado previamente y se reentrena para la nueva aplicación, o la exploración de modelos candidatos basado en heurísticas, en la cual, no se evalúan siempre todos los modelos, sino que en dependencia de las características del problema se seleccionan los modelos que se estime serán más convenientes.
- Analizar, diseñar y aplicar la sensorización del operador para trabajar con KPIs más sofisticados. Aunque el tiempo de inactividad durante la producción influye en el nivel de fatiga del operador, esta variable no refleja ciertos factores objetivos y subjetivos que pueden variar de un individuo a otro. Mediante el uso de dispositivos *wearables* se podría tener información más detallada del operador lo que permitiría llevar a cabo el proceso de optimización de la producción mejor adaptado a cada caso en concreto.

7.4. Promoción y difusión de los resultados

Durante el desarrollo de esta tesis, el candidato a doctor participó en varios proyectos europeos y nacionales, lo que permitió la colaboración con varios centros de investigación, universidades y empresas de gran prestigio en el ámbito de la Industria 4.0. Como resultado del trabajo realizado tanto en el marco de la tesis como en el de los proyectos en los que el candidato participó, se llevaron a cabo varias actividades de divulgación las cuales se presentan a continuación:

Artículos de revista

1. **Cruz, Y. J.**, Rivas, M., Quiza, R., Beruvides, G., Haber, R. E. (2020). Computer Vision System for Welding Inspection of Liquefied Petroleum Gas Pressure Vessels Based on Combined Digital Image Processing and Deep Learning Techniques. *Sensors* 20(16), 4505. https://doi.org/10.3390/s20164505
2. **Cruz, Y. J.**, Rivas, M., Quiza, R., Villalonga, A., Haber, R. E., Beruvides, G. (2021). Ensemble of convolutional neural networks based on an evolutionary algorithm applied to an industrial welding process. *Computers in Industry*, *133*, 103530. https://doi.org/10.1016/j.compind.2021.103530
3. **Cruz, Y. J.**, Rivas, M., Quiza, R., Haber, R. E., Castaño, F., Villalonga, A. (2022). A two-step machine learning approach for dynamic model selection: A case study on a micro milling process. *Computers in Industry*, *143*, 103764. https://doi.org/10.1016/j.compind.2022.103764
4. Castano, F., **Cruz, Y. J.**, Villalonga, A., Haber, R. E. (2022). Data-driven insights on time-to-failure of electromechanical manufacturing devices: a procedure and case study. *IEEE Transactions on Industrial Informatics*, 1–11. https://doi.org/10.1109/TII.2022.3216629
5. **Cruz, Y. J.**, Villalonga, A., Castaño, F., Rivas, M., Haber, R. E. (2023). Automated Machine Learning for Optimizing Production Processes in Small and Medium-Sized Enterprises. *Operations Research Perspectives* (en revisión), *Social Science Research Network* (preimpreso accesible). https://doi.org/10.2139/ssrn.4334026

Patentes y propiedades industriales

1. Haber, R. E., **Cruz, Y. J.**, Villalonga, A., Castaño, F. (Registro: 05/12/2022). *Método para Optimización de Rendimiento de una Planta de Fabricación*

mediante Generación de Algoritmos de Aprendizaje Automático completamente automatizados. Entidad titular de derechos: Consejo Superior de Investigaciones Científicas. N.º de solicitud: P202231046.

2. Haber, R. E., Castaño, F., **Cruz, Y. J.**, Villalonga, A. (Registro: 01/09/2022, Concesión: 14/09/2022). *Insight Generator: Software para generación de información útil y recomendaciones inteligentes sobre los procesos productivos*. País de inscripción: España. Entidad titular de derechos: Consejo Superior de Investigaciones Científicas. N.º de solicitud: 73acdaf39a04f96ecdd2f5c6ca4870d6b7203ead28fae55e0e687339c8492f84
3. Haber, R. E., Castaño, F., Villalonga, A., **Cruz, Y. J.** (Registro: 27/01/2023). *Method and system for self-reconfiguration of production processes*, Trade Secret registred by SiGNE_Block

Capítulos de libro

1. **Cruz, Y. J.**, Castaño, F., Haber, R. E., Villalonga, A., Ejsmont, K., Gladysz, B., Flores, Á., Alemany, P. (2024). Self-Reconfiguration for Smart Manufacturing Based on Artificial Intelligence: A Review and Case Study. En J. Soldatos (ed.), *Artificial Intelligence in Manufacturing*. Springer Nature. https://doi.org/10.1007/978-3-031-46452-2_8

Ponencias en eventos científicos

1. **Cruz, Y. J.**, Castaño, F., Villalonga, A., Haber, R. E. (21-23 de marzo, 2023). *Optimización de procesos productivos basada en gemelos digitales*. XI Convención Científica Internacional "La Universidad en la Agenda 2030", Matanzas, Cuba.

Prensa y medios de comunicación

1. (22 de septiembre, 2022). Un estudio del CSIC aplica el aprendizaje automático en la selección de modelos que explican procesos cambiantes. https://www.csic.es/es/actualidad-del-csic/un-estudio-del-csic-aplica-el-aprendizaje-automatico-en-la-seleccion-de-modelos

Premios

1. Quiza, R., Haber, R. E., Rivas, M., Beruvides, G., Villalonga, A., La Fé, I., **Cruz, Y. J.**, Pérez, R., del Risco, R. (2022). *Herramientas inteligentes para el acercamiento de los procesos de fabricación mecánica al paradigma de Industria 4.0*. Premio Nacional Anual de la Academia de Ciencias de Cuba.

Atribuciones y Reconocimientos

La Figura 1.1 ha sido inspirada en (Muhuri et al., 2019).

La Figura 1.2 ha sido inspirada en (Butt, 2020).

La Figura 1.3 ha sido inspirada en (Confederación Española de la Pequeña y Mediana Empresa, 2023).

La Figura 2.1 ha sido inspirada en (Tao et al., 2019).

La Figura 2.2 ha sido inspirada en (Kirmse et al., 2019).

La Figura 2.3 ha sido inspirada en (Y. Lu et al., 2015).

La Figura 2.4 ha sido inspirada en (Wei et al., 2017).

La Figura 2.5 ha sido inspirada en (Villalonga et al., 2020).

La Figura 2.6 ha sido inspirada en (Singh et al., 2021).

La Figura 2.7 ha sido inspirada en (Barbieri et al., 2021).

La Figura 2.8 ha sido inspirada en (Bettoni et al., 2021).

La Figura 2.10, la Figura 2.11 y la Figura 2.12 han sido inspiradas en (Google, 2023).

La Figura 2.13 ha sido inspirada en (Friederich et al., 2022).

La Figura 2.14 ha sido inspirada en (Lv & Xie, 2022).

La Figura 3.1 corresponde a la vista conceptual presentada en el proyecto KITT4SME (https://github.com/c0c0n3/kitt4sme/blob/master/arch/conceptual-view/components.png).

La Figura 3.2 ha sido inspirada en (Qi et al., 2021).

La Figura 4.14 ha sido inspirada en (J. Lu et al., 2019).

La Figura 4.15 ha sido inspirada en (Grulich et al., 2018).

En esta tesis se han utilizado imágenes de acceso abierto con reconocimiento de autor, a saber: anilofex, Assia Benkerroum, Design Circle, Eucalyp, fajarestuu, Flat Icons, Freepik, Futuer, iconixar, itim2101, juicy_fish, Leremy, mavadee, lutfix, Nikita Golubev, noomtah, Parzival' 1997, Paul J., photo3idea_studio,

smalllikeart, Smashicons, smashingstocks, Uniconlabs, vectorsmarket15, Vectors Tank, wanicon, xnimrodx, tomadas de flaticom.com.

Referencias

Acernese, A., Del Vecchio, C., Tipaldi, M., Battilani, N., & Glielmo, L. (2021). Condition-based maintenance: an industrial application on rotary machines. *Journal of Quality in Maintenance Engineering*, *27*(4), 565–585. https://doi.org/10.1108/JQME-10-2019-0101

Adam, S. P., Alexandropoulos, S.-A. N., Pardalos, P. M., & Vrahatis, M. N. (2019). *No Free Lunch Theorem: A Review BT - Approximation and Optimization : Algorithms, Complexity and Applications* (I. C. Demetriou & P. M. Pardalos (eds.); pp. 57–82). Springer International Publishing. https://doi.org/10.1007/978-3-030-12767-1_5

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. *CoRR*, *abs/1907.1*. http://arxiv.org/abs/1907.10902

Al-Tashi, Q., Abdulkadir, S. J., Rais, H. M., Mirjalili, S., & Alhussian, H. (2020). Approaches to Multi-Objective Feature Selection: A Systematic Literature Review. *IEEE Access*, *8*, 125076–125096. https://doi.org/10.1109/ACCESS.2020.3007291

Albashish, D., Hammouri, A. I., Braik, M., Atwan, J., & Sahran, S. (2021). Binary biogeography-based optimization based SVM-RFE for feature selection. *Applied Soft Computing*, *101*, 107026. https://doi.org/10.1016/j.asoc.2020.107026

Alcácer, V., & Cruz-Machado, V. (2019). Scanning the Industry 4.0: A Literature Review on Technologies for Manufacturing Systems. *Engineering Science and Technology, an International Journal*, *22*(3), 899–919. https://doi.org/10.1016/j.jestch.2019.01.006

Alexopoulos, K., Nikolakis, N., & Chryssolouris, G. (2020). Digital twin-driven supervised machine learning for the development of artificial intelligence applications in manufacturing. *International Journal of Computer Integrated Manufacturing*, *33*(5), 429–439. https://doi.org/10.1080/0951192X.2020.1747642

Aoun, A., Ilinca, A., Ghandour, M., & Ibrahim, H. (2021). A review of Industry 4.0 characteristics and challenges, with potential improvements using blockchain technology. *Computers & Industrial Engineering*, *162*, 107746. https://doi.org/10.1016/j.cie.2021.107746

Bahassine, S., Madani, A., Al-Sarem, M., & Kissi, M. (2020). Feature selection using an improved Chi-square for Arabic text classification. *Journal of King Saud University - Computer and Information Sciences*, *32*(2), 225–231. https://doi.org/10.1016/j.jksuci.2018.05.010

Bao, J., Guo, D., Li, J., & Zhang, J. (2019). The modelling and operations for the digital twin in the context of manufacturing. *Enterprise Information Systems*, *13*(4), 534–556. https://doi.org/10.1080/17517575.2018.1526324

Baptista, L. F., & Barata, J. (2021). Piloting Industry 4.0 in SMEs with RAMI 4.0: an enterprise architecture approach. *Procedia Computer Science*, *192*, 2826–2835. https://doi.org/10.1016/j.procs.2021.09.053

Barbieri, G., Bertuzzi, A., Capriotti, A., Ragazzini, L., Gutierrez, D., Negri, E., & Fumagalli, L. (2021). A virtual commissioning based methodology to integrate digital twins into manufacturing systems. *Production Engineering*, *15*(3), 397–412. https://doi.org/10.1007/s11740-021-01037-3

Beruvides, G., Castaño, F., Haber Guerra, R. E., Quiza, R., & Villalonga, A. (2017). Coping with Complexity When Predicting Surface Roughness in Milling Processes: Hybrid Incremental Model with Optimal Parametrization. *Complexity*, *2017*, 1–11. https://doi.org/10.1155/2017/7317254

Bettoni, A., Matteri, D., Montini, E., Gładysz, B., & Carpanzano, E. (2021). An AI adoption model for SMEs: a conceptual framework. *IFAC-PapersOnLine*, *54*(1), 702–708. https://doi.org/10.1016/j.ifacol.2021.08.082

Bifet, A., & Gavaldà, R. (2007). Learning from Time-Changing Data with Adaptive Windowing. In *Proceedings of the 2007 SIAM International Conference on Data Mining (SDM)* (pp. 443–448). Society for Industrial and Applied Mathematics. https://doi.org/doi:10.1137/1.9781611972771.42

Blank, J., & Deb, K. (2020). Pymoo: Multi-Objective Optimization in Python. *IEEE Access*, *8*, 89497–89509. https://doi.org/10.1109/ACCESS.2020.2990567

Bogoviz, A. V, Osipov, V. S., Chistyakova, M. K., & Borisov, M. Y. (2019). *Comparative Analysis of Formation of Industry 4.0 in Developed and Developing Countries BT - Industry 4.0: Industrial Revolution of the 21st Century* (E. G. Popkova, Y. V Ragulina, & A. V Bogoviz (eds.); pp. 155–164). Springer International Publishing. https://doi.org/10.1007/978-3-319-94310-7_15

Bretz, L., Klinkner, F., Kandler, M., Shun, Y., & Lanza, G. (2022). The ECO Maturity Model – A human-centered Industry 4.0 maturity model. *Procedia CIRP*, *106*, 90–95. https://doi.org/10.1016/j.procir.2022.02.160

Butt, J. (2020). A Strategic Roadmap for the Manufacturing Industry to Implement Industry 4.0. *Designs*, *4*(2). https://doi.org/10.3390/designs4020011

Canizo, M., Conde, A., Charramendieta, S., Miñón, R., Cid-Fuentes, R. G., & Onieva, E. (2019). Implementation of a Large-Scale Platform for Cyber-Physical System Real-Time Monitoring. *IEEE Access*, *7*, 52455–52466. https://doi.org/10.1109/ACCESS.2019.2911979

Cannizzaro, D., Simone, F., Illgner-Fehns, K., Mata, S., Mondino, I., Ghiazza, A., Poncino, M., & Cataldo, S. Di. (2022). Quality inspection of critical aircraft engine components: towards full automation. *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1–4. https://doi.org/10.1109/ETFA52439.2022.9921542

Castano, F., Cruz, Y. J., Villalonga, A., & Haber, R. E. (2022). Data-driven insights on time-to-failure of electromechanical manufacturing devices: a procedure and case study. *IEEE Transactions on Industrial Informatics*, 1–11. https://doi.org/10.1109/TII.2022.3216629

Chauhan, C., Singh, A., & Luthra, S. (2021). Barriers to industry 4.0 adoption and its performance implications: An empirical investigation of emerging economy. *Journal of Cleaner Production*, *285*, 124809. https://doi.org/10.1016/j.jclepro.2020.124809

Chauhan, K., Jani, S., Thakkar, D., Dave, R., Bhatia, J., Tanwar, S., & Obaidat, M. S. (2020). Automated Machine Learning: The New Wave of Machine Learning. *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 205–212. https://doi.org/10.1109/ICIMIA48430.2020.9074859

Chen, G., Wang, P., Feng, B., Li, Y., & Liu, D. (2020). The framework design of smart factory in discrete manufacturing industry based on cyber-physical system. *International Journal of Computer Integrated Manufacturing*, *33*(1), 79–101. https://doi.org/10.1080/0951192X.2019.1699254

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ. Computer Science*, *7*, e623. https://doi.org/10.7717/peerj-cs.623

Confederación Española de la Pequeña y Mediana Empresa. (2023). *Crecimiento Empresarial. Situación de las pymes en España comparada con la de otros países europeos*. https://cepyme.es/wp-content/uploads/2023/11/INFORME-CRECIMIENTO-EMPRESARIAL_2023-def.pdf

Conti, C. J., Varde, A. S., & Wang, W. (2022). Human-Robot Collaboration With Commonsense Reasoning in Smart Manufacturing Contexts. *IEEE Transactions on Automation Science and Engineering*, *19*(3), 1784–1797. https://doi.org/10.1109/TASE.2022.3159595

Cruz, Y. J., Rivas, M., Quiza, R., Beruvides, G., & Haber, R. E. (2020). Computer Vision System for Welding Inspection of Liquefied Petroleum Gas Pressure Vessels Based on Combined Digital Image Processing and Deep Learning Techniques. *Sensors*, *20*(16). https://doi.org/10.3390/s20164505

Cruz, Y. J., Rivas, M., Quiza, R., Haber, R. E., Castaño, F., & Villalonga, A. (2022). A two-step machine learning approach for dynamic model selection: A case study on a micro milling process. *Computers in Industry*, *143*, 103764. https://doi.org/10.1016/j.compind.2022.103764

Cruz, Y. J., Rivas, M., Quiza, R., Villalonga, A., Haber, R. E., & Beruvides, G. (2021). Ensemble of convolutional neural networks based on an evolutionary

algorithm applied to an industrial welding process. *Computers in Industry*, *133*, 103530. https://doi.org/10.1016/j.compind.2021.103530

Cruz, Y. J., Villalonga, A., Castaño, F., Rivas, M., & Haber, R. E. (2023). Automated Machine Learning for Optimizing Production Processes in Small and Medium-Sized Enterprises. In *Social Science Research Network*. https://doi.org/10.2139/ssrn.4334026

Culot, G., Nassimbeni, G., Orzes, G., & Sartor, M. (2020). Behind the definition of Industry 4.0: Analysis and open questions. *International Journal of Production Economics*, *226*, 107617. https://doi.org/10.1016/j.ijpe.2020.107617

De Blasi, S., Bahrami, M., Engels, E., & Gepperth, A. (2023). Safe contextual Bayesian optimization integrated in industrial control for self-learning machines. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-023-02087-3

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197. https://doi.org/10.1109/4235.996017

Ding, D., Han, Q.-L., Wang, Z., & Ge, X. (2019). A Survey on Model-Based Distributed Control and Filtering for Industrial Cyber-Physical Systems. *IEEE Transactions on Industrial Informatics*, *15*(5), 2483–2499. https://doi.org/10.1109/TII.2019.2905295

Dobrowolska, M., & Knop, L. (2020). Fit to Work in the Business Models of the Industry 4.0 Age. *Sustainability*, *12*(12). https://doi.org/10.3390/su12124854

Dogan, A., & Birant, D. (2021). Machine learning and data mining in manufacturing. *Expert Systems with Applications*, *166*, 114060. https://doi.org/10.1016/j.eswa.2020.114060

Durbin, M., Wonders, M. A., Flaska, M., & Lintereur, A. T. (2021). K-Nearest Neighbors regression for the discrimination of gamma rays and neutrons in organic scintillators. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, *987*, 164826. https://doi.org/10.1016/j.nima.2020.164826

Essien, A., & Giannetti, C. (2020). A Deep Learning Model for Smart Manufacturing Using Convolutional LSTM Neural Network Autoencoders. *IEEE Transactions on Industrial Informatics*, *16*(9), 6069–6078. https://doi.org/10.1109/TII.2020.2967556

Fahle, S., Prinz, C., & Kuhlenkötter, B. (2020). Systematic review on machine learning (ML) methods for manufacturing processes – Identifying artificial intelligence (AI) methods for field application. *Procedia CIRP*, *93*, 413–418. https://doi.org/10.1016/j.procir.2020.04.109

Ferrer, B. R., Mohammed, W. M., Lastra, J. L. M., Villalonga, A., Beruvides, G., Castaño, F., & Haber, R. E. (2018). Towards the Adoption of Cyber-Physical Systems of Systems Paradigm in Smart Manufacturing Environments. *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, 792–

799. https://doi.org/10.1109/INDIN.2018.8472061

Friederich, J., Francis, D. P., Lazarova-Molnar, S., & Mohamed, N. (2022). A framework for data-driven digital twins of smart manufacturing systems. *Computers in Industry*, *136*, 103586. https://doi.org/10.1016/j.compind.2021.103586

Gajdzik, B., Grabowska, S., & Saniuk, S. (2021). A Theoretical Framework for Industry 4.0 and Its Implementation with Selected Practical Schedules. *Energies*, *14*(4). https://doi.org/10.3390/en14040940

Galati, F., & Bigliardi, B. (2019). Industry 4.0: Emerging themes and future research avenues using a text mining approach. *Computers in Industry*, *109*, 100–113. https://doi.org/10.1016/j.compind.2019.04.018

Gao, X., Li, X., Zhao, B., Ji, W., Jing, X., & He, Y. (2019). Short-Term Electricity Load Forecasting Model Based on EMD-GRU with Feature Selection. *Energies*, *12*(6). https://doi.org/10.3390/en12061140

Garcia, M. A. R., Rojas, R., Gualtieri, L., Rauch, E., & Matt, D. (2019). A human-in-the-loop cyber-physical system for collaborative assembly in smart manufacturing. *Procedia CIRP*, *81*, 600–605. https://doi.org/https://doi.org/10.1016/j.procir.2019.03.162

Garg, S., Pundir, P., Rathee, G., Gupta, P. K., Garg, S., & Ahlawat, S. (2021). On Continuous Integration / Continuous Delivery for Automated Deployment of Machine Learning Models using MLOps. *2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, 25–28. https://doi.org/10.1109/AIKE52691.2021.00010

Ghobakhloo, M., & Iranmanesh, M. (2021). Digital transformation success under Industry 4.0: a strategic guideline for manufacturing SMEs. *Journal of Manufacturing Technology Management*, *32*(8), 1533–1556. https://doi.org/10.1108/JMTM-11-2020-0455

Gladysz, B., Matteri, D., Ejsmont, K., Corti, D., Bettoni, A., & Haber Guerra, R. (2023). Platform-based support for AI uptake by SMEs: guidelines to design service bundles. *Central European Management Journal*, *ahead-of-p*(ahead-of-print). https://doi.org/10.1108/CEMJ-08-2022-0096

Gong, L., Zou, B., & Kan, Z. (2019). Modeling and Optimization for Automobile Mixed Assembly Line in Industry 4.0. *Journal of Control Science and Engineering*, *2019*, 3105267. https://doi.org/10.1155/2019/3105267

Google. (2023). *MLOps: Continuous delivery and automation pipelines in machine learning*. https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

Grulich, P. M., Saitenmacher, R., Traub, J., Breß, S., Rabl, T., & Markl, V. (2018). Scalable Detection of Concept Drifts on Data Streams with Parallel Adaptive Windowing. *Proceedings of the 21st International Conference on Extending Database Technology*, 477–480. https://doi.org/https://dx.doi.org/10.5441/002/edbt.2018.51

Guerra, R. E. H., Cembrano, G., Villagrá, J., Jiménez, A., Godoy, J., Coll, A. B., Margarit-Taulé, J. M., Fonseca, L., Álvarez-Arenas, T. G., Camacho, J., Secanellas, S. A., Carmona, R., Berni, J. F., del Río Encuentra, R. A., Jiménez-Berni, J. A., & Capdevila, C. (2021). Cyber-Physical Systems and Internet of Things. In *White Paper 10: Digital and complex information (2021) , http://hdl.handle.net/10261/272588*.

Gürdür Broo, D., Boman, U., & Törngren, M. (2021). Cyber-physical systems research and education in 2030: Scenarios and strategies. *Journal of Industrial Information Integration*, *21*, 100192. https://doi.org/10.1016/j.jii.2020.100192

Haber, R. E., Alique, J. R., Alique, A., & Haber, R. (2005). Controlling a complex electromechanical process on the basis of a neurofuzzy approach. *Future Generation Computer Systems*, *21*(7), 1083–1095. https://doi.org/10.1016/j.future.2004.03.008

Haber, R. E., Liang, S. Y., Alique, J. R., & Haber, R. (2006). Fuzzy Control of Spindle Torque in High-Speed Milling Processes. *Journal of Manufacturing Science and Engineering*, *128*(4), 1014–1018. https://doi.org/10.1115/1.2194063

Haber, R. E., Quiza, R., Villalonga, A., Arenas, J., & Castaño, F. (2019). Digital Twin-Based Optimization for Ultraprecision Motion Systems With Backlash and Friction. *IEEE Access*, *7*, 93462–93472. https://doi.org/10.1109/ACCESS.2019.2928141

Haber, R. E., Schmitt-Braess, G., Haber, R., Alique, A., & Alique, J. R. (2003). Using circle criteria for verifying asymptotic stability in PI-like fuzzy control systems: application to the milling process. *IEE Proceedings - Control Theory and Applications*, *150*(6), 619-627(8). https://doi.org/10.1049/ip-cta:20030795

Hajer, B., Arwa, B., Lobna, H., & Khaled, G. (2020). Intention Mining Data preprocessing based on Multi-Agents System. *Procedia Computer Science*, *176*, 888–897. https://doi.org/10.1016/j.procs.2020.09.084

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

He, Y., Zhang, X., & Zhang, L. (2018). Variable selection for high dimensional Gaussian copula regression model: An adaptive hypothesis testing procedure. *Computational Statistics & Data Analysis*, *124*, 132–150. https://doi.org/10.1016/j.csda.2018.03.003

Hughes, L., Dwivedi, Y. K., Rana, N. P., Williams, M. D., & Raghavan, V. (2022). Perspectives on the future of manufacturing within the Industry 4.0 era. *Production Planning & Control*, *33*(2–3), 138–158. https://doi.org/10.1080/09537287.2020.1810762

Iliuță, M., Pop, E., Caramihai, S. I., & Moisescu, M. A. (2023). A Digital Twin Generic Architecture for Data-Driven Cyber-Physical Production Systems. In T. Borangiu, D. Trentesaux, & P. Leitão (Eds.), *Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future* (pp. 71–82). Springer International Publishing. https://doi.org/10.1007/978-3-031-24291-5_6

Ivanov, D., Tang, C. S., Dolgui, A., Battini, D., & Das, A. (2021). Researchers' perspectives on Industry 4.0: multi-disciplinary analysis and opportunities for operations management. *International Journal of Production Research*, *59*(7), 2055–2078. https://doi.org/10.1080/00207543.2020.1798035

J, A. K., & S, A. (2018). Aspect-based opinion ranking framework for product reviews using a Spearman's rank correlation coefficient method. *Information Sciences*, *460–461*, 23–41. https://doi.org/10.1016/j.ins.2018.05.003

Jahromi, A. A., & Kundur, D. (2020). *Fundamentals of Cyber-Physical Systems BT - Cyber-Physical Systems in the Built Environment* (C. J. Anumba & N. Roofigari-Esfahan (eds.); pp. 1–13). Springer International Publishing. https://doi.org/10.1007/978-3-030-41560-0_1

Javaid, M., Haleem, A., Singh, R. P., & Suman, R. (2021). Artificial Intelligence Applications for Industry 4.0: A Literature-Based Study. *Journal of Industrial Integration and Management*, *07*(01), 83–111. https://doi.org/10.1142/S2424862221300040

Jeon, B., & Suh, S.-H. (2018). Design Considerations and Architecture for Cooperative Smart Factory: MAPE/BD Approach. *Procedia Manufacturing*, *26*, 1094–1106. https://doi.org/10.1016/j.promfg.2018.07.146

Jha, R., Patra, P. K., & Srivastava, A. K. (2023). AI-guided optimization of manufacturing protocols for AHSS coils. *Materials and Manufacturing Processes*, *38*(2), 151–158. https://doi.org/10.1080/10426914.2022.2105871

Ji, F., Rabe-Hesketh, S., & Skrondal, A. (2023). Diagnosing and Handling Common Violations of Missing at Random. *Psychometrika*. https://doi.org/10.1007/s11336-022-09896-0

Kagermann, H., & Wahlster, W. (2022). Ten Years of Industrie 4.0. *Sci*, *4*(3). https://doi.org/10.3390/sci4030026

Kamble, S. S., Gunasekaran, A., & Sharma, R. (2018). Analysis of the driving and dependence power of barriers to adopt industry 4.0 in Indian manufacturing industry. *Computers in Industry*, *101*, 107–119. https://doi.org/https://doi.org/10.1016/j.compind.2018.06.004

Karras, A., Karras, C., Schizas, N., Avlonitis, M., & Sioutas, S. (2023). AutoML with Bayesian Optimizations for Big Data Management. In *Information* (Vol. 14, Issue 4). https://doi.org/10.3390/info14040223

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *ArXiv Preprint ArXiv:1412.6980*.

Kirmse, A., Kraus, V., Langer, T., Pomp, A., & Meisen, T. (2019). How To RAMI

4.0: Towards An Agent-based Information Management Architecture. *2019 International Conference on High Performance Computing & Simulation (HPCS)*, 961–968. https://doi.org/10.1109/HPCS48598.2019.9188087

Klötzer, C., Weißenborn, J., & Pflaum, A. (2017). The Evolution of Cyber-Physical Systems as a Driving Force Behind Digital Transformation. *2017 IEEE 19th Conference on Business Informatics (CBI)*, *02*, 5–14. https://doi.org/10.1109/CBI.2017.8

Kurniadi, K. A., Lee, S., & Ryu, K. (2018). *Digital Twin Approach for Solving Reconfiguration Planning Problems in RMS BT - Advances in Production Management Systems. Smart Manufacturing for Industry 4.0* (I. Moon, G. M. Lee, J. Park, D. Kiritsis, & G. von Cieminski (eds.); pp. 327–334). Springer International Publishing.

Kurt, R. (2019). Industry 4.0 in Terms of Industrial Relations and Its Impacts on Labour Life. *Procedia Computer Science*, *158*, 590–601. https://doi.org/10.1016/j.procs.2019.09.093

Leng, J., Liu, Q., Ye, S., Jing, J., Wang, Y., Zhang, C., Zhang, D., & Chen, X. (2020). Digital twin-driven rapid reconfiguration of the automated manufacturing system via an open architecture model. *Robotics and Computer-Integrated Manufacturing*, *63*, 101895. https://doi.org/https://doi.org/10.1016/j.rcim.2019.101895

Leng, J., Wang, D., Shen, W., Li, X., Liu, Q., & Chen, X. (2021). Digital twins-based smart manufacturing system design in Industry 4.0: A review. *Journal of Manufacturing Systems*, *60*, 119–137. https://doi.org/https://doi.org/10.1016/j.jmsy.2021.05.011

Leng, J., Zhang, H., Yan, D., Liu, Q., Chen, X., & Zhang, D. (2019). Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop. *Journal of Ambient Intelligence and Humanized Computing*, *10*(3), 1155–1166. https://doi.org/10.1007/s12652-018-0881-5

Li, L., Lei, B., & Mao, C. (2022). Digital twin in smart manufacturing. *Journal of Industrial Information Integration*, *26*, 100289. https://doi.org/10.1016/j.jii.2021.100289

Li, Q., Tang, Q., Chan, I., Wei, H., Pu, Y., Jiang, H., Li, J., & Zhou, J. (2018). Smart manufacturing standardization: Architectures, reference models and standards framework. *Computers in Industry*, *101*, 91–106. https://doi.org/10.1016/j.compind.2018.06.005

Lotfi, E., & Rezaee, A. A. (2018). A competitive functional link artificial neural network as a universal approximator. *Soft Computing*, *22*(14), 4613–4625. https://doi.org/10.1007/s00500-017-2644-1

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2019). Learning under Concept Drift: A Review. *IEEE Transactions on Knowledge and Data Engineering*, *31*(12), 2346–2363. https://doi.org/10.1109/TKDE.2018.2876857

Lu, Y., & Asghar, M. R. (2020). Semantic communications between distributed cyber-physical systems towards collaborative automation for smart

manufacturing. *Journal of Manufacturing Systems*, *55*, 348–359. https://doi.org/10.1016/j.jmsy.2020.05.001

Lu, Y., Morris, K. C., & Frechette, S. (2015). Standards landscape and directions for smart manufacturing systems. *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, 998–1005. https://doi.org/10.1109/CoASE.2015.7294229

Lugaresi, G., & Matta, A. (2021). Automated manufacturing system discovery and digital twin generation. *Journal of Manufacturing Systems*, *59*, 51–66. https://doi.org/10.1016/j.jmsy.2021.01.005

Luthra, S., Kumar, A., Zavadskas, E. K., Mangla, S. K., & Garza-Reyes, J. A. (2020). Industry 4.0 as an enabler of sustainability diffusion in supply chain: an analysis of influential strength of drivers in an emerging economy. *International Journal of Production Research*, *58*(5), 1505–1521. https://doi.org/10.1080/00207543.2019.1660828

Lv, Z., & Xie, S. (2022). Artificial intelligence in the digital twins: State of the art, challenges, and future research topics [version 2; peer review: 2 approved]. *Digital Twin*, *1*, 12. https://doi.org/10.12688/digitaltwin.17524.2

Madrigal, F., Maurice, C., & Lerasle, F. (2019). Hyper-parameter optimization tools comparison for multiple object tracking applications. *Machine Vision and Applications*, *30*(2), 269–289. https://doi.org/10.1007/s00138-018-0984-1

Mäkinen, S., Skogström, H., Laaksonen, E., & Mikkonen, T. (2021). Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help? *2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN)*, 109–112. https://doi.org/10.1109/WAIN52551.2021.00024

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene~Brevdo, Zhifeng~Chen, Craig~Citro, Greg~S.~Corrado, Andy~Davis, Jeffrey~Dean, Matthieu~Devin, Sanjay~Ghemawat, Ian~Goodfellow, Andrew~Harp, Geoffrey~Irving, Michael~Isard, Jia, Y., Rafal~Jozefowicz, Lukasz~Kaiser, Manjunath~Kudlur, … Xiaoqiang~Zheng. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. https://www.tensorflow.org/

Masood, T., & Egger, J. (2019). Augmented reality in support of Industry 4.0—Implementation challenges and success factors. *Robotics and Computer-Integrated Manufacturing*, *58*, 181–195. https://doi.org/10.1016/j.rcim.2019.02.003

Masood, T., & Sonntag, P. (2020). Industry 4.0: Adoption challenges and benefits for SMEs. *Computers in Industry*, *121*, 103261. https://doi.org/10.1016/j.compind.2020.103261

Mehrabi, M. G., Ulsoy, A. G., Koren, Y., & Heytler, P. (2002). Trends and perspectives in flexible and reconfigurable manufacturing systems. *Journal of Intelligent Manufacturing*, *13*(2), 135–146. https://doi.org/10.1023/A:1014536330551

Min, Q., Lu, Y., Liu, Z., Su, C., & Wang, B. (2019). Machine Learning based Digital Twin Framework for Production Optimization in Petrochemical Industry.

International Journal of Information Management, *49*, 502–519. https://doi.org/10.1016/j.ijinfomgt.2019.05.020

Mo, F., Rehman, H. U., Monetti, F. M., Chaplin, J. C., Sanderson, D., Popov, A., Maffei, A., & Ratchev, S. (2023). A framework for manufacturing system reconfiguration and optimisation utilising digital twins and modular artificial intelligence. *Robotics and Computer-Integrated Manufacturing*, *82*, 102524. https://doi.org/10.1016/j.rcim.2022.102524

Moeuf, A., Pellerin, R., Lamouri, S., Tamayo-Giraldo, S., & Barbaray, R. (2018). The industrial management of SMEs in the era of Industry 4.0. *International Journal of Production Research*, *56*(3), 1118–1136. https://doi.org/10.1080/00207543.2017.1372647

Moghaddam, M., Cadavid, M. N., Kenley, C. R., & Deshmukh, A. V. (2018). Reference architectures for smart manufacturing: A critical review. *Journal of Manufacturing Systems*, *49*, 215–225. https://doi.org/10.1016/j.jmsy.2018.10.006

Mohammed, W. M., Haber, R. E., & Martinez Lastra, J. L. (2022). Ontology-Driven Guidelines for Architecting Digital Twins in Factory Automation Applications. In *Machines* (Vol. 10, Issue 10). https://doi.org/10.3390/machines10100861

Montiel, J., Halford, M., Mastelini, S. M., Bolmier, G., Sourty, R., Vaysse, R., Zouitine, A., Gomes, H. M., Read, J., Abdessalem, T., & Bifet, A. (2020). River: machine learning for streaming data in Python. *CoRR*, *abs/2012.0*. https://arxiv.org/abs/2012.04740

Muhuri, P. K., Shukla, A. K., & Abraham, A. (2019). Industry 4.0: A bibliometric analysis and detailed overview. *Engineering Applications of Artificial Intelligence*, *78*, 218–235. https://doi.org/10.1016/j.engappai.2018.11.007

Mukkoti, V. V., Mohanty, C. P., Gandla, S., Sarkar, P., P, S. R., & B, D. (2020). Optimization of process parameters in CNC milling of P20 steel by cryo-treated tungsten carbide tools using NSGA-II. *Production & Manufacturing Research*, *8*(1), 291–312. https://doi.org/10.1080/21693277.2020.1790436

Napoleone, A., Macchi, M., & Pozzetti, A. (2020). A review on the characteristics of cyber-physical systems for the future smart factories. *Journal of Manufacturing Systems*, *54*, 305–335. https://doi.org/10.1016/j.jmsy.2020.01.007

Nasir, I. M., Khan, M. A., Yasmin, M., Shah, J. H., Gabryel, M., Scherer, R., & Damaševičius, R. (2020). Pearson Correlation-Based Feature Selection for Document Classification Using Balanced Training. In *Sensors* (Vol. 20, Issue 23). https://doi.org/10.3390/s20236793

Onaji, I., Tiwari, D., Soulatiantork, P., Song, B., & Tiwari, A. (2022). Digital twin in manufacturing: conceptual framework and case studies. *International Journal of Computer Integrated Manufacturing*, *35*(8), 831–858. https://doi.org/10.1080/0951192X.2022.2027014

Osswald, P. V., Mustafa, S. K., Kaa, C., Obst, P., Friedrich, M., Pfeil, M., Rietzel, D., & Witt, G. (2020). Optimization of the production processes of powder-

based additive manufacturing technologies by means of a machine learning model for the temporal prognosis of the build and cooling phase. *Production Engineering*, *14*(5), 677–691. https://doi.org/10.1007/s11740-020-00987-4

Ozaki, Y., Tanigaki, Y., Watanabe, S., & Onishi, M. (2020). Multiobjective Tree-Structured Parzen Estimator for Computationally Expensive Optimization Problems. *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, 533–541. https://doi.org/10.1145/3377930.3389817

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pivoto, D. G. S., de Almeida, L. F. F., da Rosa Righi, R., Rodrigues, J. J. P. C., Lugli, A. B., & Alberti, A. M. (2021). Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: A literature review. *Journal of Manufacturing Systems*, *58*, 176–192. https://doi.org/10.1016/j.jmsy.2020.11.017

Popkova, E. G., Ragulina, Y. V, & Bogoviz, A. V. (2019). *Fundamental Differences of Transition to Industry 4.0 from Previous Industrial Revolutions BT - Industry 4.0: Industrial Revolution of the 21st Century* (E. G. Popkova, Y. V Ragulina, & A. V Bogoviz (eds.); pp. 21–29). Springer International Publishing. https://doi.org/10.1007/978-3-319-94310-7_3

Qi, Q., & Tao, F. (2018). Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison. *IEEE Access*, *6*, 3585–3593. https://doi.org/10.1109/ACCESS.2018.2793265

Qi, Q., Tao, F., Hu, T., Anwer, N., Liu, A., Wei, Y., Wang, L., & Nee, A. Y. C. (2021). Enabling technologies and tools for digital twin. *Journal of Manufacturing Systems*, *58*, 3–21. https://doi.org/10.1016/j.jmsy.2019.10.001

Qiu, Y., Luo, J., & Deng, W. (2020). Deep Forest-based Product Completion Time Prediction Method in Discrete Manufacturing Industry: A Case Study. *2020 Chinese Automation Congress (CAC)*, 6776–6781. https://doi.org/10.1109/CAC51589.2020.9326760

Rai, R., Tiwari, M. K., Ivanov, D., & Dolgui, A. (2021). Machine learning in manufacturing and industry 4.0 applications. *International Journal of Production Research*, *59*(16), 4773–4778. https://doi.org/10.1080/00207543.2021.1956675

Raj, A., Dwivedi, G., Sharma, A., Lopes de Sousa Jabbour, A. B., & Rajak, S. (2020). Barriers to the adoption of industry 4.0 technologies in the manufacturing sector: An inter-country comparative perspective. *International Journal of Production Economics*, *224*, 107546. https://doi.org/10.1016/j.ijpe.2019.107546

Ramírez, J., Górriz, J. M., Ortiz, A., Martínez-Murcia, F. J., Segovia, F., Salas-Gonzalez, D., Castillo-Barnes, D., Illán, I. A., & Puntonet, C. G. (2018). Ensemble of random forests One vs. Rest classifiers for MCI and AD prediction

using ANOVA cortical and subcortical feature selection and partial least squares. *Journal of Neuroscience Methods*, *302*, 47–57. https://doi.org/10.1016/j.jneumeth.2017.12.005

Sarker, I. H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science*, *2*(6), 420. https://doi.org/10.1007/s42979-021-00815-1

Schuh, G., Anderl, R., Dumitrescu, R., Krüger, A., & ten Hompel, M. (Eds.). (2020). *Industrie 4.0 Maturity Index. Managing the Digital Transformation of Companies – UPDATE 2020 – (acatech STUDY)*. acatech - National Academy of Science and Engineering, Munich.

Semeraro, C., Lezoche, M., Panetto, H., & Dassisti, M. (2021). Digital twin paradigm: A systematic literature review. *Computers in Industry*, *130*, 103469. https://doi.org/10.1016/j.compind.2021.103469

Sharmin, S., Shoyaib, M., Ali, A. A., Khan, M. A. H., & Chae, O. (2019). Simultaneous feature selection and discretization based on mutual information. *Pattern Recognition*, *91*, 162–174. https://doi.org/10.1016/j.patcog.2019.02.016

Singh, M., Fuenmayor, E., Hinchy, E. P., Qiao, Y., Murray, N., & Devine, D. (2021). Digital Twin: Origin to Future. In *Applied System Innovation* (Vol. 4, Issue 2). https://doi.org/10.3390/asi4020036

Song, J., Zhang, Z., Tang, D., Zhu, H., Wang, L., & Nie, Q. (2023). Designing and modeling of self-organizing manufacturing system in a digital twin shop floor. *The International Journal of Advanced Manufacturing Technology*. https://doi.org/10.1007/s00170-023-10965-6

Spolaor, S., Fuchs, C., Cazzaniga, P., Kaymak, U., Besozzi, D., & Nobile, M. S. (2020). Simpful: A User-Friendly Python Library for Fuzzy Logic. *International Journal of Computational Intelligence Systems*, *13*(1), 1687–1698. https://doi.org/10.2991/ijcis.d.201012.002

Symeonidis, G., Nerantzis, E., Kazakis, A., & Papakostas, G. A. (2022). MLOps - Definitions, Tools and Challenges. *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, 453–460. https://doi.org/10.1109/CCWC54503.2022.9720902

Tamburri, D. A. (2020). Sustainable MLOps: Trends and Challenges. *2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 17–23. https://doi.org/10.1109/SYNASC51798.2020.00015

Tao, F., Qi, Q., Wang, L., & Nee, A. Y. C. (2019). Digital Twins and Cyber–Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison. *Engineering*, *5*(4), 653–661. https://doi.org/10.1016/j.eng.2019.01.014

Tao, F., Zhang, M., Liu, Y., & Nee, A. Y. C. (2018). Digital twin driven prognostics and health management for complex equipment. *CIRP Annals*, *67*(1), 169–172. https://doi.org/10.1016/j.cirp.2018.04.055

Taylor, N., Human, C., Kruger, K., Bekker, A., & Basson, A. (2020). *Comparison of Digital Twin Development in Manufacturing and Maritime Domains BT - Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future* (T. Borangiu, D. Trentesaux, P. Leitão, A. Giret Boggino, & V. Botti (Eds.); pp. 158–170). Springer International Publishing.

Tripathi, S. (2021). Artificial Intelligence: A Brief Review. In T. H. Musiolik & A. D. Cheok (Eds.), *Analyzing Future Applications of AI, Sensors, and Robotics in Society* (pp. 1–16). IGI Global. https://doi.org/10.4018/978-1-7998-3499-1.ch001

Truong, A., Walters, A., Goodsitt, J., Hines, K., Bruss, C. B., & Farivar, R. (2019). Towards Automated Machine Learning: Evaluation and Comparison of AutoML Approaches and Tools. *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, 1471–1479. https://doi.org/10.1109/ICTAI.2019.00209

Villalonga, A., Beruvides, G., Castaño, F., & Haber, R. E. (2020). Cloud-Based Industrial Cyber-Physical System for Data-Driven Reasoning: A Review and Use Case on an Industry 4.0 Pilot Line. *IEEE Transactions on Industrial Informatics*, *16*(9), 5975–5984. https://doi.org/10.1109/TII.2020.2971057

Villalonga, A., Negri, E., Biscardo, G., Castano, F., Haber, R. E., Fumagalli, L., & Macchi, M. (2021). A decision-making framework for dynamic scheduling of cyber-physical production systems based on digital twins. *Annual Reviews in Control*, *51*, 357–373. https://doi.org/10.1016/j.arcontrol.2021.04.008

Wan, J., Li, X., Dai, H.-N., Kusiak, A., Martínez-García, M., & Li, D. (2021). Artificial-Intelligence-Driven Customized Manufacturing Factory: Key Technologies, Applications, and Challenges. *Proceedings of the IEEE*, *109*(4), 377–398. https://doi.org/10.1109/JPROC.2020.3034808

Wang, B., Tao, F., Fang, X., Liu, C., Liu, Y., & Freiheit, T. (2021). Smart Manufacturing and Intelligent Manufacturing: A Comparative Review. *Engineering*, *7*(6), 738–757. https://doi.org/10.1016/j.eng.2020.07.017

Wei, S., Hu, J., Cheng, Y., Ma, Y., & Yu, Y. (2017). The essential elements of intelligent Manufacturing System Architecture. *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, 1006–1011. https://doi.org/10.1109/COASE.2017.8256234

Yadav, A., Mishra, S., & Sairam, A. S. (2022). A multi-objective worker selection scheme in crowdsourced platforms using NSGA-II. *Expert Systems with Applications*, *201*, 116991. https://doi.org/10.1016/j.eswa.2022.116991

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295–316. https://doi.org/10.1016/j.neucom.2020.07.061

Yu, H., Yu, D., Wang, C., Hu, Y., & Li, Y. (2023). Edge intelligence-driven digital twin of CNC system: Architecture and deployment. *Robotics and Computer-Integrated Manufacturing*, *79*, 102418. https://doi.org/10.1016/j.rcim.2022.102418

Zanjani, M. S., Salam, M. A., & Kandara, O. (2020). Data-Driven Hydrocarbon Production Forecasting Using Machine Learning Techniques. *International Journal of Computer Science and Information Security*, *18*(6), 65–72.

Zhang, C., Xu, W., Liu, J., Liu, Z., Zhou, Z., & Pham, D. T. (2019). A Reconfigurable Modeling Approach for Digital Twin-based Manufacturing System. *Procedia CIRP*, *83*, 118–125. https://doi.org/10.1016/j.procir.2019.03.141

Zhang, C., Xu, W., Liu, J., Liu, Z., Zhou, Z., & Pham, D. T. (2021). Digital twin-enabled reconfigurable modeling for smart manufacturing systems. *International Journal of Computer Integrated Manufacturing*, *34*(7–8), 709–733. https://doi.org/10.1080/0951192X.2019.1699256

Zhang, C., & Yang, J. (2020). *Second Industrial Revolution BT - A History of Mechanical Engineering* (C. Zhang & J. Yang (Eds.); pp. 137–195). Springer Singapore. https://doi.org/10.1007/978-981-15-0833-2_5

Zhang, K., Shi, Y., Karnouskos, S., Sauter, T., Fang, H., & Colombo, A. W. (2023). Advancements in Industrial Cyber-Physical Systems: An Overview and Perspectives. *IEEE Transactions on Industrial Informatics*, *19*(1), 716–729. https://doi.org/10.1109/TII.2022.3199481

Zhang, Y., Zhang, Y., Mao, X., Wu, Y., Lin, B., & Wang, S. (2022). Recommending Base Image for Docker Containers based on Deep Configuration Comprehension. *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 449–453. https://doi.org/10.1109/SANER53432.2022.00060

Zhou, G., Zhang, C., Li, Z., Ding, K., & Wang, C. (2020). Knowledge-driven digital twin manufacturing cell towards intelligent manufacturing. *International Journal of Production Research*, *58*(4), 1034–1051. https://doi.org/10.1080/00207543.2019.1607978

Zhou, K., Liu, T., & Zhou, L. (2015). Industry 4.0: Towards future industrial opportunities and challenges. *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 2147–2152. https://doi.org/10.1109/FSKD.2015.7382284

Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, *8*(2), 173–195. https://doi.org/10.1162/106365600568202

Anexos

Anexo A. Estructura de los ficheros de datos y configuración

Estructura del fichero de datos:

{“inputs”: {
“feature 1 name”: [f_{11}, f_{12}, …, f_{1N}],
“feature 2 name”: [f_{21}, f_{22}, …, f_{2N}],
…,
“feature i name”: [f_{i1}, f_{i2}, …, f_{iN}]},
“outputs”: {
“KPI 1 name”: [k_{11}, k_{12}, …, k_{1N}],
“KPI 2 name”: [k_{21}, k_{22}, …, k_{2N}],
…,
“KPI j name”: [k_{j1}, k_{j2}, …, k_{jN}]}}

Estructura del fichero de configuración:

{“ranges”: {
“feature 1 name”: [f_{1min}, f_{1max}],
“feature 2 name”: [f_{2min}, f_{2max}],
…,
“feature i name”: [f_{imin}, f_{imax}]},
“objectives”: {
“KPI 1 name”: “min”,
“KPI 2 name”: “max”,
…,
“KPI j name”: “min”},
“data source”: “path_to_data_file/data.json”}

Anexo B. Pruebas en problema ZDT4

El conjunto de problemas ZDT comprende seis funciones de prueba cuidadosamente seleccionadas, las cuales son usadas ampliamente como referencia en el campo de la optimización multiobjetivo (Zitzler et al., 2000). Específicamente, en el caso del problema ZDT4, se trata de un problema con 10 variables de entrada $X = [X_0, X_1, \dots, X_9]$, y con un frente óptimo de Pareto conocido de forma convexa. La dificultad de ZDT4 radica en que es un problema con muchos óptimos locales, por lo que los algoritmos pueden quedarse atrapados en uno de ellos con facilidad. La definición de ZDT4 es la siguiente:

$$minimizar\ y_0(X) = X_0 \tag{46}$$

$$minimizar\ y_1(X) = g(X)h(y_0(X), g(X)) \tag{47}$$

$$g(X) = 91 + \sum_{i=1}^{9}(X_i^2 - 10\cos(4\pi X_i)) \tag{48}$$

$$h(y_0, g) = 1 - \sqrt{\frac{y_0}{g}} \tag{49}$$

donde $0 \leq X_0 \leq 1$ y para el resto de las variables de entrada $-10 \leq X_i \leq 10$.

Utilizando estas funciones se creó un conjunto de datos con 10000 muestras aleatorias. El objetivo que se busca es que la metodología de aprendizaje automático automatizado sea capaz de generar los modelos a partir de las muestras generadas y utilizar dichos modelos para llevar a cabo la optimización, resultando en un frente de Pareto cercano al óptimo conocido. Para este problema, el tiempo total de ejecución de la metodología fue de aproximadamente 8241 s, de los cuales 7216 s corresponden a las operaciones hasta la selección de los modelos y 1025 s corresponden a la optimización del proceso productivo.

La estandarización de los datos permite transformar la escala de la variable X_0 a una escala similar al resto de las variables como se puede ver en la Figura B.1.

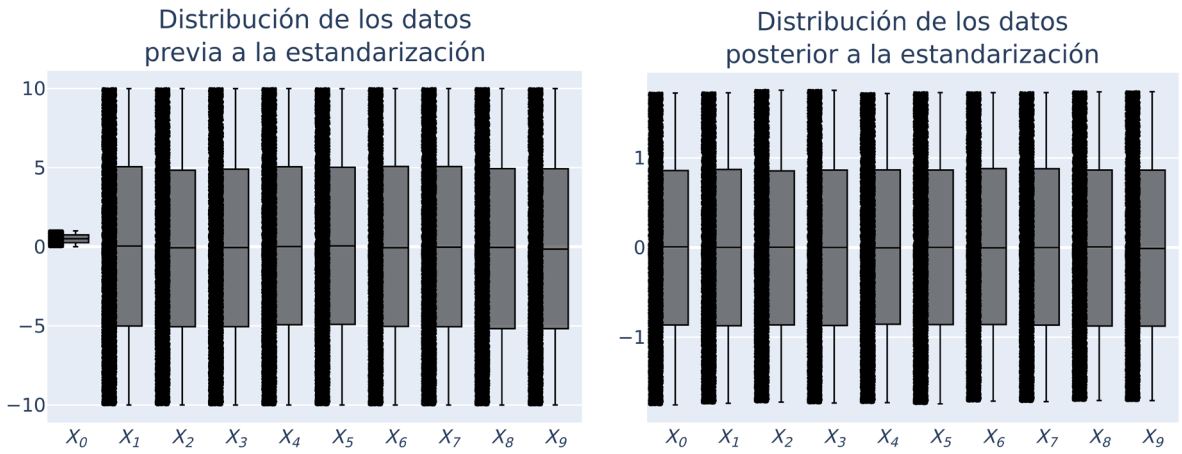

Figura B.1: Distribución de los datos antes y después de la estandarización

A continuación, se realiza el proceso de selección de variables para ambas salidas. En el caso de la primera salida ambos coeficientes de correlación son capaces de detectar la correlación perfecta entre la primera variable y esta salida, tal como se describe en la Ecuación (46). El resto de las variables son descartadas ya que no superan el umbral predefinido tal como se puede comprobar en la Figura B.2.

En el caso de la segunda salida ninguno de los dos coeficientes de correlación fue capaz de detectar alguna relación significativa entre las variables de entrada y las salidas, como se puede comprobar en la Figura B.3. Esto se debe a la gran complejidad de la función utilizada para generar esta salida. Tal como se propuso en la metodología, en caso de que ninguna variable supere el umbral predefinido, se utilizan todas las variables de entrada para crear los modelos de la salida correspondiente.

Una vez que se han seleccionado las variables para modelar cada KPI, se ejecuta el proceso de optimización de hiperparámetros y selección de modelos. A continuación, en las siguientes figuras (Figura B.4, Figura B.5, Figura B.6 y Figura B.7) se muestra información relacionada con la optimización de hiperparámetros de MLP, SVR, XGBoost y *k*NN, respectivamente, para la salida y_0.

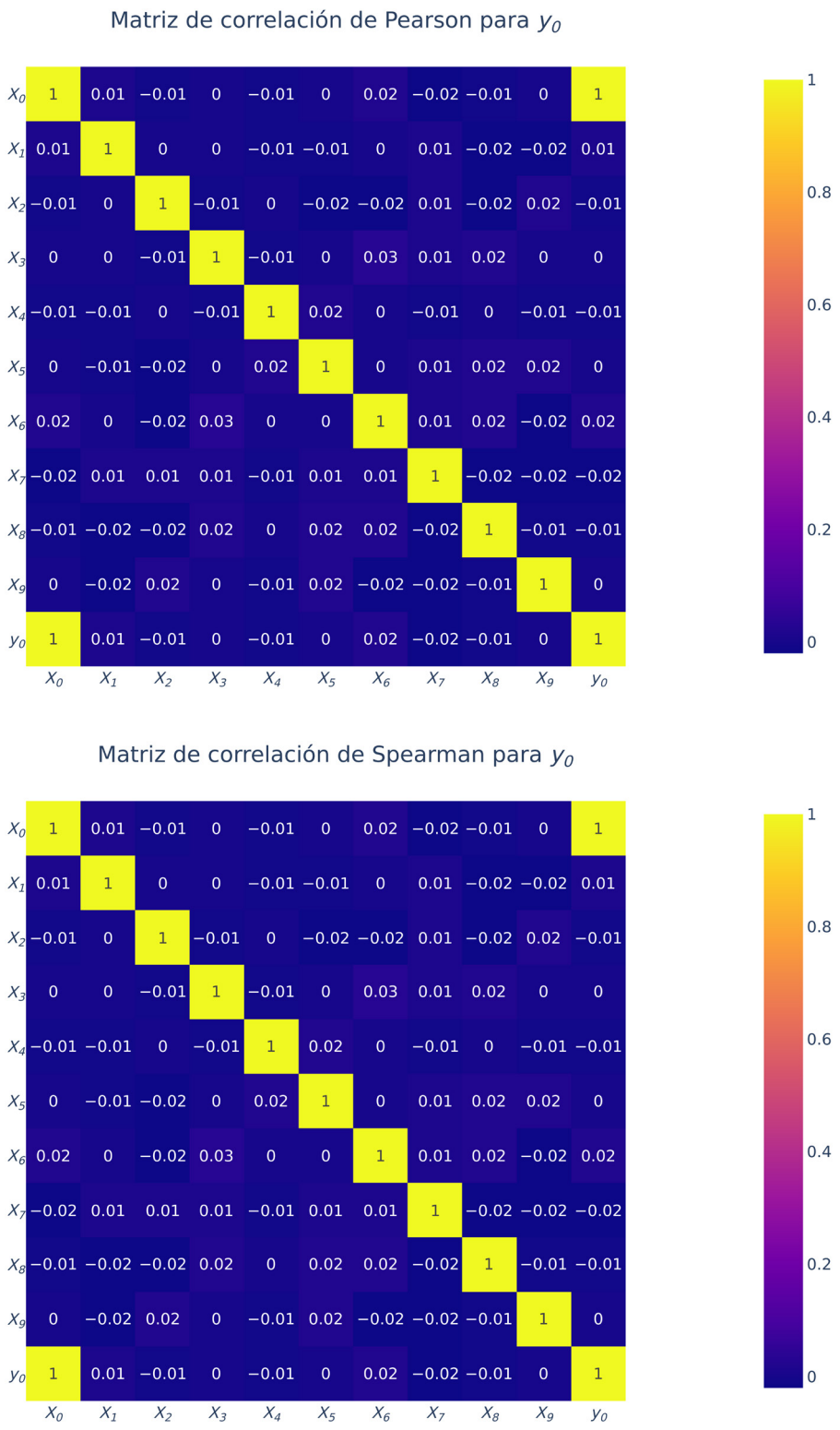

Figura B.2: Matrices de correlación para la salida y_0

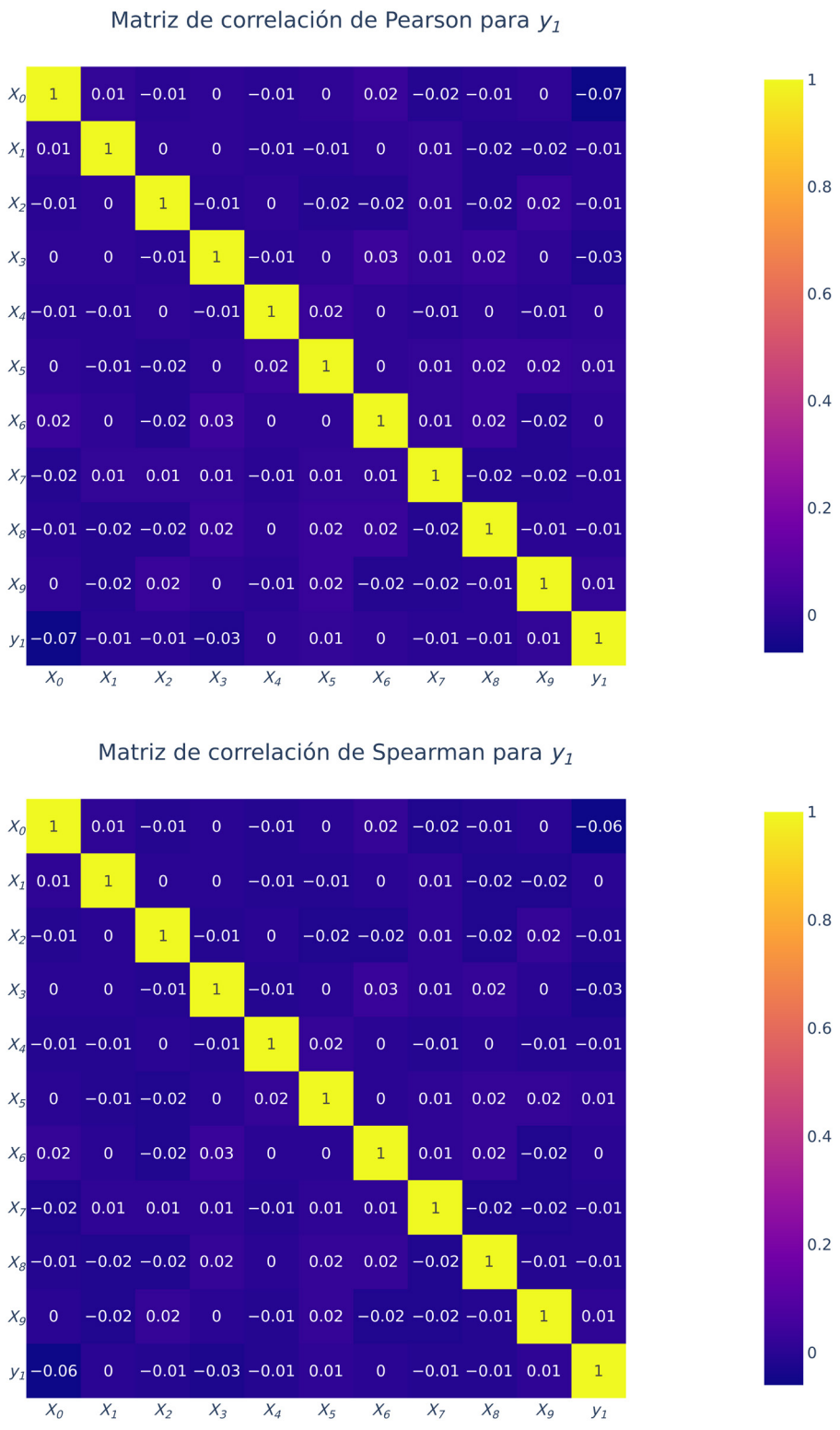

Figura B.3: Matrices de correlación para la salida y_1

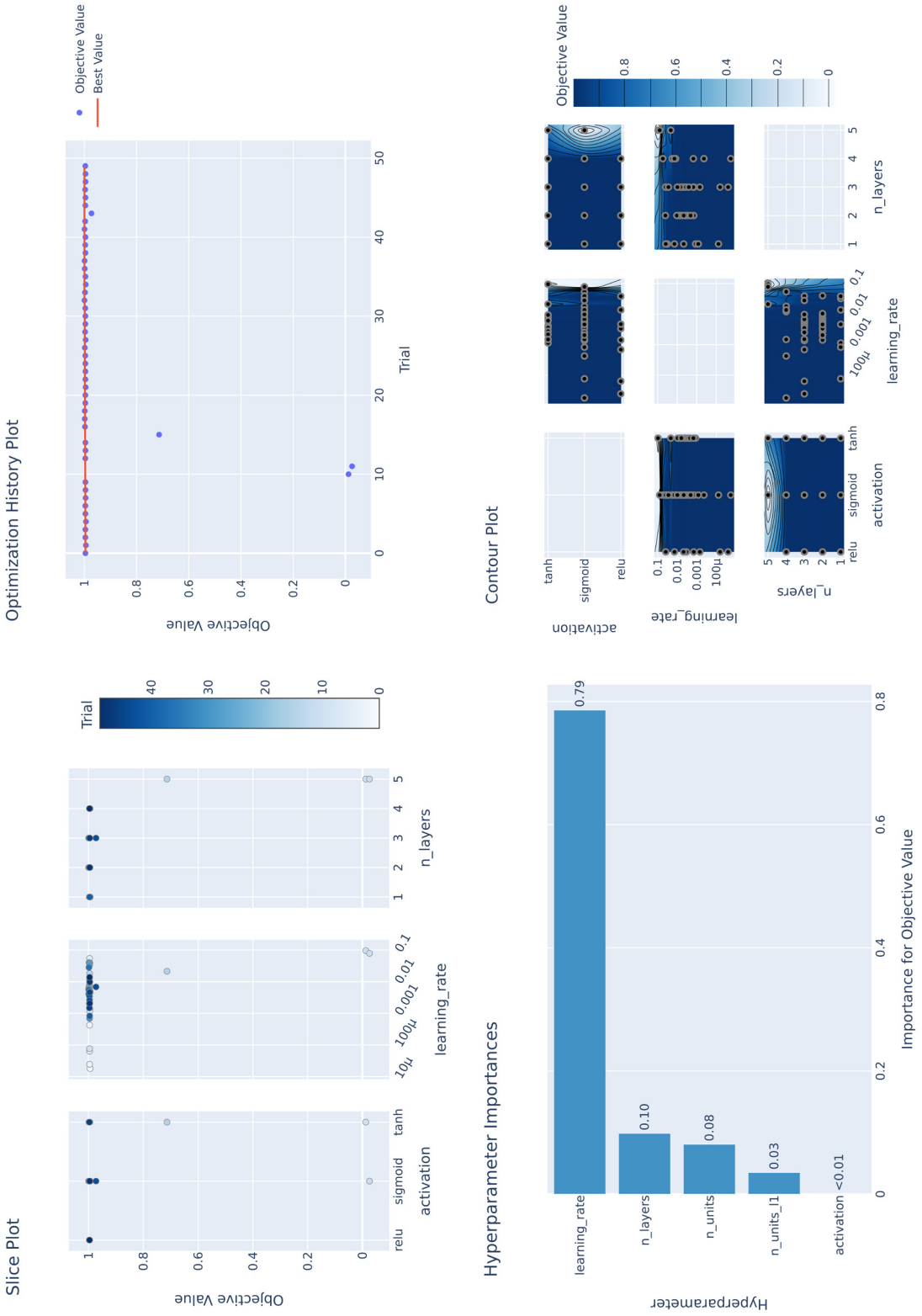

Figura B.4: Optimización de hiperparámetros de MLP para salida y_0

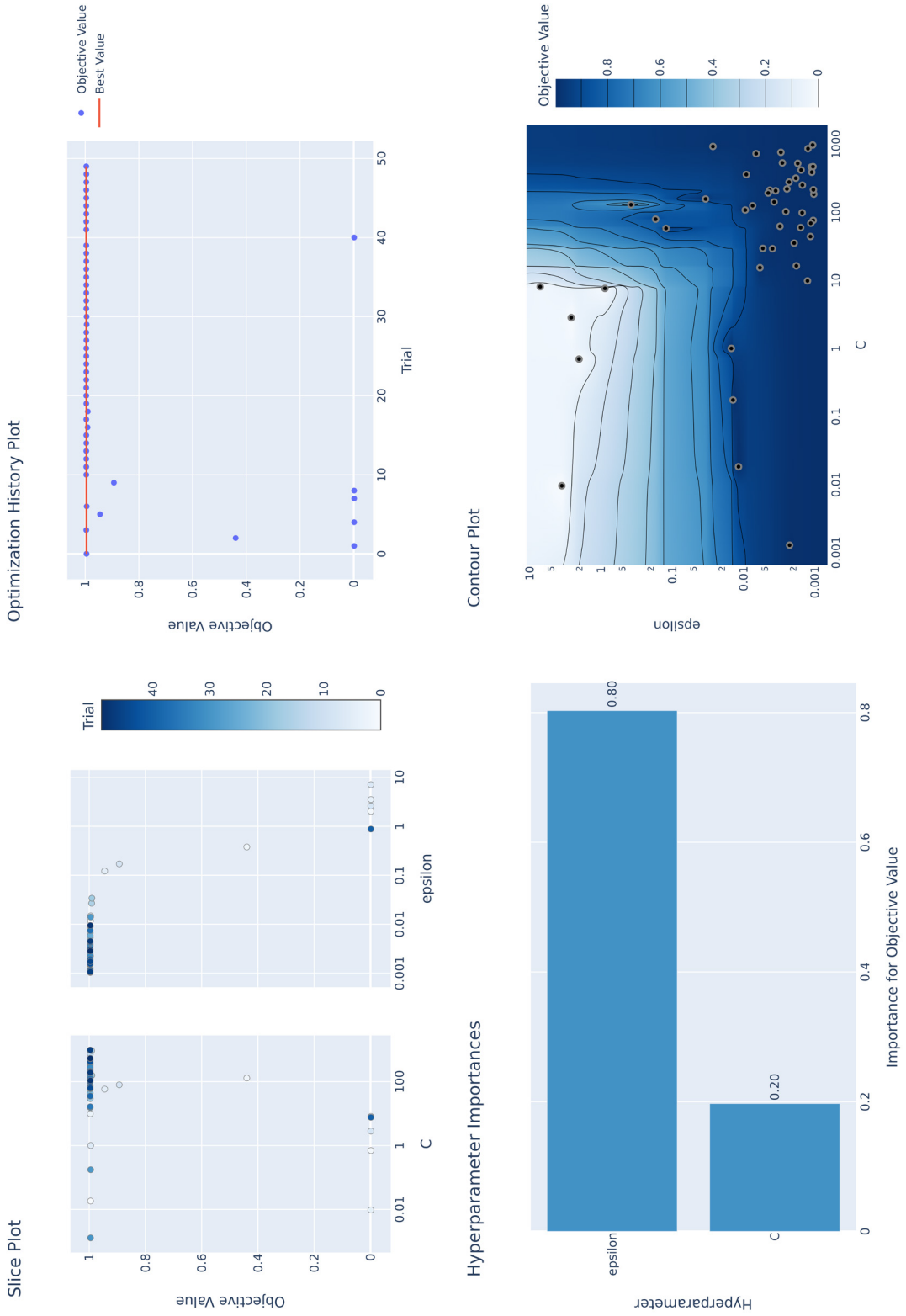

Figura B.5: Optimización de hiperparámetros de SVR para salida y_0

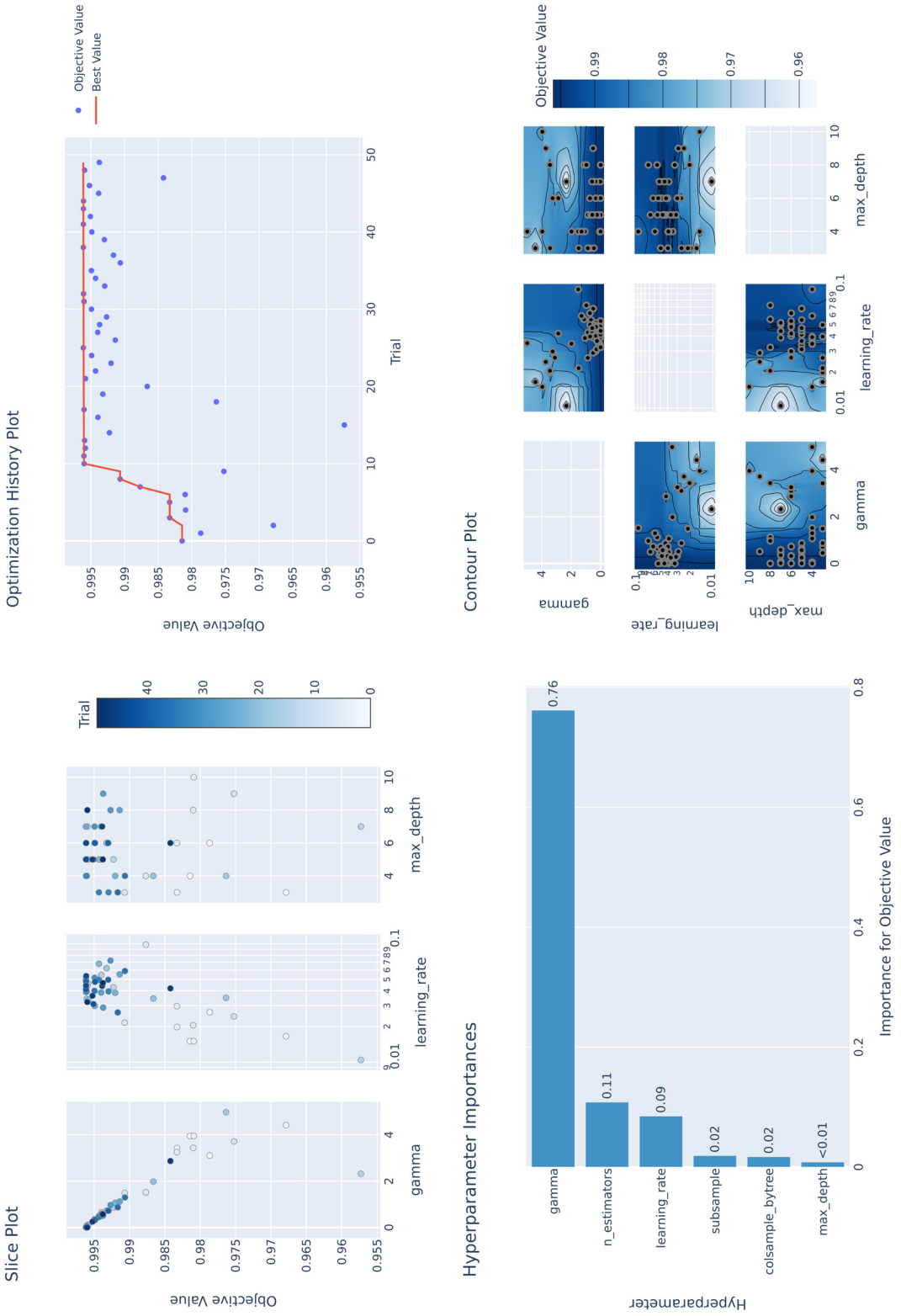

Figura B.6: Optimización de hiperparámetros de XGBoost para y_0

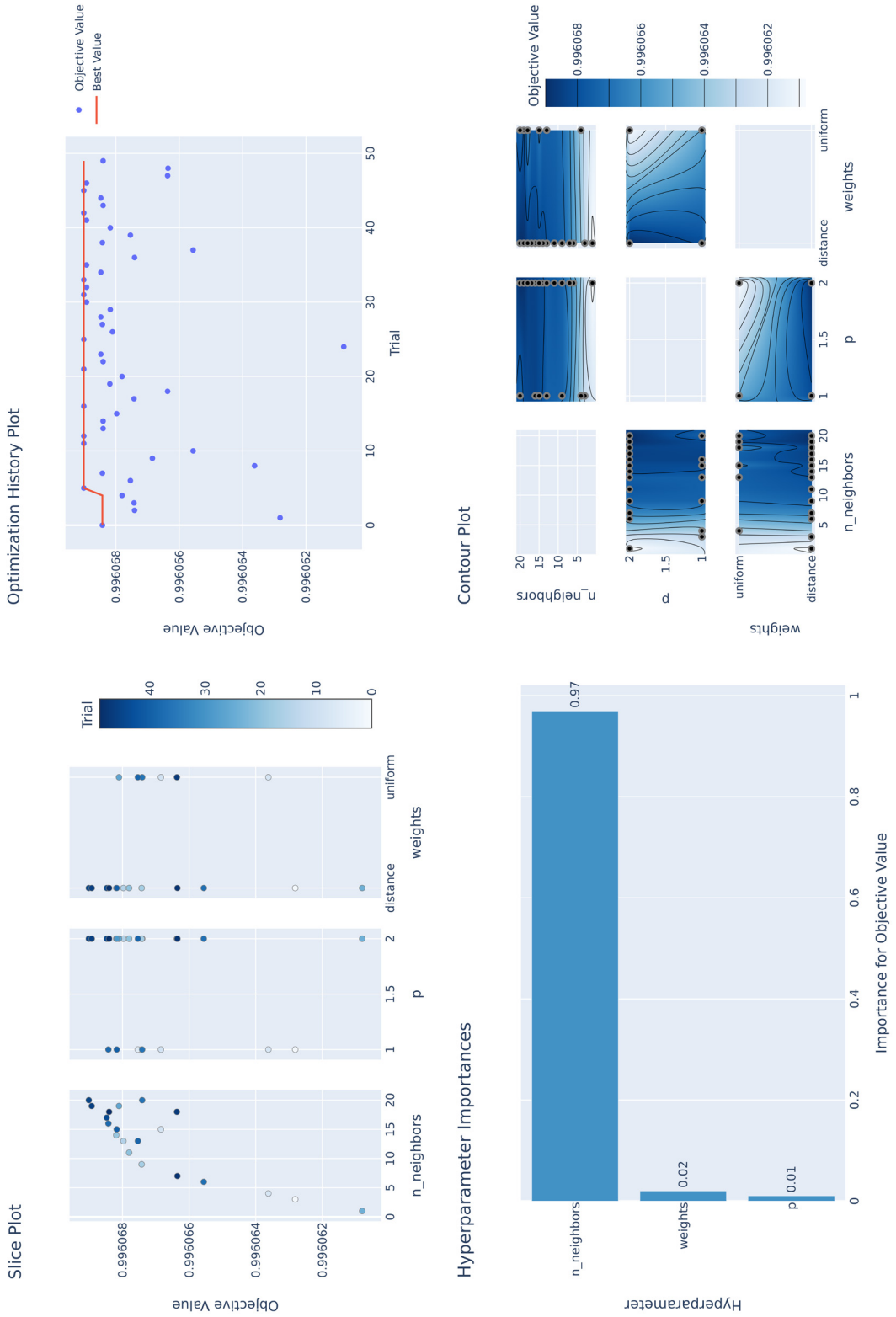

Figura B.7: Optimización de hiperparámetros de kNN para salida y_0

Una vez que se han optimizado los hiperparámetros de los cuatro modelos base, se procede a la selección del modelo para la salida y_0. En esta etapa también se

consideran las estrategias de conjunto de modelos y metamodelado. La Tabla B.1 resume los resultados obtenidos. El modelo seleccionado para la salida y_0 fue el metamodelo ya que obtuvo el mayor valor de R^2. Se puede notar que debido a la sencillez de la función utilizada para crear la salida y_0 todos los modelos demostraron una alta capacidad de generalización. La Figura B.8 muestra las predicciones de los modelos contra los valores reales para el conjunto de datos de prueba.

Tabla B.1: Hiperparámetros seleccionados y resultados de validación para la salida y_0

Modelo	**Hiperparámetros**	**R^2 validación**	**R^2 prueba**	**Tiempo de optimización de hiperparámetros / entrenamiento**	**Tiempo de inferencia**
MLP	número de capas ocultas = 2, σ = sigmoide, número de unidades por capa = [100, 21, 38], $\propto$ = 0.006	0.999	0.999	3137.428 s	122.502 ms
SVR	C= 214.773, ε = 0.001	0.996	0.996	**7.367 s**	7.154 ms
XGBoost	número de árboles = 201, profundidad máxima = 7, tasa de aprendizaje = 0.050, tasa de submuestreo = 0.777, tasa de submuestreo de columnas por árbol = 0.834, γ = 0.014	0.996	0.996	38.094 s	**2.178 ms**
*k*NN	k = 20, función de ponderación = distancia, orden de la distancia de Minkowski = 2	0.996	0.996	0.615 s	2.329 ms
Conjunto de modelos	-	0.997	0.998	-	148.641 ms
Metamodelo	-	**0.999**	**0.999**	0.017 s	188.754 ms

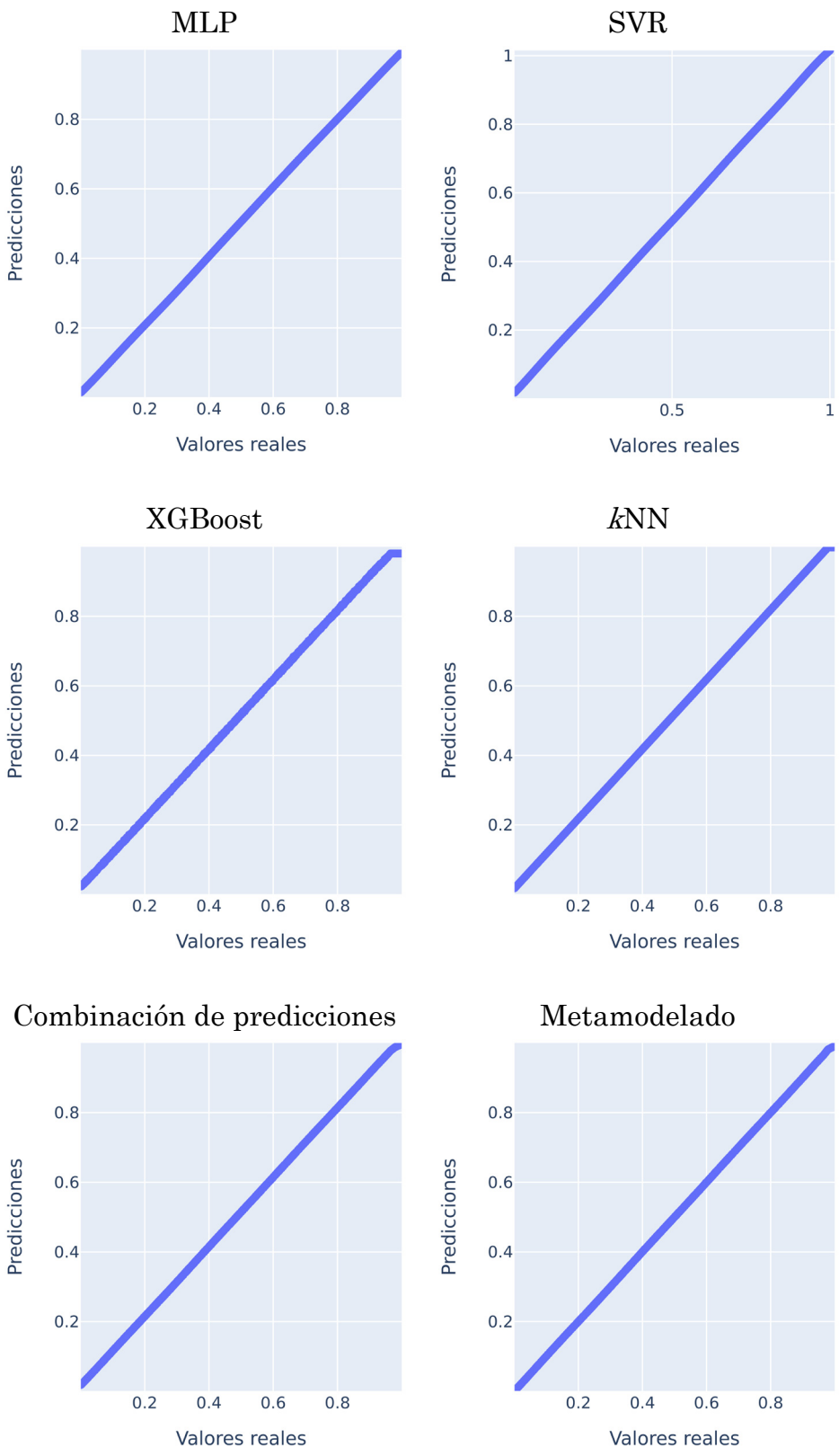

Figura B.8: Valores reales vs. predichos por los diferentes modelos para y_0

Una vez que se ha seleccionado el modelo para la salida y_0, se repite el proceso de optimización de hiperparámetros y selección de modelo para la próxima salida. Las siguientes figuras (Figura B.9, Figura B.10, Figura B.11 y Figura B.12) muestran información relacionada con la optimización de hiperparámetros de MLP, SVR, XGBoost y *k*NN, respectivamente, para la salida y_1.

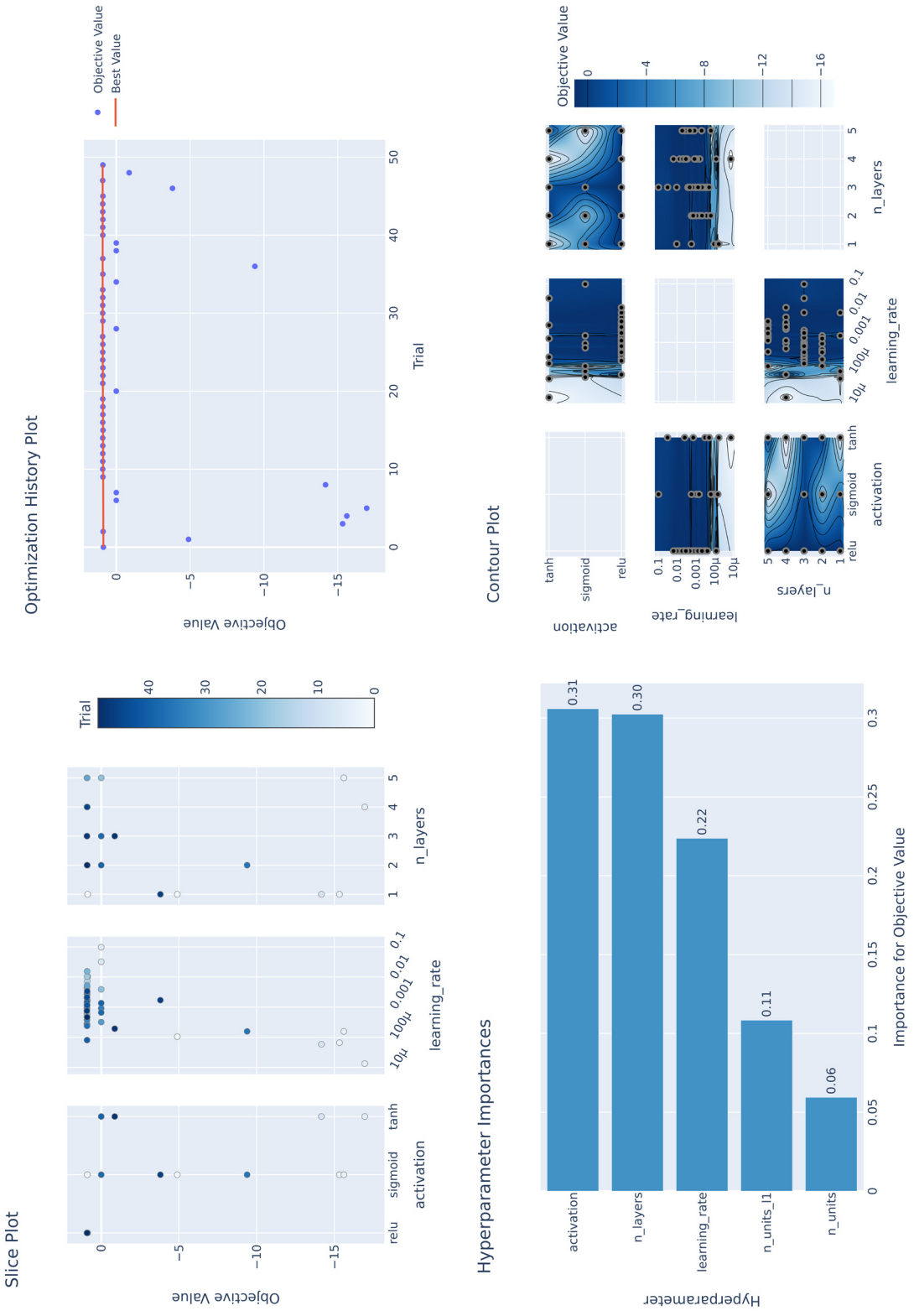

Figura B.9: Optimización de hiperparámetros de MLP para salida y_1

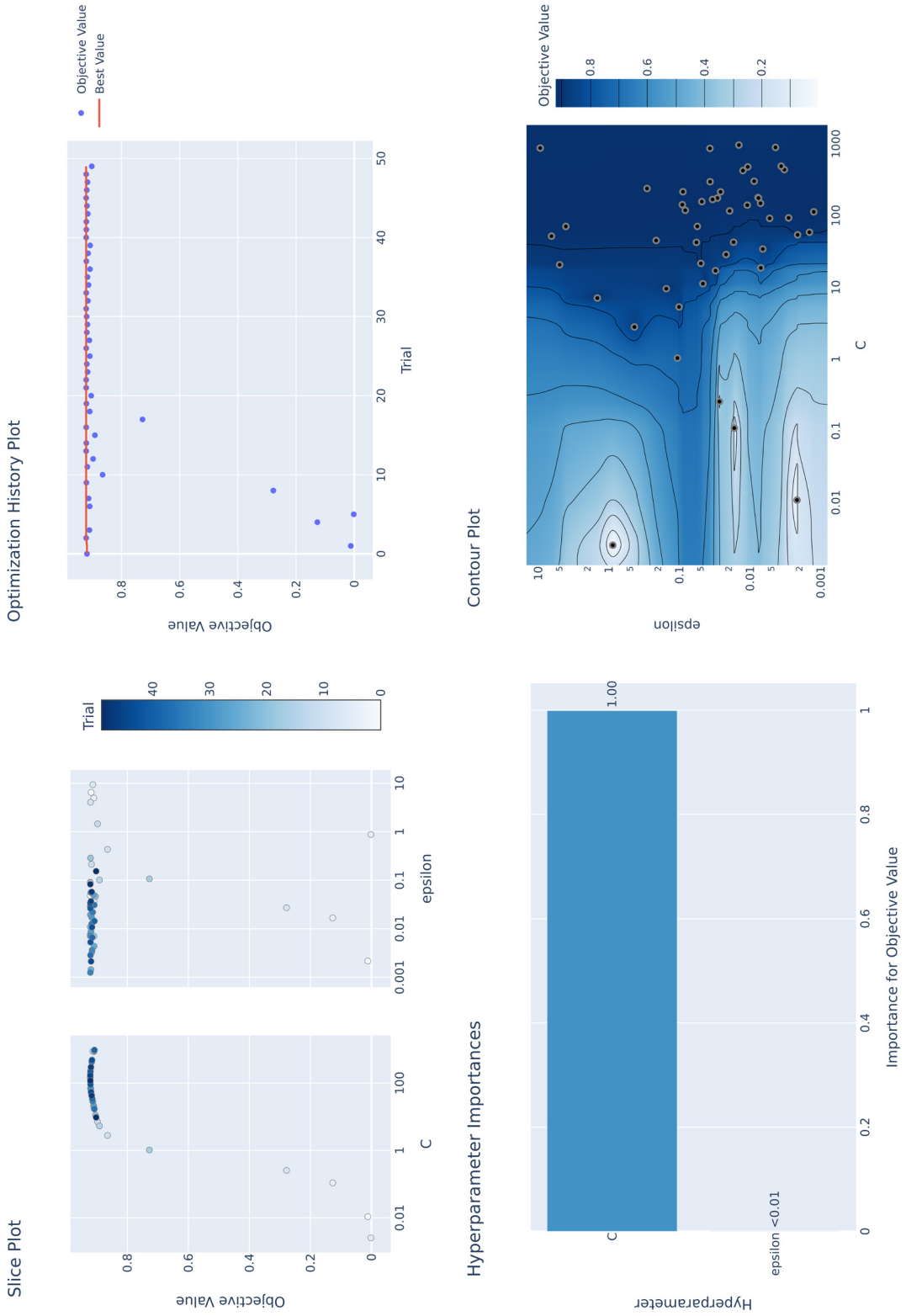

Figura B.10: Optimización de hiperparámetros de SVR para salida y_1

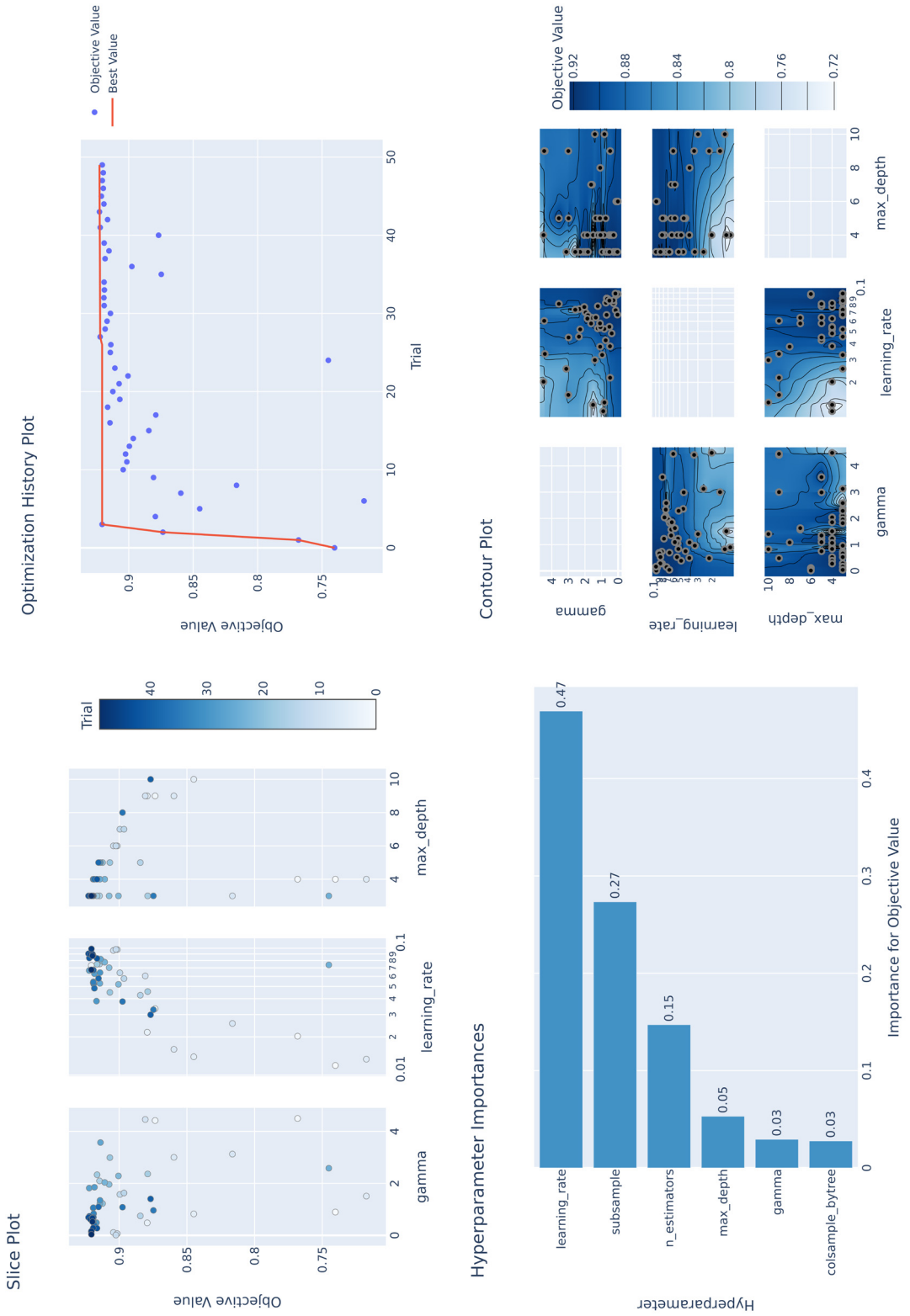

Figura B.11: Optimización de hiperparámetros de XGBoost para y_1

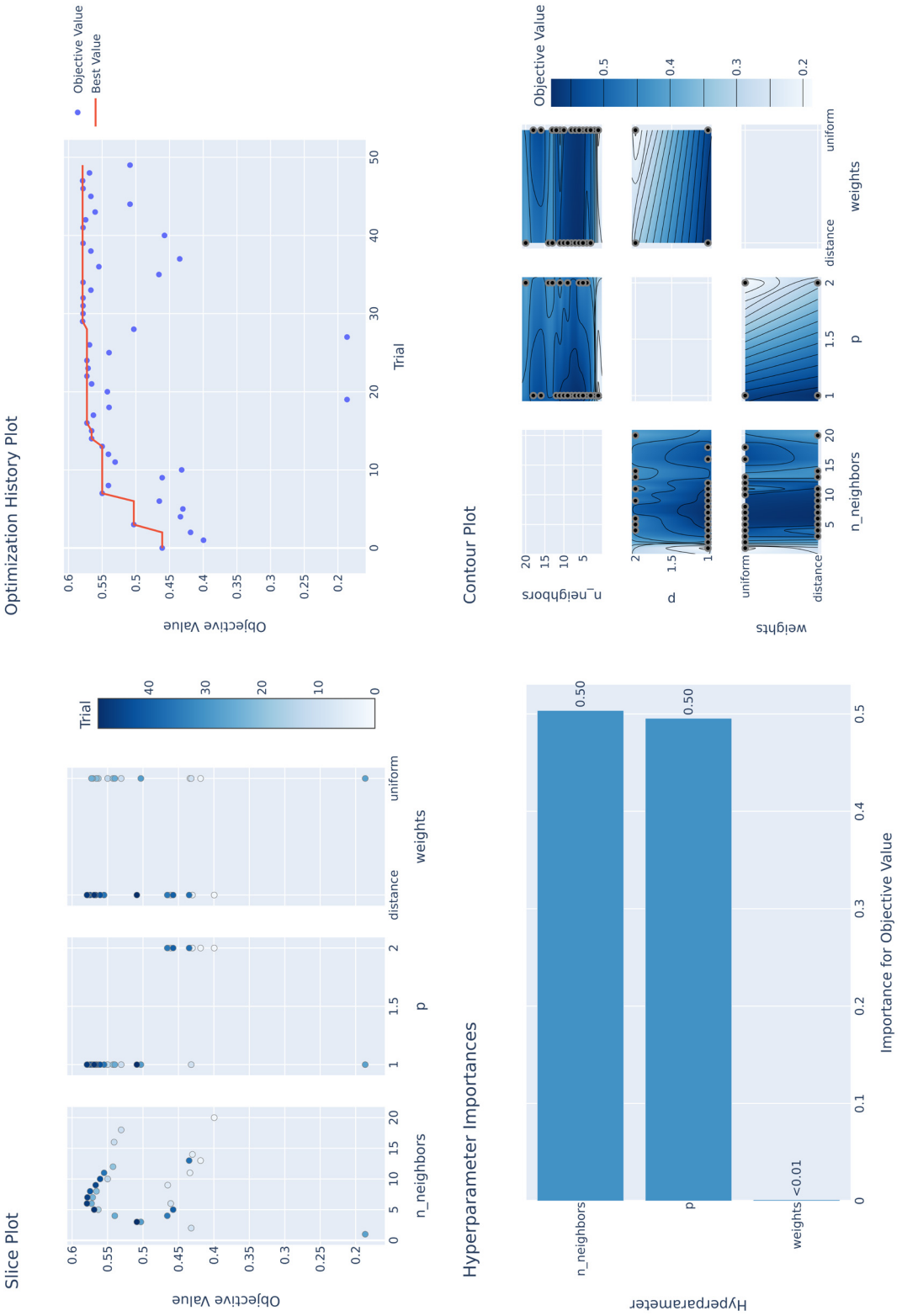

Figura B.12: Optimización de hiperparámetros de *k*NN para salida y_1

Una vez que se han optimizado los hiperparámetros de los cuatro modelos base, se procede a la selección del modelo para la salida y_1. La Tabla B.2 resume los

resultados obtenidos. El modelo seleccionado para la salida y_1 también fue el metamodelo. En este caso, a pesar de que la salida y_1 está representada por una función de alta complejidad, la mayoría de los modelos, excepto *k*NN, demostraron una alta capacidad de generalización. La Figura B.13 muestra las predicciones de los modelos contra los valores reales para el conjunto de datos de prueba.

Tabla B.2: Hiperparámetros seleccionados y resultados de validación para la salida y_1

Modelo	**Hiperparámetros**	**R^2 validación**	**R^2 prueba**	**Tiempo de optimización de hiperparámetros / entrenamiento**	**Tiempo de inferencia**
MLP	número de capas ocultas = 3, σ = relu, número de unidades por capa = [102, 33, 94], $\propto$ = 0.000	0.924	0.923	3401.536 s	141.117 ms
SVR	C= 156.125, ε = 0.048	0.921	0.925	341.111 s	231.773 ms
XGBoost	número de árboles = 448, profundidad máxima = 3, tasa de aprendizaje = 0.091, tasa de submuestreo = 0.588, tasa de submuestreo de columnas por árbol = 0.622, γ = 0.687	0.923	0.923	69.335 s	**4.963 ms**
*k*NN	k = 6, función de ponderación = distancia, orden de la distancia de Minkowski = 1	0.579	0.555	**5.897 s**	114.522 ms
Conjunto de modelos	-	0.911	0.911	-	494.691 ms
Metamodelo	-	**0.931**	**0.933**	0.001 s	498.843 ms

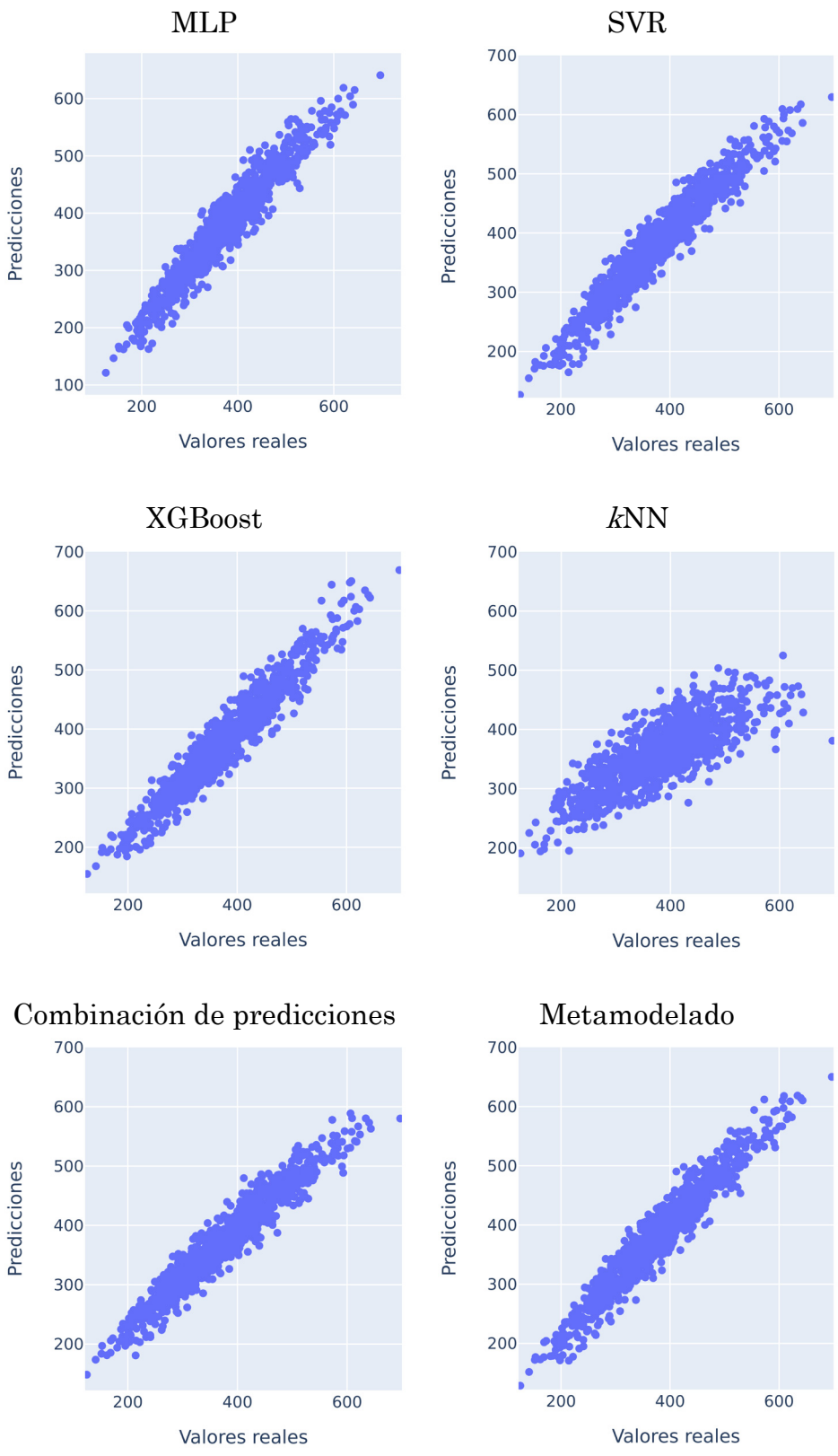

Figura B.13: Valores reales vs. predichos por los diferentes modelos para y_1

Finalmente, una vez que se tienen los modelos para ambas salidas, se ejecuta a la optimización multiobjetivo mediante NSGA-II con una población de 100 individuos y durante 5000 generaciones. Este problema tiene una frontera de Pareto convexa conocida tal como se muestra en la Figura B.14. Tras efectuar la optimización se obtiene la frontera de Pareto estimada que se muestra en la Figura B.15. Aunque este resultado pueda parecer distante de la frontera de Pareto real, se debe recordar que la optimización se está realizando utilizando como funciones objetivo los modelos obtenidos a partir de muestras generadas aleatoriamente, no las funciones originales de ZDT4 directamente. Aunque los modelos obtenidos demostraron una alta capacidad de generalización, estos no lograron un ajuste perfecto. Además, otro de los factores a tener en cuenta, es que ZDT4 cuenta con múltiples óptimos locales, por lo que es un problema complejo. Una posible solución para este inconveniente puede ser aumentar el tamaño de la población o el número de generaciones; sin embargo, en la metodología se eligieron los valores especificados para estos parámetros como una solución de compromiso, ya que aumentar cualquiera de ellos implicaría un aumento del coste computacional.

A pesar de las limitaciones descritas anteriormente, al analizar los datos con los que se crearon los modelos, los cuales forman parte del espacio realizable de ZDT4, y compararlos con la frontera de Pareto tal como se muestra en la Figura B.16, se puede tener una mejor perspectiva de la efectividad de la metodología propuesta para, primeramente, representar las relaciones entre las variables de entrada y salida y, en segundo lugar, llevar a cabo el proceso de optimización multiobjetivo convergiendo a la frontera de Pareto real. Además, al analizar la Figura B.17 se puede observar como los valores de la mayoría de las variables del conjunto de Pareto obtenido se asemejan al conjunto de Pareto real. La mayor diferencia ocurre para la primera variable cuyos valores se agrupan en una zona cercana al límite inferior del rango predefinido.

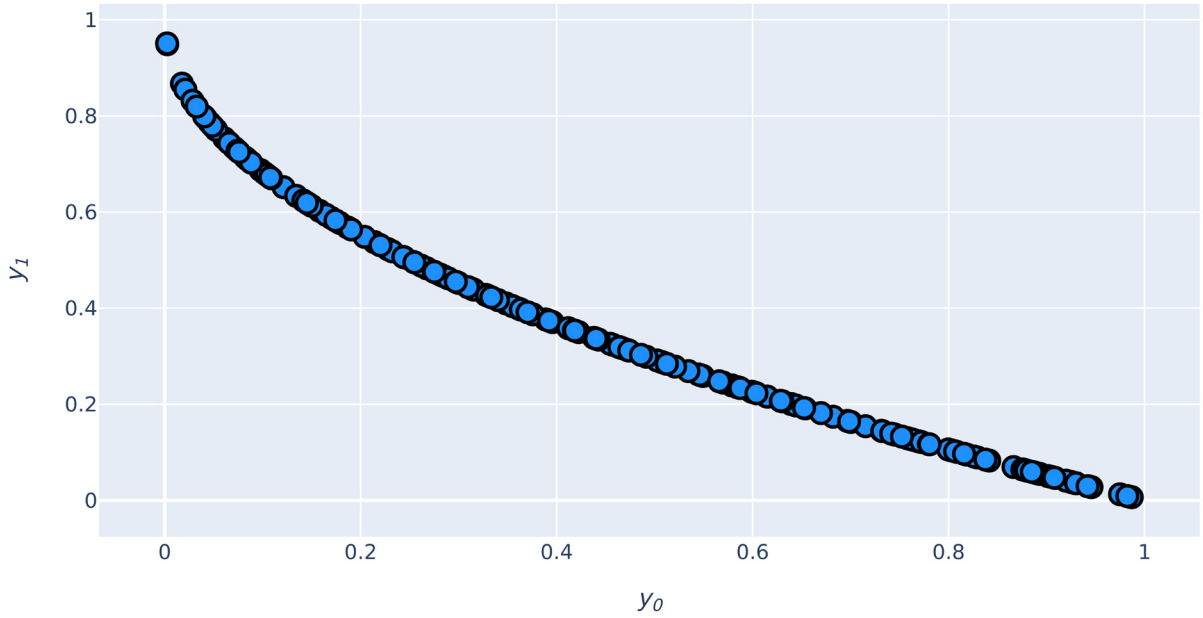

Figura B.14: Frontera de Pareto real para ZDT4

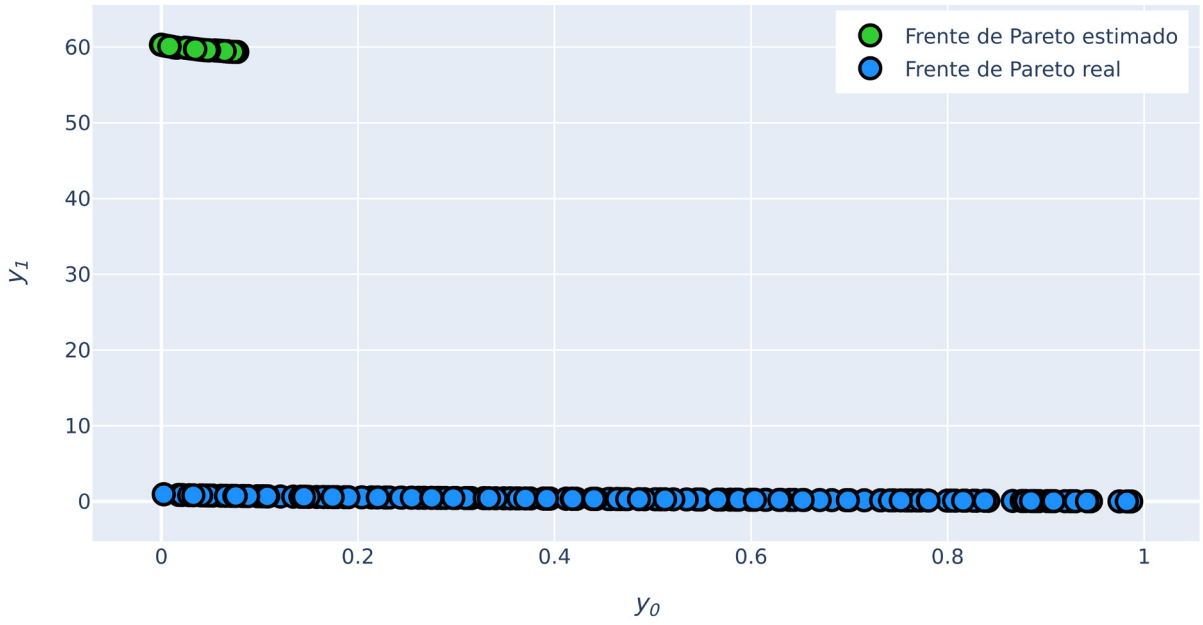

Figura B.15: Frontera de Pareto estimada por la metodología

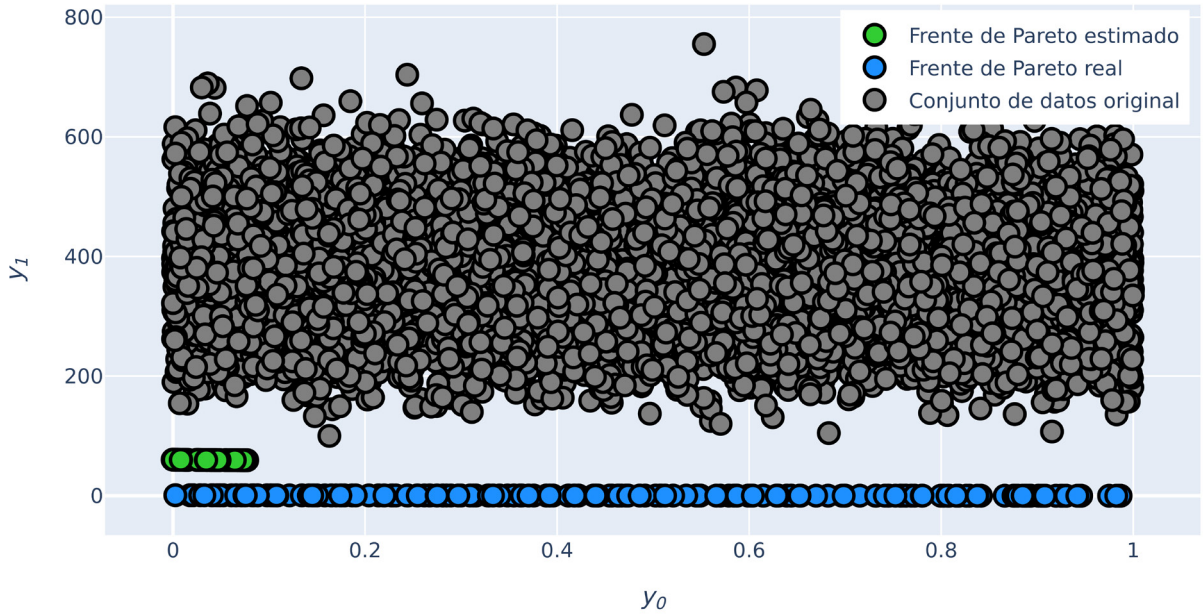

Figura B.16: Comparación de la solución encontrada con el espacio realizable

Figura B.17: Conjuntos de Pareto real y obtenido