



Universidad de Valladolid



ESCUELA DE INGENIERÍAS  
INDUSTRIALES

Máster en Ingeniería Industrial

**MÁSTER EN INGENIERÍA INDUSTRIAL**  
**ESCUELA DE INGENIERÍAS INDUSTRIALES**  
**UNIVERSIDAD DE VALLADOLID**

**TRABAJO FIN DE MÁSTER**

**DETECCIÓN TEMPRANA DE FALLOS EN UNA PLANTA  
DEPURADORA DE AGUA RESIDUALES (EDAR) MEDIANTE  
AUTOENCODERS VARIACIONALES**

Autor: Dña. Marta Arroyo Arévalo  
Tutores: D. Gregorio I. Sainz Palmero  
Dña. M<sup>a</sup> Jesús de la Fuente Aparicio

Valladolid, septiembre 2024

# RESUMEN

---

La creciente escasez de recursos hídricos, agravada por el incremento de la densidad poblacional y la mayor contaminación de las aguas, ha impulsado la necesidad de desarrollar sistemas más avanzados para la monitorización de las Estaciones Depuradoras de Aguas Residuales (EDAR).

Para ello, en este Trabajo Fin de Máster se propone una metodología innovadora que permita la detección temprana de fallos en procesos industriales basada en la aplicación de técnicas de *Deep learning* y aplicada al caso particular de una EDAR. Esta metodología emplea un tipo de redes neuronales para la extracción de las características principales del proceso conocido como autoencoder variacional (VAE). El empleo de un modelo tipo VAE permite construir un sistema capaz de detectar cualquier tipo de fallo con rapidez, sin necesidad de categorizar los fallos con anterioridad y sin un conocimiento experto del dominio. Se utiliza el modelo de VAE mínimo con el fin evaluar su capacidad en la detección temprana de fallos empleando técnicas de control estadístico de procesos multivariados.

Además, dada la complejidad y la dimensionalidad de una EDAR, en este Trabajo se estudia la aplicación de la metodología propuesta en sistemas de control centralizados y distribuidos para realizar un estudio comparativo de su eficacia y velocidad de detección. Las técnicas de monitorización distribuidas descomponen la planta en bloques de variables sobre los que se generan los distintos modelos de VAE para la posterior detección distribuida de fallos.

Así, con la segmentación de la planta no solo se persigue reducir la complejidad del proceso, sino también reducir la carga computacional del sistema mejorando su rendimiento, el coste económico y la huella de carbono derivada del consumo energético que supone tanto la diagnosis como el almacenamiento de información en la operación de la EDAR.

Palabras clave: Detección de Fallos, Deep Learning, Autoencoder Variacional, Monitorización Centralizada/distribuida, mRMR, EDAR.

# ABSTRACT

---

The increasing scarcity of water resources, aggravated by the increase in population density and water pollution, has driven the need to develop more advanced monitoring systems for Wastewater Treatment Plants (WWTP).

To this end, this Master's dissertation proposes an innovative methodology for the early detection of failures in industrial processes based on the application of deep learning techniques and applied to the particular case of a WWTP. This methodology uses a type of neural networks for the extraction of the main features of the process known as variational autoencoder (VAE). The minimum VAE model is used in order to evaluate its capability in early fault detection using multivariate statistical process control techniques. The use of a VAE type model allows the construction of a system capable of detecting any type of failure quickly, without the need to categorise the failures beforehand and without expert knowledge of the domain.

In addition, given the complexity and dimensionality of a WWTP, this work analyses the application of the proposed methodology in centralised and distributed control systems in order to carry out a comparative study of its efficiency and detection speed. Distributed monitoring techniques decompose the plant into blocks of variables on which the different VAE models are generated for the subsequent distributed detection of faults.

Thus, the segmentation of the plant not only aims to reduce the complexity of the process, but also to reduce the computational load of the system by improving its performance, the associated energy cost and, consequently, improving the carbon footprint of the WWTP operation.

**Key words:** Fault detection, Deep Learning, Variational Autoencoder, Centralized/Distributed Monitoring, mRMR, WWTP

# ÍNDICE DE CONTENIDOS

---

<b>1. INTRODUCCIÓN.....</b>	<b>1</b>
1.1 Control del tratamiento de aguas residuales .....	1
1.2 Motivación del proyecto .....	3
1.3 Objetivo y alcance .....	4
1.4 Estructura de la memoria.....	4
<b>2. ESTADO DEL ARTE.....</b>	<b>6</b>
2.1 Control de procesos.....	6
2.2 Deep learning para la detección de fallos.....	18
<b>3. DETECCIÓN DE FALLOS CENTRALIZADA VS DISTRIBUIDA MEDIANTE VAE</b>	<b>27</b>
3.1 Planteamiento general.....	27
3.2 Detección centralizada de fallos .....	30
3.3 Detección distribuida de fallos .....	37
<b>4. EXPERIMENTACIÓN Y RESULTADOS.....</b>	<b>41</b>
4.1 Caso de estudio.....	41
4.2 Metodología experimental .....	43
4.3 Análisis de los resultados .....	45
4.4 Comparativa de resultados.....	75
<b>5. ESTUDIO ECONÓMICO .....</b>	<b>80</b>
5.1 Fases del proyecto .....	80
5.2 Planificación .....	82
5.3 Análisis económico.....	84
<b>6. CONCLUSIONES Y LÍNEAS FUTURAS .....</b>	<b>88</b>
6.1 Conclusiones .....	88
6.2 Líneas futuras.....	89
<b>7. BIBLIOGRAFÍA.....</b>	<b>91</b>

# LISTA DE FIGURAS

---

Figura 1. Estación Depuradora de Aguas Residuales de Valladolid. Fuente: Aquavall .....	1
Figura 2. Ejemplo de gráfico control de proceso industrial [19] .....	6
Figura 3. Métodos basados en el conocimiento: jerarquía de abstracción [25]; diágrafos [26]; árboles de fallas [27]; sistemas difusos .....	8
Figura 4. Análisis de componentes principales (PCA): a) Distribución original tridimensional de los datos de entrada y componentes principales correspondientes; b) Proyección bidimensional de los datos en los ejes PC <sub>1</sub> y PC <sub>2</sub> [31] .....	10
Figura 5. Comparativa componentes principales de PCA (a) y componentes independientes de ICA (b) para una distribución de datos bidimensional no gaussiana [35]. .....	11
Figura 6. Arquitectura redes neuronales .....	12
Figura 7. Esquema ilustrativo del método descentralizado de monitorización de procesos .....	14
Figura 8. Esquema ilustrativo de selección de características mediante mRMR.....	16
Figura 9. Clasificación de las técnicas de DL según la naturaleza de los datos.....	18
Figura 10. Clasificación de las técnicas de DL según la naturaleza de los datos.....	19
Figura 11. Arquitectura de un autoencoder .....	22
Figura 12. Comparativa representación espacio latente AE y VAE .....	23
Figura 13. Esquema estructura del VAE con una capa .....	24
Figura 14. Función de pérdida VAE: error de reconstrucción y divergencia KL .....	26
Figura 15. Esquema alto nivel metodología sistema detección temprana de fallos propuesto.....	28
Figura 16. Diagrama del proceso <i>offline</i> , método centralizado.....	30
Figura 17. Diagrama del proceso <i>online</i> , método centralizado.....	34
Figura 18. Diagrama del proceso <i>offline</i> , método distribuido.....	38
Figura 19. Diagrama del proceso <i>online</i> , método distribuido.....	40
Figura 20. Diagrama del proceso BSM2 [72], [73].....	42
Figura 21. Validación entrenamiento VAE centralizado, 300 días. Comparativa entrada y salida VAE con datos entrenamiento (fichero sin fallos) .....	46
Figura 22. Validación entrenamiento VAE centralizado, 3 días. Comparativa entrada y salida VAE con datos entrenamiento (fichero sin fallos).....	47
Figura 23. Validación entrenamiento VAEs distribuido, 30 días. Comparativa entrada y salida VAEs con datos entrenamiento (fichero sin fallos) .....	59
Figura 24. Validación entrenamiento VAEs distribuido, 3 días. Comparativa entrada y salida VAEs con datos entrenamiento (fichero sin fallos).....	61

Figura 25. Diagrama de Gantt ..... 83

# LISTA DE TABLAS

---

Tabla 1. Funciones de activación utilizadas en redes neuronales.....	20
Tabla 2. Tiempo y amplitud simulación fallos .....	43
Tabla 3. Hiperparámetros y parámetros en el diseño del VAE .....	44
Tabla 4. Parametrización modelo VAE centralizado RMSE test mínimo, 300 días ..	45
Tabla 5. Parametrización modelo VAE centralizado RMSE test mínimo, 3 días .....	47
Tabla 6. Determinación umbrales de control estadístico método centralizado .....	48
Tabla 7. Grupos de variables resultantes de la técnica mRMR. Proceso offline .....	57
Tabla 8. Parametrización modelo VAE descentralizado RMSE test mínimo, 300 días .....	58
Tabla 9. Parametrización modelo VAE descentralizado RMSE test mínimo, 3 días	60
Tabla 10. Determinación umbrales de control estadístico método distribuido .....	62
Tabla 11. Comparativa tiempo de ejecución proceso <i>offline</i> .....	75
Tabla 12. Comparativa resultados detección de fallo. Fallos detectados .....	77
Tabla 13. Comparativa resultados detección de fallo. Fallos no detectados .....	78
Tabla 14. Análisis resultados detección de fallo en proceso online. Método distribuido .....	78
Tabla 15. Enumeración tareas fases desarrollo del proyecto .....	82
Tabla 16. Cálculo coste empleado para empresa (jornada completa).....	84
Tabla 17. Cálculo jornadas efectivas al año de un trabajador .....	85
Tabla 18. Cálculo coste mano de obra por fase de proyecto.....	85
Tabla 19. Cálculo coste amortización equipos ofimáticos .....	85
Tabla 20. Cálculo coste amortización equipos por fase de proyecto .....	86
Tabla 21. Cálculo costes indirectos .....	86
Tabla 22. Cálculo coste total del proyecto .....	87
Tabla 23. Precio de venta final .....	87

# LISTA DE GRÁFICOS

---

Gráfico 1. Umbral superior control TCL2 método centralizado.....	48
Gráfico 2. Control centralizado. Fallo 1 .....	50
Gráfico 3. Control centralizado. Fallo 2 .....	51
Gráfico 4. Control centralizado. Fallo 3 .....	52
Gráfico 5. Control centralizado. Fallo 4 .....	53
Gráfico 6. Control centralizado. Fallo 5 .....	54
Gráfico 7. Control centralizado. Fallo 6 .....	54
Gráfico 8. Control centralizado. Fallo 7 .....	55
Gráfico 9. Control centralizado. Fallo 8 .....	56
Gráfico 10. Umbral superior control TCL2 método distribuido, grupo 1 .....	63
Gráfico 11. Umbral superior control TCL2 método distribuido, grupo 2 .....	63
Gráfico 12. Umbral superior control TCL2 método distribuido, grupo 3 .....	64
Gráfico 13. Umbral superior control TCL2 método distribuido, grupo 4 .....	64
Gráfico 14. Umbral superior control TCL2 método distribuido, grupo 5 .....	65
Gráfico 15. Umbral superior control TCL2 método distribuido, grupo 6 .....	65
Gráfico 16. Umbral superior control TCL2 método distribuido, grupo 7 .....	66
Gráfico 17. Umbral superior control TCL2 método distribuido, grupo 10.....	66
Gráfico 18. Control distribuido. Fallo 1 .....	68
Gráfico 19. Control distribuido. Fallo 2 .....	69
Gráfico 20. Control distribuido. Fallo 3 .....	70
Gráfico 21. Control distribuido. Fallo 4 .....	71
Gráfico 22. Control distribuido. Fallo 5 .....	72
Gráfico 23. Control distribuido. Fallo 6 .....	73
Gráfico 24. Control distribuido. Fallo 7 .....	74
Gráfico 25. Control distribuido. Fallo 8 .....	75





# 1. INTRODUCCIÓN

---

## 1.1 CONTROL DEL TRATAMIENTO DE AGUAS RESIDUALES

El abastecimiento de agua potable para el consumo humano constituye uno de los principales desafíos a los que se enfrenta la sociedad y que se está viendo agravado por los cambios hidrológicos inducidos por el cambio climático [1].

El agua es un recurso natural escaso en su estado de potabilidad. Los fenómenos climáticos extremos, como las inundaciones o las sequías, añadido a la pérdida de biodiversidad y la utilización insostenible de los recursos, tienen una repercusión adversa directa en el medio hídrico, en términos de calidad y cantidad. El consecuente aumento de la escasez de agua puede llegar a afectar a la salud humana y al desarrollo de las ciudades [2].

Además, su gestión sostenible es cada vez más necesaria debido al rápido aumento de la población, que no solo conduce a una extracción excesiva del recurso, sino también a la mayor contaminación de los ríos, lagos y aguas subterráneas por residuos municipales, industriales y agrícolas [2].

Esta preocupación por el recurso hídrico fue incorporada como uno de Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030 sobre el Desarrollo Sostenible de la ONU: “Garantizar la disponibilidad de agua y su gestión sostenible y el saneamiento para todos” (ODS nº6).

Es por ello por lo que el papel de las estaciones depuradoras de aguas residuales (EDAR) en el tratamiento de éstas, junto al agua procedente de los sistemas de aguas pluviales y la escorrentía de las tierras urbanas y agrícolas, resulta fundamental para la gestión ambiental del agua.

En una EDAR el agua se somete a diferentes procesos físicos, químicos y biológicos para reducir su contenido en elementos contaminantes hasta límites aceptables [3] [4] antes de verterla a los cauces o derivarse para su uso en otras aplicaciones como el riego, servicios de limpieza u otros.



Figura 1. Estación Depuradora de Aguas Residuales de Valladolid. Fuente: Aquavall

El entorno político en Europa se ha vuelto especialmente favorable al impulso de la inversión en la reutilización de aguas residuales municipales. Según el analista de Bluefield Research, Zineb Moumen, “se espera que el gasto en reutilización de aguas municipales en Europa crezca de 140 millones de euros en 2023 a aproximadamente 360 en 2030” [5].

Esta afirmación denota la necesidad de desarrollar nuevos sistemas avanzados de control de las EDAR que permitan maximizar la reutilización segura y sostenible del agua [6], garantizando la estabilidad del proceso, el cumplimiento de los límites normativos de calidad del efluente y, a su vez, reduciendo costes operativos y de mantenimiento derivados de una detección tardía de fallos [7], [8].

Tradicionalmente, los sistemas de control de procesos se basan en la construcción de una representación matemática del sistema, lo cual requiere de un alto grado de conocimiento del dominio. Estos modelos tradicionales no suelen ser capaces de capturar la variabilidad y la dinámica real de los procesos, haciendo ineficaz la predicción cuando se trata de relaciones entre variables complejas y no lineales. Este hecho se hace aún más evidente en instalaciones que incluyen diversas operaciones unitarias como es el caso de una EDAR.

Una EDAR se compone de múltiples procesos dinámicos, no lineales y sujetos a continuas perturbaciones ambientales externas [9]. Esta heterogeneidad y variabilidad de los datos supone que los métodos de control clásicos resulten ser sistemas de bajo rendimiento [8] o incluso inviables para la toma de decisiones [10].

Así, surge la oportunidad de desarrollar nuevas metodologías basadas en técnicas de Inteligencia Artificial (IA) que faciliten la evolución de los sistemas tradicionales hacia la diagnosis inteligente de industrias complejas.

La IA se basa en la idea de crear sistemas que emulen la inteligencia humana, lo que incluye el razonamiento lógico, la toma de decisiones y la resolución de problemas. A diferencia de las técnicas convencionales, los algoritmos basados en IA como el aprendizaje automático o *Machine Learning* (ML), son capaces de aprender patrones entre datos no estructurados y sin la necesidad de ser programados explícitamente, lo cual es de especial interés en procesos complejos y no lineales [11].

Es precisamente esta capacidad de identificar patrones y su adaptabilidad a la dinámica real del proceso lo que hace a las técnicas de ML relevantes en la detección temprana de fallos en los sistemas industriales [12].

En la última década, entre los métodos de ML se ha abierto paso el desarrollo de técnicas de aprendizaje profundo o *Deep Learning* (DL), compuestos por un conjunto de neuronas que forman una red con múltiples capas [8], [13]. Las redes neuronales profundas son altamente efectivas para procesar grandes volúmenes de datos, aprender patrones e identificar relaciones no lineales.

En este contexto, se presenta este Trabajo Fin de Máster donde se estudia la aplicación un tipo particular de redes neuronales, el autoencoder variacional (VAE), para la detección de fallos en el proceso de una EDAR.

Un VAE es un modelo generativo basado en datos de aprendizaje no supervisado que se ha utilizado en numerosos estudios para el desarrollo de técnicas avanzadas de detección de anomalías por su potencial para la modelización y extracción de características de conjuntos complejos no lineales y no gaussianos como una EDAR

[65]. La aplicación de VAE en EDAR ya ha sido demostrada con resultados satisfactorios por algunos autores [71], [62]. En base a ello, este trabajo aúna el éxito de su aplicación en EDAR con su potencial en la detección de anomalías con el objetivo de desarrollar una metodología basada en VAE aplicada a una EDAR que permita una eficaz y temprana detección de fallos.

En particular, se estudiará el modelo de VAE con la arquitectura más sencilla para evaluar si la estructura mínima es suficiente para la detección de fallos. Pese a su limitación en cuanto al aprendizaje de relaciones complejas entre variables, con el modelo de VAE mínimo se lograrían importantes ahorros no solo en tiempo de ejecución, sino también económicos derivados del menor tamaño de base de datos, menor consumo energético y, por tanto, menor impacto ambiental.

Otra de las problemáticas emergentes en los procesos industriales complejos que se abordará en este Trabajo es el aumento del volumen de datos. El incremento en la dimensionalidad de los conjuntos de datos ha provocado que algunas de las técnicas de DL se vuelvan ineficientes [49]. Por ello, se estudiarán las ventajas que ofrece la aplicación del sistema propuesto en métodos de control de procesos distribuidos frente a los clásicos métodos centralizados.

Los procesos de control distribuidos descomponen el proceso en múltiples grupos de variables sobre los que posteriormente se aplican los modelos de datos estadísticos para la detección de fallas [37]. Al reducir el volumen de datos de cada agrupación se prevé que mejore el rendimiento y eficiencia de las técnicas de DL que se apliquen.

## 1.2 MOTIVACIÓN DEL PROYECTO

Este desarrollo de este trabajo se ha realizado en el marco del proyecto del proyecto “Metodologías de Supervisión Inteligente y Distribuidas Aplicadas a Sistemas Integrados de Aguas”, (PID219-105434RB-C32) financiado por el Ministerio de Ciencia, Innovación y Universidades, y desarrollado en el Departamento de Ingeniería de Sistemas y Automática de la Escuela de Ingenierías Industriales de la Universidad de Valladolid.

El proyecto se engloba dentro del Programa Estatal de I+D+i Orientada a los Retos de la Sociedad [14], en el marco del Plan Estatal de Investigación Científica y Técnica y de Innovación 2017-2020, de la Agencia Estatal de Investigación del Ministerio de Ciencia, Innovación y Universidades.

Este Programa Estatal tiene como objetivo “contribuir a definir y consolidar las bases del liderazgo científico y tecnológico de España en ámbitos en los que la generación de conocimiento, la convergencia de paradigmas y tecnologías, y sus aplicaciones e innovaciones constituyen factores claves de la prosperidad y calidad de vida de nuestra sociedad” [15]. Así, este Programa contribuye a fomentar las actividades de I+D+i con el fin de incrementar la competitividad y liderazgo internacional de la ciencia y la tecnología españolas.



### 1.3 OBJETIVO Y ALCANCE

El objetivo principal de este trabajo es desarrollar un sistema de detección temprana de fallos para una EDAR y comparar su eficacia y rendimiento y las ventajas que ofrece según si se aplica en métodos de control centralizados o distribuidos. A diferencia del control centralizado de procesos, los métodos distribuidos segregan las variables de control en grupos de menor dimensión, siendo múltiples las entradas de estudio del sistema.

De este modo, se persigue mejorar las técnicas tradicionales de control de procesos que se muestran ineficientes en el entorno industrial moderno, cada vez más complejo y dinámico. En particular, este trabajo contribuye en el desarrollo de nuevas técnicas de gestión eficiente del agua en EDAR basadas en IA en línea del ODS nº6: “Garantizar la disponibilidad de agua y su gestión sostenible y el saneamiento para todos”.

Para ello, se propone emplear herramientas de aprendizaje profundo no supervisado y, en particular, una arquitectura de VAE. Con ello se busca desarrollar una metodología basado de redes neuronales que, a partir de datos históricos de operación de la planta bajo control, sea capaz de aprender patrones de funcionamiento del proceso y extraer las características principales del mismo sin la necesidad de un conocimiento previo del dominio.

Como punto de partida para la aplicación de VAE en sistemas de detección de fallos de EDAR se estudia el modelo de VAE mínimo, es decir, con una sola capa oculta de neuronas. Así, se tiene por objeto estudiar la eficacia y velocidad en la detección de fallos utilizando el modelo más básico de VAE tratando de establecer la línea base de la metodología propuesta en cuanto a complejidad del VAE se refiere.

Finalmente, para alcanzar los objetivos anteriores se utilizará el control estadístico de procesos multivariados para identificar de manera temprana cualquier desviación del funcionamiento de la EDAR. Una detección temprana permite la rápida implementación de medidas correctivas sobre el proceso objeto de estudio.

Es importante señalar que el alcance de este trabajo se limita a evaluar la efectividad del modelo en la detección temprana de fallos, excluyendo la identificación y diagnóstico de los mismos del alcance.

### 1.4 ESTRUCTURA DE LA MEMORIA

Para alcanzar el objetivo anterior, se ha estructurado esta memoria en 6 capítulos. En este primer capítulo se contextualiza el trabajo mediante una introducción a la necesidad del desarrollo de nuevos modelos de control y diagnóstico de EDAR y se expone la motivación y los objetivos que se persiguen.

El Capítulo 2 proporciona el marco teórico relativo a los sistemas de detección de fallos, el control estadístico de procesos multivariados y la monitorización distribuida de procesos. Adicionalmente, se realiza una descripción del concepto de DL y el funcionamiento de las redes neuronales para concluir con el fundamento teórico de los VAE.

El Capítulo 3 describe la metodología propuesta para la concepción y desarrollo del sistema de detección temprana de fallos mediante la aplicación del modelo VAE mínimo y técnicas de control estadístico de procesos. En primer lugar, se presenta en este tercer capítulo el planteamiento general de la metodología desarrollada y, a continuación, se describen las distintas etapas del proceso tanto para el método de control centralizado como distribuido.

En el Capítulo 4 se presenta el caso de EDAR objeto de estudio que se emplea para la experimentación de la metodología propuesta y se recogen los resultados de dicha experimentación. El objetivo será analizar la eficacia y velocidad de detección de fallos del sistema aplicado sobre el método centralizado y el distribuido. Adicionalmente, se realiza la comparativa entre los dos métodos propuestos con objeto de analizar qué ventajas ofrece cada uno de ellos.

El Capítulo 5 constituye una evaluación económica del proyecto que incluye la cuantificación del coste de los recursos humanos y materiales empleados en su desarrollo.

Por último, en el Capítulo 6 se exponen las conclusiones obtenidas en este trabajo y las contribuciones principales del mismo, analizando el alcance de los objetivos propuestos. Además, se proponen posibles líneas de trabajo que permitirían continuar con la mejora y optimización del proyecto.

## 2. ESTADO DEL ARTE

### 2.1 CONTROL DE PROCESOS

#### 2.1.1. Sistemas de detección de fallos

Las técnicas de monitorización de procesos y detección de fallos son fundamentales para mantener la estabilidad en los procesos industriales [16], [17]. La detección temprana y el diagnóstico de fallos en tiempo real pueden ayudar a evitar la propagación del evento.

Además, en la industria de procesos las expectativas en cuanto a la eficiencia de la producción, la calidad del producto, los niveles de seguridad y la protección del medio ambiente son cada vez más exigentes [18], lo que impulsa a introducir sistemas de control automático que monitoreen un mayor número de mediciones de variables del proceso.

Esta elevada cantidad de información a procesar hace que dichos sistemas de control sean cada vez más complejos, lo que implica la necesidad de desarrollar técnicas más avanzadas que puedan manejar un alto número de variables y detectar posibles anomalías en el proceso con rapidez.

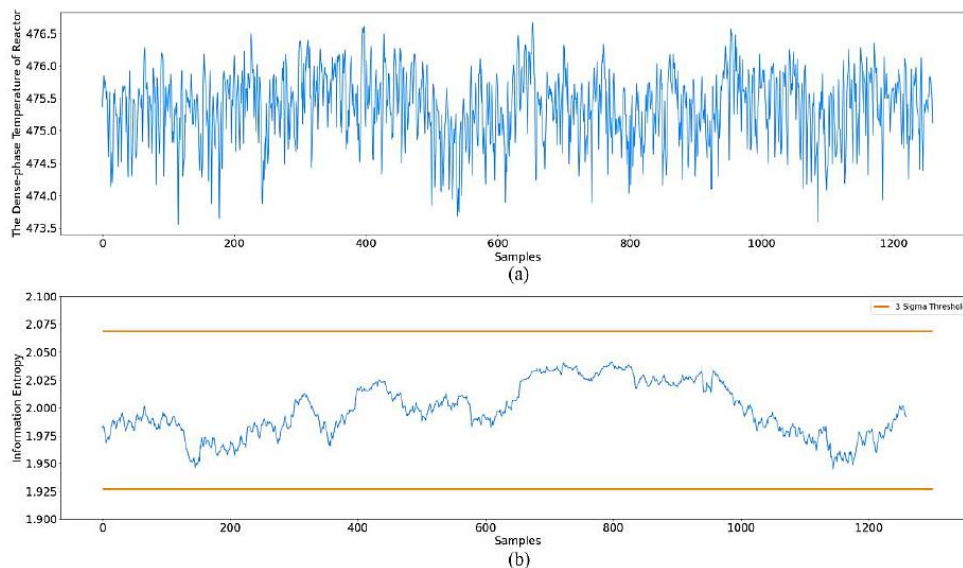


Figura 2. Ejemplo de gráfico control de proceso industrial [19]

En condiciones normales de funcionamiento, estas variables están estrechamente relacionadas debido a principios físicos y químicos. Estas relaciones pueden ser modeladas y el modelo obtenido utilizarse para analizar nuevos datos del proceso con el fin de detectar anomalías.

Dado que el modelo se construye a partir de las características de la planta, es importante elegir la técnica de modelado adecuada. Existe una gran cantidad de literatura sobre el control y diagnóstico de fallos en procesos que incluyen métodos analíticos, enfoques estadísticos y, más recientemente, la aplicación de modelos basados en IA [20].

Desde el punto de vista de la modelización, las técnicas de detección de fallos utilizadas tradicionalmente son métodos basados en modelos de proceso precisos y métodos basados en conocimientos o cualitativos. Actualmente, son los métodos basados en datos los que atraen cada vez más atención en la investigación y en los procesos industriales debido, principalmente, a su facilidad de implementación y la mínima necesidad de conocimiento experto previo [21] ya que se basan solo en datos históricos del proceso [20].

Así, es posible clasificar los métodos de detección de fallos en tres categorías: método basado en modelos, método basado en conocimientos y método basado en datos.

#### 2.1.1.1 Métodos basados en modelos de procesos

Los métodos en modelos, también conocidos como métodos de conocimiento profundo o causal, se desarrollan sobre el conocimiento de la física del proceso y se expresa en términos de funciones matemáticas entre las entradas y salidas del sistema. Los balances de masa, energía y momento del proceso, así como las ecuaciones de estado, se utilizan para desarrollar las ecuaciones que definen el modelo [22].

Si bien en el pasado los métodos de conocimiento profundo no eran frecuentes en el control de procesos y el diagnóstico de fallos, los avances tecnológicos y la potencia computacional han facilitado el desarrollo de estos modelos ya que suelen ser modelos matemáticos complejos que requieren un alto nivel de recursos computacionales.

Entre sus ventajas destaca la facilidad de interpretación y análisis de los fallos al estar basados en el modelo físico de la planta. Además, al no formularse en base al histórico del sistema, son modelos útiles cuando hay pocos datos disponibles o cuando las condiciones de operación se modifican e, incluso, pueden generalizarse y aplicarse a diferentes sistemas si el modelo matemático subyacente es válido.

Sin embargo, el tipo de modelos que pueden manejar los enfoques analíticos se limita a modelos lineales y, en algunos casos, a modelos no lineales muy específicos. En procesos que, por lo general, se caracterizan por la no linealidad de sus variables, como la mayoría de los procesos de la industria química, las aproximaciones lineales pueden resultar pobres y poco precisas, lo que reduce considerablemente la efectividad del sistema [22].

Adicionalmente, modelos demasiado complejos pueden ser costosos y difíciles de interpretar y mantener. A ello hay que añadir que para su definición es necesario disponer de un conocimiento experto en modelado matemático y en el dominio específico del sistema que se está diagnosticando.

Así, si bien los métodos basados en modelos cuantitativos pueden ser útiles y fácilmente interpretables en procesos lineales, su aplicación efectiva y rentable no puede asegurarse en entornos demasiado complejos o no lineales.



### 2.1.1.2 Métodos basados en el conocimiento

Los métodos basados en el conocimiento o métodos cualitativos se fundamentan en la información existente acerca del comportamiento del proceso y en la experiencia acumulada por los trabajadores de la planta [23]. La relación entre las entradas y salidas del sistema se expresan en términos de funciones cualitativas que se centran alrededor de diferentes unidades en el proceso.

La experiencia relacionada con anomalías en el proceso se utiliza para establecer un conjunto de reglas if-/then-/else que relacionen un conjunto de observaciones con causas específicas. Mediante un motor de inferencia se realiza el procesamiento lógico para derivar conclusiones o tomar decisiones basadas en dicho conocimiento previo [24].

Los modelos cualitativos generalmente pueden clasificarse en jerarquía de abstracción, árboles de fallas, diágrafos y sistemas difusos [20].

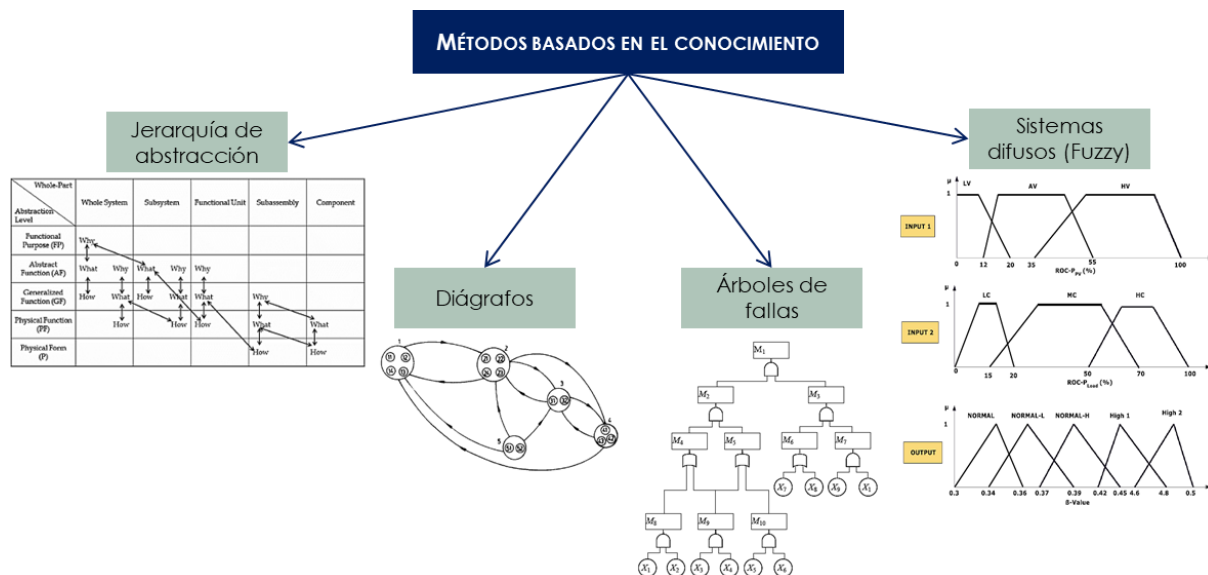


Figura 3. Métodos basados en el conocimiento: jerarquía de abstracción [25]; diágrafos [26]; árboles de fallas [27]; sistemas difusos

El problema más significativo de este tipo de métodos es que no tiene en cuenta el fundamento físico subyacente del sistema y, en consecuencia, falla en situaciones altamente complejas o inesperadas que no habían sido consideradas en la codificación [24], requiriendo una actualización constante del modelo.

Además, es importante señalar que su efectividad depende de la calidad del modelo y disponibilidad de expertos en el dominio para capturar y formalizar el conocimiento relevante, lo que limita enormemente su aplicación y su capacidad de predicción.

### 2.1.1.3 Métodos basados en datos de señales del proceso

Los métodos basados en datos, a diferencia de los anteriores, no requieren un conocimiento del modelo del proceso y el conocimiento experto asociado [20]. Estos métodos se construyen sobre la minería de datos, es decir, sobre el histórico de mediciones de las señales del proceso [8].

Estos enfoques basados en datos tienen por objeto generar un modelo del proceso que relacione matemáticamente las mediciones de entradas y salidas del sistema. Posteriormente, se utiliza este modelo en el proceso real para generar residuos y aplicar técnicas de detección de fallos [20].

La aplicación de estos métodos se ha vuelto cada vez de mayor en el dominio de la investigación y, especialmente, en procesos industriales complejos y/o de gran escala donde el modelado matemático o conocimiento experto previo son difíciles de obtener y construir interés [8], [23], [28], [29].

Dentro de los principales métodos basados en datos de diagnóstico de procesos se incluyen principalmente técnicas estadísticas multivariantes y métodos de aprendizaje profundo basados en inteligencia artificial, siendo algunos de los más relevantes el análisis de componentes principales (PCA), los mínimos cuadrados parciales (PLS), el análisis de componentes independientes (ICA), la red neuronal artificial (ANN) y algunos métodos mejorados basados en estos métodos básicos [23], [29].

La elección de la técnica dependerá de diversos factores, como la naturaleza de los datos, el objetivo del contexto de su aplicación y las limitaciones computacionales.

### **A. Análisis de componentes principales (PCA)**

Debido a su fácil aplicación y su eficiencia en el procesamiento de datos de alta dimensión [29], PCA es uno de los algoritmos más utilizados para el monitoreo de procesos [18], [28], además de aplicarse en otras áreas como la compresión de datos, el procesamiento de imágenes, la extracción de características o el reconocimiento de patrones [29].

La idea principal de PCA es proyectar el conjunto de datos de entrada correlacionado en un conjunto de menor dimensión de variables no correlacionadas, denominadas componentes principales (PC), manteniendo la mayor cantidad posible de información de las características originales.

Para ello, mediante el PCA se buscan los PC, es decir, los vectores propios de la matriz de covarianza de los datos de entrada, y se ordenan de mayor a menor en términos de la variabilidad que capturan en los datos originales. Una vez identificados los PC, se reduce la dimensionalidad del conjunto de datos de entrada proyectándolos sobre las direcciones que capturan la mayor varianza de los datos [18], [28], [30].

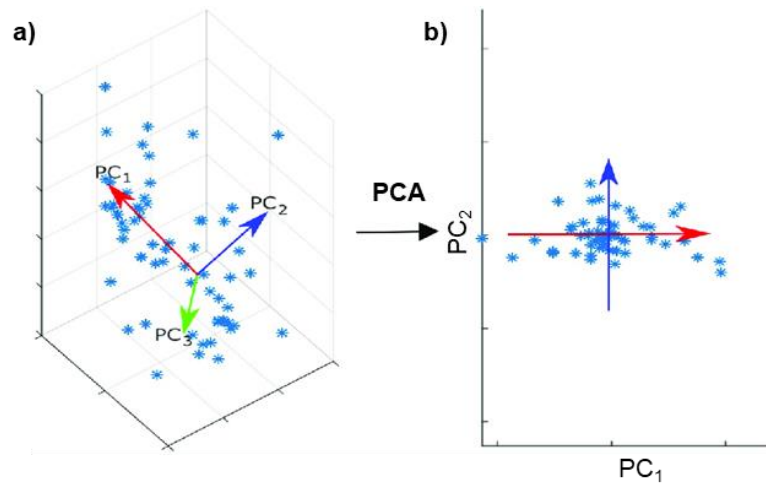


Figura 4. Análisis de componentes principales (PCA): a) Distribución original tridimensional de los datos de entrada y componentes principales correspondientes; b) Proyección bidimensional de los datos en los ejes  $PC_1$  y  $PC_2$  [31]

Según se afirma en [30], los gráficos de control basados en PCA mejoran la capacidad de detección temprana de fallos, así como los cambios en la estructura de covarianza propuestos.

Sin embargo, la principal limitación de esta técnica es que asume que las relaciones entre las características en el conjunto de datos son lineales y que las variables del espacio latente, de menor dimensión, siguen una distribución normal. Esto limita la efectividad de su aplicación, principalmente en procesos industriales complejos con características no lineales o cuando las variables latentes siguen distribuciones diferentes a la gaussiana.

## **B. Mínimos cuadrados parciales (PLS)**

Además de PCA, PLS es otro de los métodos estadísticos basados en datos ampliamente utilizado para la construcción de modelos, la detección de fallas y el diagnóstico [29]. También es posible su aplicación para la reducción de dimensionalidad de los datos del proceso [32].

PLS tiene como objetivo encontrar relaciones lineales entre las variables predictoras (entrada) y las variables de respuesta (salida). Mediante PLS se busca transformar linealmente la matriz predictora y la de respuesta en sus respectivos conjuntos de variables latentes que reflejen las mayores variaciones en los datos de ambas matrices, al mismo tiempo que se maximiza la correlación cruzada entre las variables predictoras y las de respuesta [32].

Este método también ha sido empleado en la supervisión de procesos [32]. Sin embargo, al igual que PCA, se trata de método de modelado lineal, por lo que no sería adecuado para describir características no lineales significativas propias de procesos industriales complejos [17], [33].

### C. Análisis de componentes independientes (ICA)

Más recientemente ha ganado popular otro método estadístico de monitoreo de procesos basado en datos: el análisis independiente de componentes (ICA) [18]. A diferencia del PCA, mediante el que se proyecta variables correlacionadas (relación lineal) en un nuevo conjunto de variables no correlacionadas, pero no independientes, el objetivo del ICA es descomponer los datos en combinaciones lineales de componentes estadísticamente independientes (CI) [34].

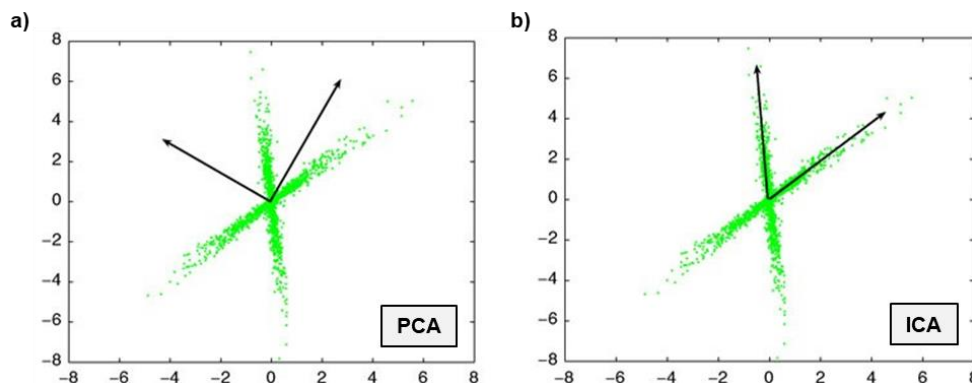


Figura 5. Comparativa componentes principales de PCA (a) y componentes independientes de ICA (b) para una distribución de datos bidimensional no gaussiana [35].

Este método puede explotar los datos del proceso de forma más eficiente que el método PCA. No obstante, hay que tener en cuenta que el costo computacional y procedimiento de optimización de PCA es mucho más simple que el método ICA [18].

Sin embargo, los tres métodos basados en datos descritos (PCA, PLS e ICA) presenta la misma problemática y es la asunción de linealidad en los datos de entrada, cuestión que no será aplicable al caso de estudio del presente trabajo.

Para superar esta desventaja, se han desarrollado otros métodos no lineales basados en los clásicos anteriores aplicando métodos de kernel, como son como el kernel PCA (KPCA), kernel PLS (KPLS) o el kernel ICA (KICA) [30], [36], [37]. Uno de los inconvenientes de estos métodos es que el tiempo de cálculo puede aumentar con el número de muestras [38] convirtiéndolo en un proceso costoso computacionalmente.

### D. Redes neuronales

Frente a los modelos tradicionales de aprendizaje basados en datos, que precisan que la entrada cumpla ciertas condiciones estadísticas como independencia de las variables, simetría y normalidad en su distribución para su correcta aplicación, surgen las técnicas basadas en *Deep Learning* (DL) [8], [39].

Los modelos de aprendizaje profundo o DL se desarrollan principalmente en base a modelos de redes neuronales, es decir, modelos matemáticos inspirados en la estructura básica del cerebro humano [40]. Estos modelos extraen características o representaciones esenciales de los datos de entrada sin procesar organizando sus unidades de procesamiento computacional, llamadas neuronas, en una jerarquía de capas [9].

Así, las redes neuronales consisten en un conjunto de capas de neuronas interconectadas a través de las cuales se propagan los datos de entrada en un proceso conocido como propagación hacia adelante (*forward propagation*).

Cada neurona realiza una combinación lineal de las entradas ponderadas por pesos, aplica una función de activación no lineal y pasa su salida como entrada a las neuronas de la capa siguiente. Durante el entrenamiento, la red ajusta automáticamente los pesos minimizando una función de pérdida que evalúa la diferencia entre las predicciones de la red y la salida deseada, utilizando un algoritmo de optimización como el descenso del gradiente (*backpropagation*).

Una vez que la red neuronal está entrenada y validada, se puede utilizar para hacer predicciones sobre nuevos datos de entrada.

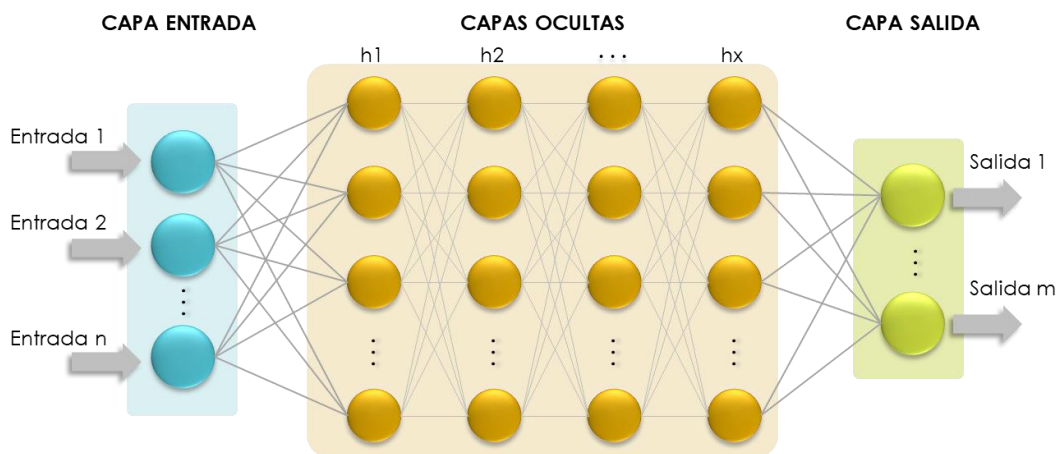


Figura 6. Arquitectura redes neuronales

Este proceso permite que las redes neuronales tiendan a producir una alta precisión de predicción [11] superando con éxito a los métodos estadísticos basados en datos tradicionales en la resolución de problemas no lineales [40] y complejos [41]. Así, las redes neuronales son adecuadas para aprender patrones de funcionamiento y aplicar el diagnóstico de fallos en sistemas dinámicos no lineales [20].

En particular, los métodos basados en datos que aplican técnicas de DL para el modelado de procesos son especialmente interesantes en los procesos de tratamiento de aguas debido a su simplicidad, precisión predictiva y mayor velocidad de cálculo [42], [43]. Son múltiples los estudios de investigación que muestran que la aplicación de redes neuronales en EDAR es favorable tanto en la simulación, como en el modelado y el control de este tipo de plantas [42].

## 2.1.2. Control estadístico de procesos multivariantes

En el ámbito de la monitorización de procesos industriales, el control estadístico de procesos ha demostrado ser una técnica eficaz [44]. Se basa en el uso de técnicas estadísticas para recopilar datos durante la operación del proceso, analizar estos datos y tomar decisiones basadas en evidencias estadísticas para mantener el proceso dentro de límites predefinidos en base al estado de operación normal [8].

Aunque el control estadístico de procesos univariante ha sido ampliamente utilizado, adolece de falta de aplicabilidad en entornos que manejan una gran cantidad de datos. Esto se debe a que, en la mayoría de las ocasiones, una variación no afecta solo a la variable donde se produce la falla, sino que se propaga por múltiples variables del proceso que están correlacionadas con la de origen [21], [44].

En consecuencia, como mejora al método tradicional surge el Control Estadístico de Procesos Multivariantes (MSPC). El MSPC considera conjuntamente todas las variables críticas ofreciendo una visión más completa y precisa del estado y comportamiento del proceso. En caso de producirse una situación fuera de control, ésta puede localizarse en una variable o por un cambio en la relación existente entre variables [45].

Una característica fundamental del MSPC es el uso de variables latentes que capturan la variabilidad conjunta de las variables observadas [44], [46]. Estas variables latentes son construidas a partir de las variables originales y pueden representar patrones subyacentes del comportamiento del proceso. Así, estos métodos se aplican sobre modelos construidos mediante métodos basados en datos [44] como los descritos en el apartado 2.1.1.3.

En este contexto, el gráfico de control estadístico  $T^2$  de Hotelling se utiliza típicamente para monitorear datos multivariados [18], [41], [44], [47].

### 2.1.2.1 Estadístico $T^2$ de Hotelling

El gráfico de control  $T^2$  de Hotelling es una técnica estadística fundamental para el monitoreo simultáneo de múltiples características correlacionadas. [16]. Proporciona una indicación de variabilidad inusual dentro del subespacio normal [18], es decir, de las variables latentes extraídas de las características principales del proceso.

El estadístico  $T^2$  se basa en la distancia de Mahalanobis [48] que cuantifica la distancia entre un punto  $x$  y un conjunto de puntos representados por un conjunto de datos multivariados considerando la correlación entre variables y sus varianzas. Esto lo hace útil para comparar la similitud entre observaciones en datos multivariados [36].

En esencia,  $T^2$  representa la distancia multivariante entre las muestras y la media del proceso. El cálculo de  $T^2$  de Hotelling se define según la Ec. 1 [23], [30]:

$$T_i^2 = (x_i - \bar{x})^T S^{-1} (x_i - \bar{x}) \quad \text{Ec. 1}$$

Donde:

$x_i$ , es el vector de medias de la muestra de estudio  $i$

$\bar{x}$ , es el vector de medias de la muestra bajo control

$S^{-1}$ , es la inversa de la matriz de covarianza de muestra bajo control

Bajo la suposición de que el vector de datos muestrales de  $p$  variables sigue una distribución gaussiana, el umbral de control de  $T^2$  puede aproximarse aplicando la distribución F de Fisher-Snedecor mediante la Ec. 2 [23], [30]:

$$T_{CL}^2 = \frac{p(n^2 - 1)}{n(n - p)} F_{(\alpha, p, n-p)} \quad \text{Ec. 2}$$

Donde:

$n$ , es el número de componentes de la entrada de datos

$p$ , es el número de variables latentes (PC en el caso de PCA)

$F_{(\alpha, p, n-p)}$ , es el valor de la distribución F con nivel de significación  $\alpha$ , y con  $p$  y  $(n - p)$  grados de libertad

La aplicación del cálculo de  $T^2$  para la diagnosis de procesos implica una primera fase de establecimiento del umbral de control a partir de datos históricos del proceso en operación normal y una segunda fase de seguimiento del proceso.

En esta segunda fase se asumirá que la matriz de covarianza permanece constante para el cálculo de los valores de  $T_i^2$  de la muestra de estudio. En el caso de que  $T_i^2$  exceda el umbral de control superior  $T_{CL}^2$ , se considera que se ha producido un fallo y el proceso está fuera de control.

### 2.1.3. Monitorización distribuida de procesos

El aumento en la dimensionalidad de los conjuntos de datos ha provocado que muchos algoritmos de aprendizaje automático se conviertan en métodos obsoletos al no poder lidiar con grandes tamaños de muestra de entrada de manera eficiente [49]. Por ello, es necesario adaptar dichos métodos a la realidad actual de los procesos.

Una de las alternativas que ha ganado gran atención son los métodos de monitoreo descentralizados [36], [37]. Los esquemas de monitoreo distribuido descomponen el proceso en múltiples grupos de variables sobre los que posteriormente se aplican los modelos de datos estadísticos para la detección de fallas para cada grupo por separado [37].

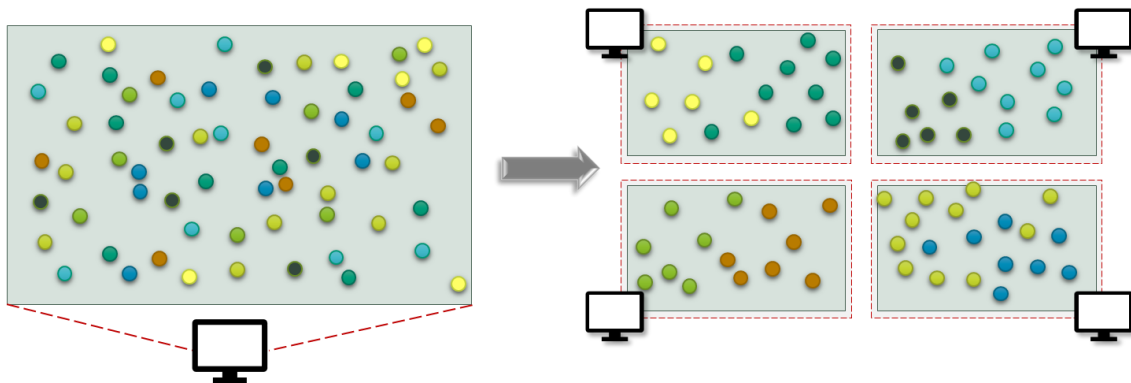


Figura 7. Esquema ilustrativo del método descentralizado de monitorización de procesos

Esta segmentación reduce la complejidad [16] y puede mejorar considerablemente el rendimiento de los procesos de aprendizaje posteriores al reducir la carga computacional [37], [49], [50]. Además, la construcción de modelos por grupos con sus correspondientes umbrales de detección también puede facilitar la localización y aislamiento de fallas [8], [37]. Por último, una diagnosis del proceso descentralizada a su vez contribuiría a la robustez de todo el proceso de control ya que las actualizaciones y ajustes periódicos en ciertos grupos no afectarían al resto [37].

En la práctica, la descomposición adecuada del proceso en grupos no es una tarea fácil. En los conjuntos de datos de alta dimensión es común encontrar un gran número de características que son irrelevantes, redundantes y ruidosas con respecto al concepto de clase [49], [50].

Para abordar esta cuestión, se presenta la estrategia de selección de características. La selección de características reduce la dimensión del conjunto a partir de datos históricos [37] eliminando características irrelevantes y redundantes [49]. Además, al contrario que otros métodos, en la reducción se mantienen las características originales lo que lo hace de especial interés en procesos donde la interpretación del modelo es fundamental para el control.

### 2.1.3.1 Selección de características/variables

La selección de características ha sido validada en estudios teóricos y empíricos como una estrategia efectiva para optimizar la eficacia del aprendizaje, mejorar la precisión en las predicciones y simplificar la estructura de los modelos resultantes [51], [52].

Por un lado, la selección de características se clasifica en función de si es supervisada o no supervisada [51]. Es decir, en función de si dispone previamente de las etiquetas de clase de todas las muestras y se utilizan como criterio para la evaluación de características (supervisado) o, por el contrario, la selección de características se basa en funciones de distribución o agrupaciones en clústeres al no disponer de etiquetas de clase (no supervisado) [50].

Por otro lado, los distintos métodos de selección de características se clasifican en modelos basados en filtros o modelos basados en envolturas o contenedores. Los métodos basados en filtros evalúan las características utilizando medidas de relevancia estadísticas o heurísticas [51], independientemente del proceso de aprendizaje [53].

Los métodos basados en contenedores, por su parte, evalúan subconjuntos de características entrenando un modelo con cada subconjunto y considerando el rendimiento del modelo como criterio para seleccionar un subconjunto óptimo de características [51]. Por ello, estos métodos son computacionalmente más costosos que los basados en filtros ya que requieren entrenar y validar un modelo para cada subconjunto de características.

Adicionalmente, se destaca que los enfoques basados en filtros generalmente muestran buenos aspectos de generalización, aunque pueden ser menos efectivos para reducir la dimensionalidad del conjunto de características [50].

Dentro de los métodos de selección de características basadas en filtros no supervisado destaca el algoritmo de máxima relevancia y mínima redundancia (mRMR) [49]. Como su propio nombre indica, la técnica mRMR se enfoca en



seleccionar las características que son más relevantes para la clasificación y minimizar la redundancia [50]. Según algunos autores, su popularidad se debe a su precisión a pesar de ser un método computacionalmente costoso: crece cuadráticamente con el número de características y linealmente con el tamaño de la muestra [49], [53].

### 2.1.3.2 Técnica mRMR

El método mRMR fue desarrollado por Peng et al. [54] y desde entonces ha adquirido gran relevancia en ML y la reducción de la dimensionalidad mediante la selección de características. Sus primeras aplicaciones se centraron en la selección de características de datos de expresión génica de microarrays. Actualmente, su uso se ha expandido a múltiples campos como preprocesamiento de datos en el análisis de texto o imágenes [49], entre otros.

El objetivo del algoritmo de mRMR es seleccionar el subconjunto de características que mayor correlación guarden con la clase (relevancia) penalizando la redundancia o correlación entre entidades [49], [53]. El tamaño del subconjunto de salida (K) es un valor predefinido por el usuario [50].

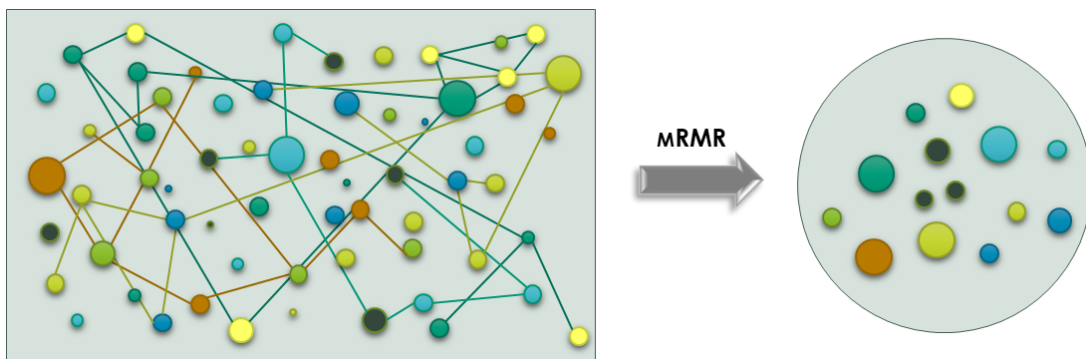


Figura 8. Esquema ilustrativo de selección de características mediante mRMR

Para variables de carácter continuo propias de procesos de control de plantas industriales, la relevancia se puede calcular mediante el estadístico F para evaluar la correlación de las características con la clase, mientras que la redundancia se puede determinar mediante el coeficiente de correlación de Pearson [53], [55].

mRMR funciona de forma iterativa, calculando para cada iteración la característica que es más relevante para la variable de salida y que está menos relacionada con cualquiera de las características del conjunto de datos. Una vez que se selecciona una característica, se elimina del conjunto de datos original y comienza la siguiente iteración hasta que el tamaño del conjunto de salida es igual a K [53].

Para la aplicación de mRMR sobre un conjunto de características se realiza en función de la información mutua  $I(x_1; x_2)$ , definida como la Ec. 3 [37], [55]:

$$I(x_1; x_2) = \iint p(x_1, x_2) \log \frac{p(x_1, x_2)}{p(x_1)p(x_2)} dx_1 dx_2 \quad \text{Ec. 3}$$

siendo  $p(x_1)$  y  $p(x_2)$  la densidad de probabilidad marginal de las características  $x_1$  y  $x_2$ , y  $p(x_1, x_2)$ , la densidad de probabilidad conjunta de ambas.

Dado un conjunto de características  $S = \{x_1, x_2, \dots, x_n\}$  y una clase de muestra  $c$ , la máxima relevancia entre  $S$  y  $c$  se calcula como la media de la información mutua entre cada característica  $x_i$  y clase  $c$ , según la Ec. 4 [37], [55]:

$$\max D(S, c) \rightarrow D = \frac{1}{|S|} \sum_{x_i \in S} I(x_i, c) \quad \text{Ec. 4}$$

siendo  $|S|$  el número de entidades en  $S$  e  $I(x_i, c)$  la información mutua entre la característica  $x_i$  y la clase  $c$ .

El criterio de redundancia mínima de todas las características en  $S$  se determina a partir de la Ec. 5, calculando la media de la información mutua entre la característica  $x_i$  y la característica  $x_j$  [37], [55]:

$$\min R(S) \rightarrow R = \frac{1}{|S|^2} \sum_{x_i, x_j \in S} I(x_i, x_j) \quad \text{Ec. 5}$$

La combinación y optimización de los criterios de máxima relevancia  $D(S, c)$  y mínima redundancia  $R(S)$  permite definir el método mRMR según la Ec. 6 [37], [55]:

$$\varphi(D, R) = \max_S (D(S, c) - R(S)) \quad \text{Ec. 6}$$

Las variables que maximizan la función  $\varphi(D, R)$  se encasillan en el mismo grupo de variables seleccionadas junto con la variable objetivo [37].

Este proceso se realiza de forma iterativa. Partiendo del conjunto de características sin procesar  $X$ , si se ha seleccionado la característica óptima  $S_{m-1}$ , en la siguiente iteración la aplicación de mRMR continúa buscando la característica óptima en el espacio de características restante  $X - S_{m-1}$ . Así, haciendo uso de la función  $\varphi(D, R)$ , para las siguientes iteraciones pueden ilustrarse como la Ec. 7.

$$\max \varphi(D, R) \rightarrow \max_{x_j \in X - S_{m-1}} \left[ I(x_j; c) - \frac{1}{m-1} \sum_{x_i \in S_{m-1}} I(x_j; x_i) \right] \quad \text{Ec. 7}$$

El proceso finaliza cuando el grupo de variables de salida alcanza la dimensión  $K$  fijada inicialmente por el usuario [55].

Aunque la aplicación de mRMR original es la reducción de la dimensionalidad teniendo en cuenta tanto la relevancia como la redundancia de las características del espacio muestral, este método también es aplicable para la segmentación de las variables de monitorización de un proceso. Estableciendo el valor de  $K$  igual al número de características del conjunto de datos, con la aplicación de la técnica mRMR es posible obtener una distribución del proceso de  $N$  grupos según la cual cada bloque viene representado por una característica  $x_i$ , que maximiza la función  $\varphi(D, R)$ , y un conjunto de características  $[x_k, \dots, x_m]$  de menor relevancia y mayor redundancia respecto la variable principal  $x_i$ .

## 2.2 DEEP LEARNING PARA LA DETECCIÓN DE FALLOS

### 2.2.1. Deep learning

Dentro del aprendizaje automático, el aprendizaje profundo o *Deep Learning* (DL) emerge como un subcampo especializado que se distingue por su uso de redes neuronales profundas. Mientras que los métodos tradicionales de ML requieren el conocimiento del dominio y la extracción manual de características [13], DL elimina parte de este procesamiento previo de datos mediante el uso de múltiples capas de redes neuronales que pueden aprender representaciones jerárquicas de los datos.

Este enfoque ha revolucionado numerosas áreas de la IA debido a su mayor precisión y eficiencia en tareas como el reconocimiento de voz o imágenes, el procesamiento del lenguaje natural y la toma de decisiones automatizadas, sin requerir una programación explícita. También se ha demostrado que la aplicación de DL es adecuada para la detección y el diagnóstico de fallas de procesos multivariantes [56] eliminando las barreras de los métodos tradicionales señaladas.

#### 2.2.1.1 Clasificación de las técnicas de DL

Los modelos de DL se pueden clasificar en función de la intervención humana en el etiquetado previo de los datos de entrada como [9], [43]:

- **Aprendizaje supervisado.** Utiliza conjuntos de datos etiquetados para, lo que implica la intervención humana para la clasificación previa de los datos de entrada. Algunos ejemplos comunes de problemas de aprendizaje supervisado son la clasificación y la regresión.
- **Aprendizaje no supervisado.** Detecta patrones o subgrupos latentes en conjuntos de datos no etiquetados, es decir, sin necesidad de intervención humana o conocimiento del dominio. Aplicaciones comunes de problemas de aprendizaje no supervisado son la agrupación o *clustering*, la reducción de la dimensionalidad y detección de anomalías.
- **Aprendizaje semisupervisado.** Combina elementos de los enfoques supervisado y no supervisado. El sistema aprende el comportamiento de datos donde solo algunas muestras están etiquetadas.

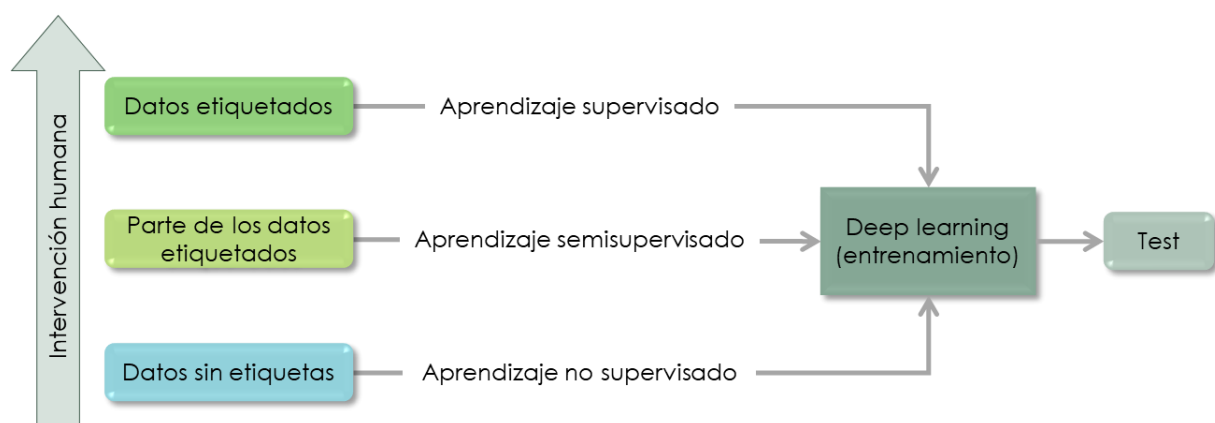


Figura 9. Clasificación de las técnicas de DL según la naturaleza de los datos

La popularidad de los algoritmos de DL ha aumentado significativamente en los últimos años gracias al aumento de la capacidad computacional y a la mayor disponibilidad de grandes bases de datos necesarias para su entrenamiento.

En los siguientes apartados se presenta la estructura básica de una red neuronal y el modo de funcionamiento de sus unidades. Posteriormente, se introducen las principales arquitecturas de redes neuronales empleadas en la diagnosis de procesos concluyendo con un caso concreto del aprendizaje no supervisado denominado autoencoder variacional, sobre el cual se desarrolla la metodología propuesta en este Trabajo.

### 2.2.1.2 Estructura y funcionamiento básico de redes neuronales

Las redes neuronales son modelos computacionales inspirados en el funcionamiento del cerebro humano [57]. Cada red neuronal consta de una serie de capas de nodos (neuronas) conectados entre sí. Estas capas incluyen una capa de entrada, una o más capas ocultas y una capa de salida. Aunque una red neuronal con una sola capa puede realizar predicciones aproximadas, las capas ocultas adicionales ayudan a optimizar y refinar la precisión.

Cada neurona recibe una o más señales numéricas ( $x_1, x_2, \dots, x_n$ ), cada una con un peso asociado ( $w_1, w_2, \dots, w_n$ ). La neurona realiza una suma ponderada de estos inputs y aplica una función de activación para agregar una relación no lineal [58]. Finalmente, se transmite la señal transformada a las neuronas de la siguiente capa.

Matemáticamente, la operación de una neurona se expresa según la Ec. 8:

$$z = \sum_{i=1}^n w_i x_i + \beta \quad \text{Ec. 8}$$

donde  $\beta$  es el sesgo (bias) de la neurona [57]. Para la obtención de la salida de la neurona se aplica la función de activación,  $\phi$ , a la suma lineal ponderada  $z$ :

$$y = \phi(z) \quad \text{Ec. 9}$$

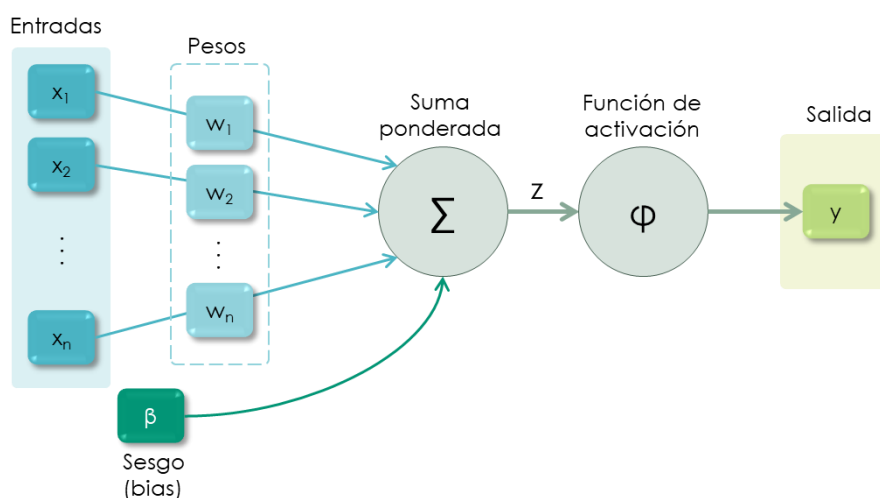
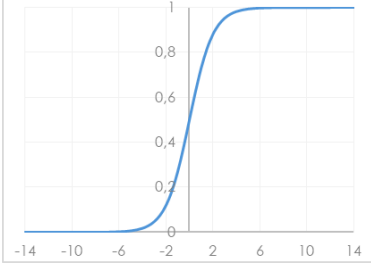
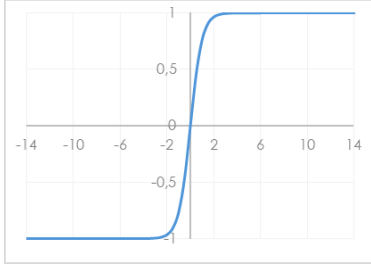
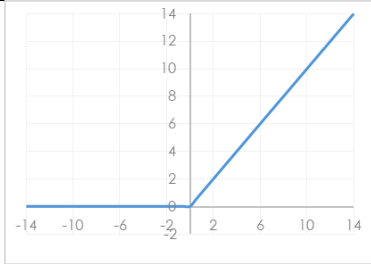
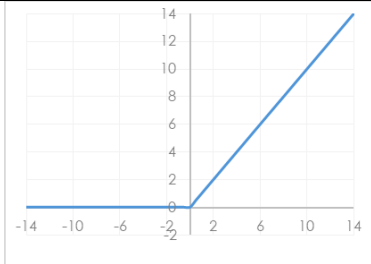


Figura 10. Clasificación de las técnicas de DL según la naturaleza de los datos

En la Tabla 1 se presentan las funciones de activación más populares en la literatura [57], [59]:

Tabla 1. Funciones de activación utilizadas en redes neuronales

Función	Expresión matemática	Rango	Gráfica
<b>Sigmoide</b>	$f(x) = \frac{1}{1 + e^{-x}}$	(0,1)	
<b>Tangente hiperbólica</b>	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	(-1,1)	
<b>ReLU</b> (Rectified Linear Unit)	$f(x) = x^+ = \max(0, x)$	[0,∞)	
<b>Leaky ReLU</b>	$f(x) = \begin{cases} x, & \text{si } x > 0 \\ m \cdot x, & \text{si } x \leq 0 \end{cases}$	(-∞, ∞)	

La selección de la función de activación depende del problema específico en cuestión y de la arquitectura de la red. Tanto la función sigmoide como la tangente hiperbólica presentan zonas de saturación donde los gradientes son casi nulos, lo que genera el problema de la desaparición del gradiente. En contraste, ReLU es capaz de proporcionar gradientes estables en el rango de  $[0, \infty)$ , evitando así este problema [8].

El entrenamiento de las redes neuronales implica el ajuste de los pesos de las conexiones entre neuronas mediante algoritmos de optimización [60], típicamente propagación de datos hacia adelante para realizar predicciones (*forward propagation*) y la retropropagación del error (*backpropagation*).

En función de cómo se interconecten las capas y cómo se comuniquen los nodos, surgen diferentes tipos de arquitecturas [57], siendo algunas de las más comunes las redes neuronales artificiales (ANN), las redes neuronales convolucionales (CNN), las redes neuronales recurrentes (RNN), las redes neuronales de memoria a largo plazo (LSTM) o los autoencoders (AE) [8], [39], [56], [61], [62].

En particular, los AE se consideran como una generalización no lineal de la técnica de PCA [39] descrita en el apartado 2.1.1.3.

Dado que en la mayoría de los procesos industriales es complicado disponer de una gran cantidad de muestras de datos etiquetados para diferentes tipos de fallas, los AE se presentan como una alternativa por ser redes entrenadas de forma no supervisada [39]. Esto significa que pueden identificar patrones y características inherentes a los datos de operación normales de un proceso sin necesidad de etiquetas o información previa sobre el comportamiento anómalo.

Esta característica los ha convertido en un modelo de DL de gran interés en el ámbito de la detección y diagnóstico de fallos [8], aunque también para tareas como la reducción de dimensionalidad o la generación de características.

### 2.2.2. Autoencoder (AE)

Los autoencoders (AE) surgen en el contexto del aprendizaje automático no supervisado [8] con el fin de abordar la extracción de características de conjuntos de datos con relaciones entre variables no lineales, obteniendo así otro espacio de menor dimensión, pero manteniendo la estructura y propiedades esenciales [30].

Los AE destacan por su buen rendimiento en el aprendizaje de representación [63] lo cual es particularmente útil en tareas como la compresión de datos, reconstrucción de imágenes o en el preentrenamiento de redes profundas.

Gracias a su alta capacidad para aprender representaciones, su estructura simple y su facilidad de entrenamiento, los AE también se han empleado ampliamente en el ámbito de la detección de fallos de procesos industriales [8], siendo éste su punto de interés para el desarrollo de este proyecto.

El AE se compone de tres partes principales: un codificador o *encoder*, un decodificador o *decoder* y un espacio latente que interconecta los dos anteriores [64]:

- **Encoder.** Comprime la entrada en una representación de tamaño reducido.
- **Espacio latente:** Capa oculta en la que se encuentra la representación comprimida de los datos
- **Decoder.** Reconstruye la entrada original a partir de la representación comprimida

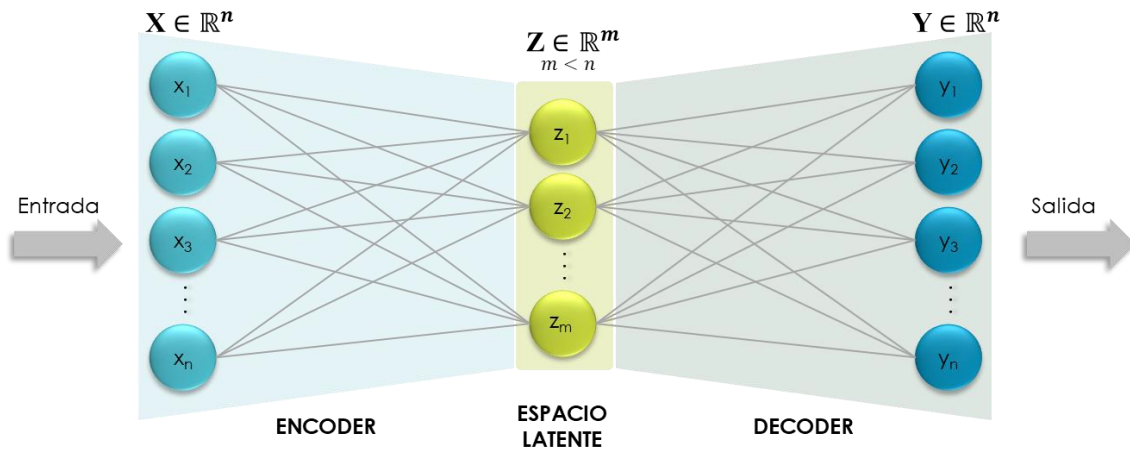


Figura 11. Arquitectura de un autoencoder

El *encoder* es una red neuronal que toma los datos de entrada y extrae sus características principales comprimiendo el conjunto original a una capa oculta de menor dimensionalidad conocida como espacio latente o cuello de botella [65] [64].

Es importante señalar que, aunque el AE básico y el PCA están estrechamente relacionados, AE es mucho más flexible. Cuando se utiliza con relaciones lineales, AE podría aprender el mismo subespacio que con el PCA [8], [66]. Sin embargo, AE puede capturar relaciones no lineales entre variables, lo que amplía su ámbito de aplicación a procesos más complejos, mientras que PCA solo es aplicable a datos con relaciones lineales.

El último componente de un AE es el *decoder*. Su misión es reconstruir las entradas del espacio latente y obtener una salida lo más aproximada al dominio de entrada original del *encoder* [65] [64].

En cuanto a su formulación matemática, si se denomina  $X \in \mathbb{R}^n$  al vector de características de entrada del AE y, por definición, del *encoder*, e  $Y \in \mathbb{R}^n$  a la reconstrucción obtenida a la salida del decoder, las funciones que describen al *encoder* ( $f_\phi$ ) y *decoder* ( $g_\theta$ ) se expresan como [8]:

$$Z = f_\phi(X) = \varphi(W_e X + \beta_e) \quad \text{Ec. 10}$$

$$Y = g_\theta(Z) = \hat{\varphi}(W_d Z + \beta_d) \quad \text{Ec. 11}$$

donde  $Z \in \mathbb{R}^{m < n}$  es la salida del encoder;  $\Phi$  y  $\theta$  son los parámetros – pesos ( $W$ ) y sesgos ( $\beta$ ) – del encoder y decoder, respectivamente; y  $\varphi$  y  $\hat{\varphi}$  son las funciones de activación (ver Tabla 1).

El objetivo del entrenamiento del AE es minimizar la diferencia la entrada  $X$  y la reconstrucción del mapeo  $Y$  [61]. A esta diferencia se denomina error de reconstrucción o función de pérdida y se calcula típicamente como el error cuadrático medio (MSE) [61] o la raíz del MSE (RMSE).

Los parámetros en AE se entrenan minimizando la función de pérdida según la Ec. 12 mediante un algoritmo de optimización (ej. gradiente descendente mediante algoritmo de *backpropagation*, Adam, RMSprop [8]):

$$L_{AE}(\phi, \theta) = \frac{1}{n} \sum_1^n \|X - Y\|^2 = \frac{1}{n} \sum_1^n \|X - g_{\theta}(f_{\phi}(X))\|^2 \quad \text{Ec. 12}$$

La arquitectura de los AE puede hacerse más compleja a medida que se incluyen capas ocultas adicionales tanto al *encoder* como al *decoder*, lo que permite, no obstante, añadir profundidad a la red y una mejor extracción de características [64].

Pese a su gran potencial, el espacio latente del AE que es, en definitiva, la parte esencial del algoritmo, presenta una serie de inconvenientes que no pueden ser ignorados si el objetivo es hacer uso de él en aplicaciones de control de procesos multivariantes.

El espacio latente de un AE se compone del conjunto de puntos que representan las características principales extraídas del espacio de entrada. Sin embargo, los AE no imponen ninguna restricción sobre la distribución de estos datos en el espacio latente. Esto puede resultar en una representación latente donde los datos se produzcan las siguientes situaciones donde:

- Los datos no están centrados en el origen de coordenadas (0,0)
- Los datos no se distribuyen de manera uniforme por el espacio
- Existen discontinuidades

Estos tres aspectos implican que sea difícil muestrear efectivamente el espacio latente pudiendo caer regiones densamente pobladas, lo que conlleva un sobreajuste, o regiones no representativas, ocasionando una mala reconstrucción de la entrada. Además, las discontinuidades en el espacio latente implican que pequeñas variaciones pueden llevar a grandes saltos o cambios abruptos en la interpolación entre puntos y, con ello, a una mala reconstrucción de la entrada.

En el año 2013, Kingma y Welling [67] propusieron un modelo generativo avanzado de AE conocido como autoencoder variacional (VAE). Esta nueva variante aborda con éxito las tres problemáticas señaladas anteriormente del AE mediante la imposición de una distribución normal estándar centrada en el origen sobre el espacio latente.

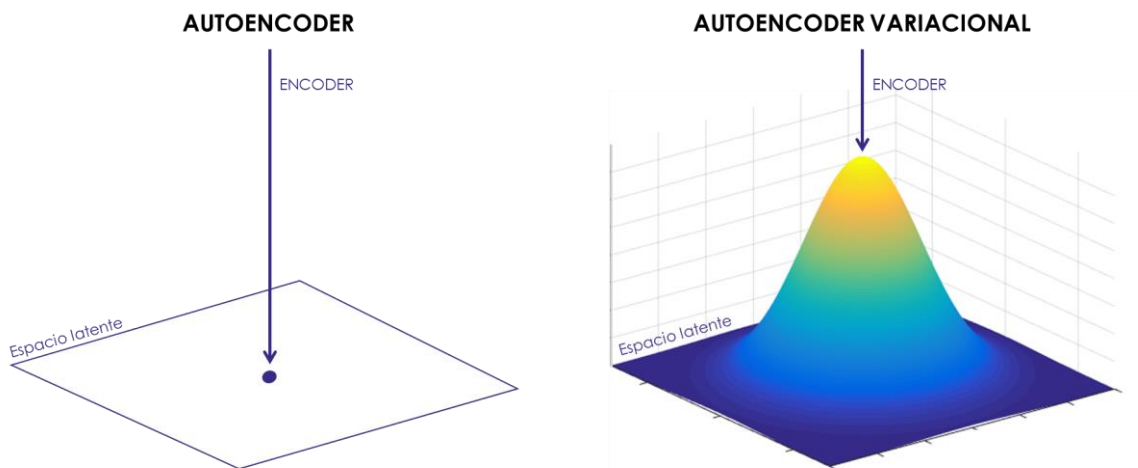


Figura 12. Comparativa representación espacio latente AE y VAE



Esto lo hace particularmente interesante para el control de procesos [63] porque pueden convertir un conjunto de datos de entrada no lineales y/o que no siguen en una distribución normal, en un espacio que cumple las condiciones requeridas para la aplicación de los gráficos de control estadístico como  $T^2$  de Hotelling [30].

### 2.2.3. Autoencoder variacional (VAE)

Al igual que el AE, el VAE se compone de una arquitectura *encoder-decoder* pero se diferencia del modelo básico en que introduce una regularización en el espacio latente haciéndolo continuo y estructurado [68]. Esto se logra al aproximar la distribución de las características extraídas por el *encoder* como una distribución normal multivariante definida como  $N(\mu(x), \sigma(x)^2)$  siendo  $x$  el conjunto de entrada del VAE [8], [69].

Como se muestra en la Figura 13, el objetivo del *encoder* del VAE es aprender la distribución de probabilidad de los datos de entrada  $x$  produciendo un vector de medias  $\mu(x)$  y un vector de desviaciones estándar  $\sigma(x)$ . A partir de estos dos vectores, se muestrea el vector de variables latentes aleatorias continuas,  $z$ , que se utiliza como entrada del *decoder* para la reconstrucción de los datos.

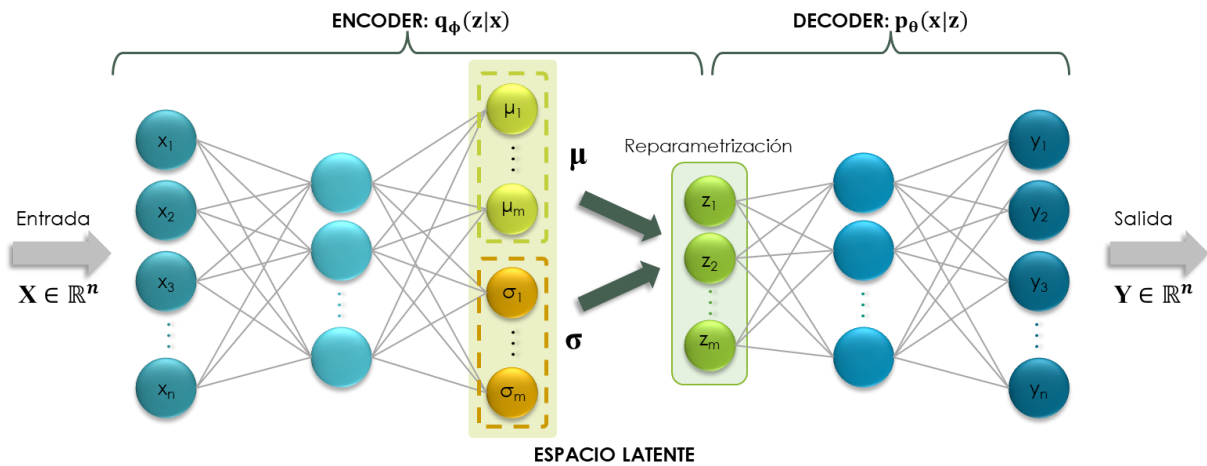


Figura 13. Esquema estructura del VAE con una capa

#### 2.2.3.1 Formulación matemática

El VAE considera un modelo probabilístico tanto para el *encoder* como para el *decoder*, descritos como  $q_\phi(z|x)$  y  $p_\theta(x|z)$  respectivamente [69], donde  $\Phi$  y  $\theta$  representan los parámetros de la red (pesos y sesgos del *encoder* y *decoder*).

En primer lugar, se modela la distribución  $p_\theta(x|z)$  para después inferir en la posterior  $p_\theta(z|x)$ . Sin embargo, el cálculo de la distribución posterior verdadera  $p_\theta(z|x)$  es intratable y difícil de resolver debido a la complejidad de la integral de la verosimilitud  $p_\theta(x)$  [70]:

$$p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz \quad \text{Ec. 13}$$

siendo  $p_\theta(z)$  la distribución a priori del espacio latente (distribución normal estándar).

Para abordar este problema, el VAE aplica inferencia variacional para aproximar  $p_\theta(z|x)$  con una distribución más simple y manejable,  $q_\phi(z|x)$ , que sigue una distribución normal aproximada [69]:

$$q_\phi(z|x) \sim N(\mu_\phi(x), \sigma_\phi(x)^2) \quad \text{Ec. 14}$$

En la inferencia variacional de Bayes, la probabilidad marginal de cada punto de datos  $x_i$  se describe a partir de la siguiente expresión [65]:

$$\log p_\theta(x_i) = D_{\text{KL}}(q_\phi(z|x_i)||p_\theta(z|x_i)) + L(\theta, \phi, x_i) \quad \text{Ec. 15}$$

El primer término de la Ec. 15 corresponde a la divergencia de Kullback-Leibler ( $D_{\text{KL}}$ ) entre las distribuciones  $q_\phi(z|x_i)$  y  $p_\theta(z|x_i)$ . Este término mide cuán similares son las dos distribuciones de probabilidad, entre otras palabras, mide cuánto se desvía la distribución posterior aproximada  $q_\phi(z|x_i)$  de la distribución real posterior  $p_\theta(z|x_i)$  [8].

El segundo término de la Ec. 15 corresponde al límite inferior variacional en la probabilidad marginal o límite inferior de evidencia (ELBO), que se escribe como [70]:

$$L(\theta, \phi, x_i) = E_{q_\phi(z|x_i)}[\log p_\theta(x_i|z)] - D_{\text{KL}}(q_\phi(z|x_i)||p_\theta(z)) \quad \text{Ec. 16}$$

Como se comprueba en la expresión ante, el ELBO consta de dos términos. El primer término es el error de reconstrucción y correspondería al MSE o RMSE del entrenamiento de un AE tradicional. El segundo término es la divergencia KL entre la distribución  $q_\phi(z|x_i)$  y la distribución normal estándar esperada,  $p_\theta(z)$ .

De este modo, VAE convierte el problema de maximizar la verosimilitud marginal en maximizar el ELBO [70] [8]

En general, el término de  $D_{\text{KL}}$  puede obtener una solución analítica precisa. Sin embargo, el término  $E_{q_\phi(z|x_i)}[\log p_\theta(x_i|z)]$  es difícil de calcular al no ser derivable. Como estrategia, VAE utiliza el truco de reparametrización para entrenar el modelo mediante la retropropagación del gradiente [67] a través del muestreo de  $z \sim q_\phi(z|x)$  [69]:

$$E_{q_\phi(z|x_i)}[\log p_\theta(x_i|z)] \approx \frac{1}{L} \sum_{l=1}^L \log p_\theta(x_i|z_{i,l}) \quad \text{Ec. 17}$$

donde la variable latente  $z$  se expresa como una función determinista de la variable ruido  $\varepsilon$ , siendo  $\varepsilon$  un valor arbitrario tomado de la distribución  $\varepsilon(l) \sim N(0, I)$ , y  $L$  es la dimensión de  $z$  [67], [69]:

$$z_{i,l} = \mu_i + \sigma_i \odot \varepsilon_l \quad \text{Ec. 18}$$

Esto permite derivar la ELBO con respecto a  $\theta$  y  $\Phi$  usando técnicas estándar de optimización de redes neuronales.

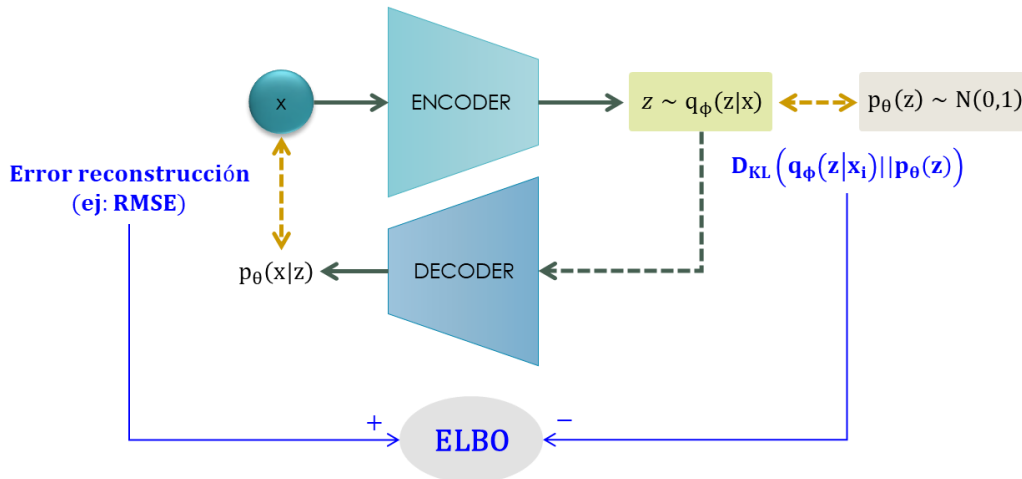


Figura 14. Función de pérdida VAE: error de reconstrucción y divergencia KL

Se comprueba que el VAE no sólo se diferencia del AE en la distribución probabilística del *encoder*, sino también en la función de pérdida. En el AE tradicional, la función de pérdida cuenta únicamente con el término de error de reconstrucción, que compara los datos reconstruidos con la entrada original.

El VAE introduce además un término de regularización del espacio latente expresado como la divergencia de KL entre la distribución de probabilidad latente predicha de los datos y la distribución normal estándar, que puede expresarse como:

$$D_{KL}[N(\mu, \sigma) || N(0,1)] = \frac{1}{2} \sum_{j=1}^J (1 + \log(\sigma)^2 - \mu^2 - \sigma^2) \quad \text{Ec. 19}$$

Uniendo las expresiones de las Ec. 17 y Ec. 19, el entrenamiento del VAE puede representarse mediante la maximización del ELBO según la siguiente fórmula:

$$L(\theta, \phi, x_i) = \frac{1}{2} \sum_{j=1}^J (1 + \log(\sigma_{i,j})^2 - \mu_{i,j}^2 - \sigma_{i,j}^2) + \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(x_i | z_{i,l}) \quad \text{Ec. 20}$$

### 2.2.3.2 Entrenamiento del VAE

Teniendo en consideración la formulación anterior, el proceso de entrenamiento del VAE se puede resumir en los siguientes pasos iterativos:

- 1) Los datos pasan a través del *encoder* para obtener  $\mu_{\phi}(x)$  y  $\sigma_{\phi}(x)$ .
- 2) Se aplica el truco de reparametrización para generar una muestra aleatoria  $z$ .
- 3) La variable latente aleatoria  $z$  se introduce en el *decoder* para obtener la salida reconstruida.
- 4) Cálculo de la ELBO a partir del error de reconstrucción y la  $D_{KL}$ .
- 5) Actualización de los parámetros  $\theta$  y  $\Phi$  para maximizar la ELBO.

### 3. DETECCIÓN DE FALLOS CENTRALIZADA VS DISTRIBUIDA MEDIANTE VAE

---

A partir de los fundamentos teóricos de las técnicas de control de procesos y los modelos de aprendizaje profundo no supervisados descritos en los apartados 2.1 y 2.2, se desarrolla en este capítulo la metodología propuesta para la rápida detección de posibles fallos que se den durante la operación de una EDAR aplicada tanto en métodos de control multivariable centralizados como distribuidos de proceso.

Esta metodología se fundamenta en el modelo de VAE mínimo para aprender la distribución subyacente del conjunto de datos no etiquetados del proceso y reducir la dimensionalidad de las variables del sistema de control mediante la extracción de características constituyendo un espacio latente de menor tamaño que la entrada. Sobre este espacio reducido se determinará el umbral de control del proceso para el uso de la técnica de control estadístico de procesos multivariantes  $T^2$ , descrita en el apartado 2.1.2.1.

Es importante señalar que, dado que el VAE se sintonizará con un conjunto de datos de operación bajo control, al introducir posteriormente datos del proceso con fallos, el VAE no puede reconstruir la entrada de manera efectiva, lo que resultará en un gran error de reconstrucción (salida *decoder*). Pero, además, las características del espacio latente extraídas por el *encoder* también mostrarán una desviación en comparación con el estado normal [8]. Será precisamente esta desviación la base del sistema de detección de fallas propuesto.

Al utilizar la salida del *encoder* para el control estadístico se elimina, por un lado, el tiempo de procesamiento asociado a la predicción del *decoder* y, por otro, el costo computacional al aplicarse el sistema de detección de fallas sobre una matriz de dimensionalidad inferior a la entrada/salida del VAE.

#### 3.1 PLANTEAMIENTO GENERAL

La metodología que a continuación se desarrolla, aplicada tanto al método multivariable centralizado como el distribuido, incluye en primer lugar un proceso *offline* de construcción del modelo VAE y establecimiento de umbrales de control y, en segundo lugar, se validará mediante un proceso *online* la detección de fallos.

El objetivo será detectar los fallos que se produzcan en el proceso lo más rápido posible y sin necesidad de haber entrenado el modelo con fallos etiquetados y comparar los resultados y ventajas que se obtienen según si se aplica mediante una diagnosis centralizada o distribuida del sistema.

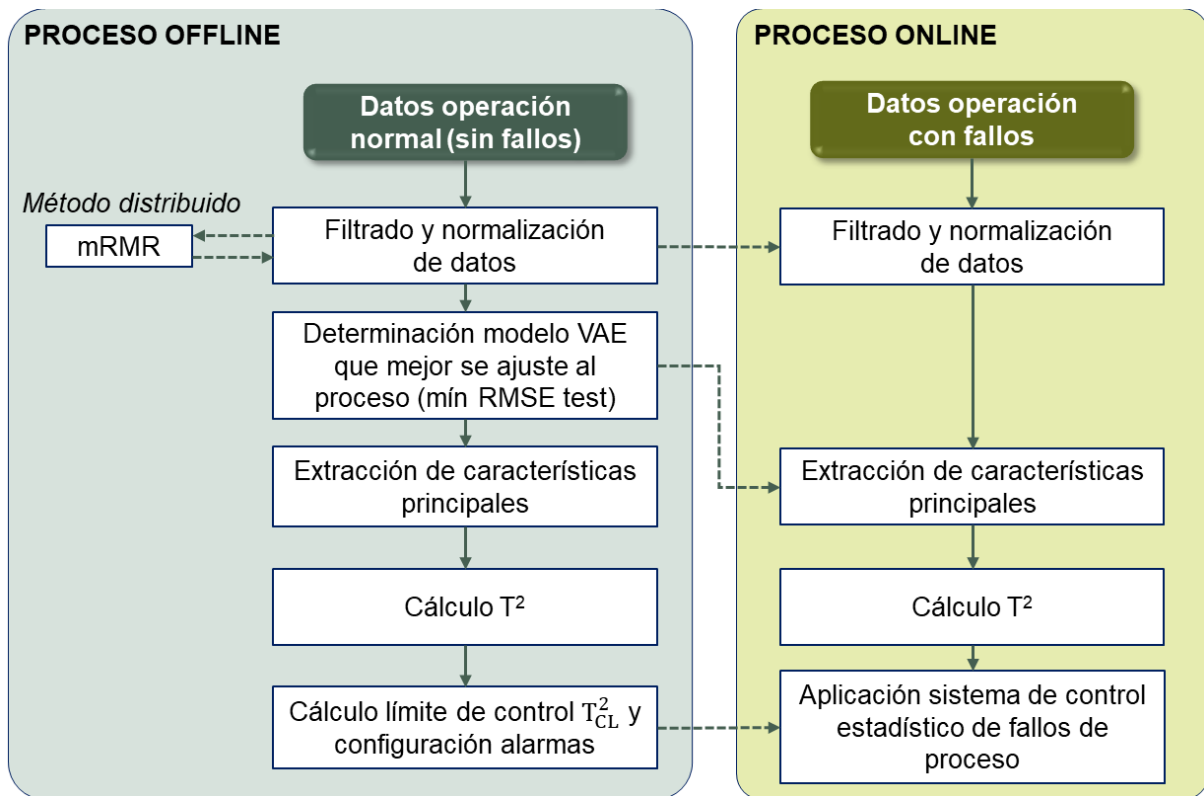


Figura 15. Esquema alto nivel metodología sistema detección temprana de fallos propuesto

Al tratarse de un modelo VAE, el espacio latente estará definido por un vector de medias,  $\mu$ , y un vector de desviaciones típicas,  $\sigma$ , y se aplicará el truco de reparametrización para el muestreo de  $z$ . Con el valor de  $z$  se determinará el estadístico  $T^2$  para posteriormente determinar el umbral superior de control  $T_{CL}^2$  (proceso *offline*) o comparar su valor con éste para el diagnóstico de fallos (proceso *online*).

Dado que el método de detección de fallos se basa en un parámetro estadístico, no pueda asegurarse que en la totalidad de ocasiones que se supere el umbral  $T_{CL}^2$  se esté produciendo un fallo.

Es por ello que en la detección de fallos del proceso *online* se configura una frecuencia mínima de fallos consecutivos, '*max\_desvíos*', según la cual desde que se detecta un primer desvío de  $T^2$  se ha de detectar al menos un cierto número consecutivo de desvíos de  $T^2$  para la consideración de fallo.

En caso de que desde que se detecta un primer desvío de  $T^2$  no se produzca de forma consecutiva '*max\_desvíos*' desvíos, se considerará como falsa alarma y no se registrará como fallo.

El valor de '*max\_desvíos*' será un parámetro a introducir manualmente por el usuario en función del objetivo de análisis y rigurosidad que quiera otorgarse al sistema y basándose en el comportamiento del mismo con datos históricos sin fallo.

Por otra parte, se describe la aplicación del sistema de detección de fallos a métodos centralizados y distribuidos. Para la segmentación de la planta en el caso distribuido se aplicará la técnica mRMR descrita en el apartado 2.1.3.2.

Esta técnica se aplica habitualmente para la selección de características y reducción de la dimensionalidad. No obstante, en el sistema que se propone en este trabajo, mRMR se emplea como método de agrupación de variables, manteniendo la dimensionalidad del conjunto.

Se incluyen a continuación una descripción a alto nivel de las etapas de los procesos *offline* y *online*:

- ▶ **Proceso *offline*.** Utilizada datos históricos de la monitorización de la planta en condiciones de operación sin fallos.
  1. Filtrado y normalización de los datos de entrada. En el caso de método distribuido, tras el filtrado de los datos y previo a la normalización, se distribuyen las variables del proceso en agrupaciones mediante la técnica mRMR.
  2. Ajuste de parámetros e hiperparámetros para la creación del VAE del proceso, estableciendo como criterio de selección el error de reconstrucción del test mínimo entre las posibles combinaciones preestablecidas, o cuyo valor se considere lo suficientemente bajo o adecuado.
  3. Extracción de características principales mediante VAE (*encoder*) y cálculo del estadístico  $T^2$ .
  4. Cálculo del umbral superior de control  $T_{CL}^2$  y valor de frecuencia de valores consecutivos en los que  $T^2$  supera el umbral con los datos históricos utilizados en el proceso *offline* para la configuración de las alarmas por fallo, '*max\_desvíos*'.
- ▶ **Proceso *online*.** Utiliza los datos recogidos por el sistema de monitorización en tiempo real de operación del proceso.
  1. Pretratamiento de datos según lo establecido en el proceso *offline* (filtrado, normalización y, en su caso, agrupación de variables).
  2. Extracción de las características principales de los datos de entrada haciendo uso del VAE seleccionado en el proceso *offline*.
  3. Cálculo de la variable de control estadístico  $T^2$  con las características del espacio latente obtenido en el punto 2.
  4. Comparación del valor de  $T^2$  con el umbral superior  $T_{CL}^2$  establecido en el proceso *offline*.
  5. Las observaciones cuyo estadístico supere el valor del umbral en un determinado número consecutivo de veces, '*max\_desvíos*', se considerará que el proceso podría no estar bajo control y, por tanto, se estaría produciendo un fallo.

A continuación, se desarrolla en detalle la metodología propuesta para la detección temprana de fallos en una EDAR mediante VAE, tanto en el caso de **detección centralizada como distribuida** siendo uno de los objetivos del Trabajo la comparación entre ambos métodos para evaluar su eficacia y velocidad de detección.

En el Anexo A. Diagrama de flujo del sistema, se incluye el diagrama del proceso global del sistema de detección temprana de fallos propuesto, incluyendo el proceso *offline* y *online* y unificando la metodología del control centralizado y distribuido.

## 3.2 DETECCIÓN CENTRALIZADA DE FALLOS

La esencia de la detección centralizada propuesta es modelar la planta objeto de estudio a partir del histórico de datos de operación registrados considerando todo el proceso como un único conjunto, es decir, sin separar las variables por grupos o subprocesos.

### 3.2.1. Proceso offline

Siguiendo el planteamiento general anterior, durante el proceso *offline* se utilizan datos históricos del proceso en operación bajo control, es decir, sin fallos.

En la Figura 16 se muestra la estructura de funcionamiento del proceso *offline* propuesto para el control centralizado del proceso:

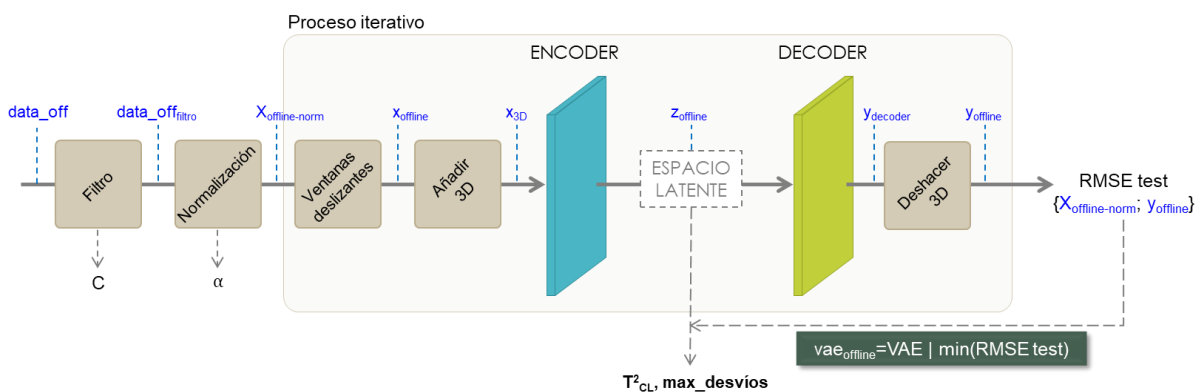


Figura 16. Diagrama del proceso *offline*, método centralizado

#### 3.2.1.1 Filtrado de datos

En primer lugar, se aplica un filtrado de los datos de entrada para eliminar aquellas variables que permanecen constantes en el tiempo,  $C$ , y que, en consecuencia, no presentan variabilidad. Estas variables no resultan de interés para el entrenamiento del VAE.

Las variables que son eliminadas durante el filtrado de datos del proceso *offline* se deberán eliminar también en el proceso *online*.

#### 3.2.1.2 Normalización

Al igual que el filtrado de datos, la etapa de normalización se aplica en el proceso *offline* y *online*, tanto en el caso de estudio centralizado como distribuido.

A cada variable de los datos filtrados se aplica la normalización sigmoide con objeto de escalar los valores de entrada al rango  $[0,1]$ .

Para cada variable se calcula un factor multiplicador  $\alpha$  de la exponencial que permita adecuar la curva de la sigmoide a los datos de la variable en cuestión. Se establece como criterio para el cálculo de  $\alpha_i$  que el valor máximo de la variable  $x_i$  tome el valor normalizado de 0,95.

El vector  $\alpha$  se almacena en un fichero para su posterior uso en el proceso *online*.

### 3.2.1.3 Entrenamiento iterativo y selección del VAE

Una vez procesados y estandarizados los datos de entrada, se procede a la configuración del modelo de VAE de la planta.

El objetivo es ajustar los parámetros e hiperparámetros del VAE de manera que se obtenga el error de reconstrucción RMSE de test mínimo o de valor aceptable para el proceso. Dado que el número de combinaciones de dichos parámetros podría ser infinito, inicialmente se definen unos valores entre los cuales iterar. Estos valores se establecen en base a una fase previa de investigación bibliográfica.

En el **Algoritmo 1** se describe el proceso de entrenamiento iterativo de modelos VAE y cálculo del error de reconstrucción del test. A continuación, se selecciona el modelo de VAE a utilizar para el sistema de detección de fallos aplicando las etapas del **Algoritmo 2**.

#### Proceso iterativo de entrenamiento

1. Sobre los datos normalizados se aplica la transformación de ventanas temporales deslizantes según el parámetro frecuencia temporal de análisis—valor preestablecido por el usuario. Esta etapa será clave para el control en línea del sistema.
2. División de los registros de entrada entre datos de entrenamiento y datos de testeo, en una relación 80/20. En el caso de este trabajo no se aplica validación cruzada (*cross validation*).
3. Adición de una tercera dimensión de ceros al conjunto de datos original. Esta práctica podría asimilarse como la conversión de una imagen en blanco y negro (2D) a una imagen en escala de grises (3D). Así, es posible aplicar la construcción de modelos de redes neuronales ampliamente utilizados en procesos de reconocimiento y clasificación de imágenes, a un proceso industrial con registros bidimensionales.
4. Entrenamiento y test del modelo VAE: guardar pesos y sesgos del VAE y calcular el RMSE entre la entrada y la salida del test.

#### Selección VAE

5. En base al RMSE del test de los distintos modelos VAE, selección del modelo de VAE cuyo RMSE del test sea mínimo



### Algoritmo 1. Proceso iterativo entrenamiento y test VAE

**Input:**  $x_{\text{offline-norm}}$ , param\_hyperparam

**Output:**  $vae_{\text{trained}}$ ,  $RMSE_{\text{test}}$

//  $vae_{\text{trained}}$ : k modelos VAE entrenados para distintas combinaciones de parámetros e hiperparámetros

//  $RMSE_{\text{test}}$ : Error cuadrático medio del test para cada  $vae_{\text{trained}}$

1. **Definir** valores de parámetros e hiperparámetros VAE:

param\_hyperparam = {filters, kernels, strides, latent\_space\_dim, batch\_size, learning\_rate, time\_analysis, epochs}

2. **for each** combinación i de elementos de param\_hyperparam **do**

$x_{\text{offline}}^{(i)} \leftarrow$  Dividir  $x_{\text{offline-norm}}$  en ventanas deslizantes tamaño time\_analysis<sup>(i)</sup>

$x_{\text{train}}^{(i)}, x_{\text{test}}^{(i)} \leftarrow$  Dividir  $x_{\text{offline}}^{(i)}$  en relación 80/20

$x_{\text{train-3D}}^{(i)}, x_{\text{test-3D}}^{(i)} \leftarrow$  Añadir a  $x_{\text{train}}^{(i)}$  y  $x_{\text{test}}^{(i)}$  una 3ª dimensión de ceros

$vae_{\text{trained}}^{(i)} \leftarrow$  Creación, compilación y entrenamiento del VAE con  $x_{\text{train-3D}}^{(i)}$ , filters<sup>(i)</sup>, kernels<sup>(i)</sup>, strides<sup>(i)</sup>, latent\_space\_dim<sup>(i)</sup>, batch\_size<sup>(i)</sup>, learning\_rate<sup>(i)</sup>, epochs<sup>(i)</sup>

**Calcular**  $RMSE_{\text{test}}^{(i)}(x_{\text{test}}^{(i)}, y_{\text{test}}^{(i)}) \leftarrow$  Predicción  $y_{\text{test-3D}}^{(i)} = vae_{\text{trained}}^{(i)}(x_{\text{test-3D}}^{(i)})$

**Guardar**  $vae_{\text{trained}}^{(i)}$ , combinación i de param\_hyperparam y  $RMSE_{\text{test}}^{(i)}$

**end for**

### Algoritmo 2. Selección modelo VAE

**Input:**  $vae_{\text{trained}}$ ,  $RMSE_{\text{test}}$

**Output:**  $vae_{\text{offline}}$  // VAE con menor RMSE test para el proceso objeto de estudio

1. **Leer**  $RMSE_{\text{test}}$ : i, combinación param-hyperparam i,  $RMSE_{\text{test}}^{(i)}$ , con  $i=1, \dots, k$
2. **Buscar**  $i | RMSE_{\text{test}}^{(i)} = \min(RMSE_{\text{test-1}}, \dots, RMSE_{\text{test-k}})$
3.  $vae_{\text{offline}} \leftarrow$  **Seleccionar** modelo  $vae_{\text{trained}}^{(i)}$

#### 3.2.1.4 Determinación umbral de control estadístico $T_{CL}^2$

Tras la sintonización del modelo VAE, se procede al cálculo del umbral superior del estadístico de procesos multivariados  $T_{CL}^2$  para la posterior detección de fallos en el proceso online. El umbral inferior de  $T^2$  es 0.

El umbral  $T_{CL}^2$  no puede determinarse con el valor crítico de la distribución F como en PCA ya que parte de la suposición teórica de que la muestra de datos sea una distribución gaussiana.

Por ello, se calculará  $T_{CL}^2$  como el percentil p de los valores de  $T^2$  calculados con los datos de entrenamiento según se define en el **Algoritmo 3**.

### Algoritmo 3. Cálculo umbral $T^2$ detección de fallos

**Input:**  $vae_{offline}$ ,  $x_{train-3D}$

**Output:**  $T_{CL}^2$  // Umbral superior de control del estadístico  $T^2$  calculado durante el entrenamiento de  $vae_{offline}$

1.  $z_{offline} \leftarrow$  **Predicción** encoder  $vae_{offline}(x_{train-3D})$
2. **Calcular** media de cada característica del espacio latente,  $\bar{z}_j = \frac{1}{m} \sum_{i=1}^m z_{offline}^{(j)}$
3. **Calcular** inversa matriz covarianza de las características del espacio latente,  $\Sigma_z^{-1}$
4. **Calcular**  $T^2(i) = (z_{offline}^{(i)} - \bar{z}_j)^T \Sigma_z^{-1} (z_{offline}^{(i)} - \bar{z}_j)$
5.  $T_{CL}^2 = \text{percentil}(T^2, p)$
6. **Guardar**  $T_{CL}^2, \bar{z}, \Sigma_z^{-1}$

A continuación, utilizando el mismo registro de datos históricos de la planta en operación sin fallo, se evalúa el número de ocasiones que  $T^2(i) > T_{CL}^2$  y la frecuencia en que esto se produce de forma consecutiva.

En función de los resultados se configura el valor de '*max\_desvíos*' como número mínimo de veces consecutivas que ha de detectarse un desvío para considerarlo fallo, por lo general, el valor de frecuencia máxima registrada.

No obstante, este paso requiere el control humano dado que en algunos casos el valor máximo puede ser puntualmente excesivo y, por tanto, es preferible analizar cada caso particular.

### Algoritmo 4. Cálculo límite consecutivo de desvíos para consideración de fallo

**Input:**  $T^2, T_{CL}^2$  // Parámetros obtenidos durante proceso *offline*

**Output:** *max\_desvíos* // Vector de observaciones en las que se detecta fallo

1. **Inicializar** contador,  $cont = 0$
2. **for**  $i$  in range(longitud( $T^2$ )) **do**
  - if**  $T_i^2 > T_{CL}^2$  **then**
    - Se detecta posible fallo. Incrementar valor contador,  $cont += 1$

```

else:
    if cont ≠ 0 then
        Guardar valor contador, v_cont
    end if
    Reiniciar contador, cont = 0
end if
end for
3. Evaluar límite consecutivo de desvíos, max_desvíos = max(v_cont)
    
```

### 3.2.2. Proceso online

Una vez determinado el modelo VAE de la planta y configurado el sistema de detección de fallos, se evalúa su eficacia y velocidad de detección utilizando datos históricos del proceso donde se conoce que se ha producido un fallo en alguna de sus variables.

En la Figura 17 se muestra la estructura de funcionamiento del proceso *online* propuesto para el control centralizado del proceso:

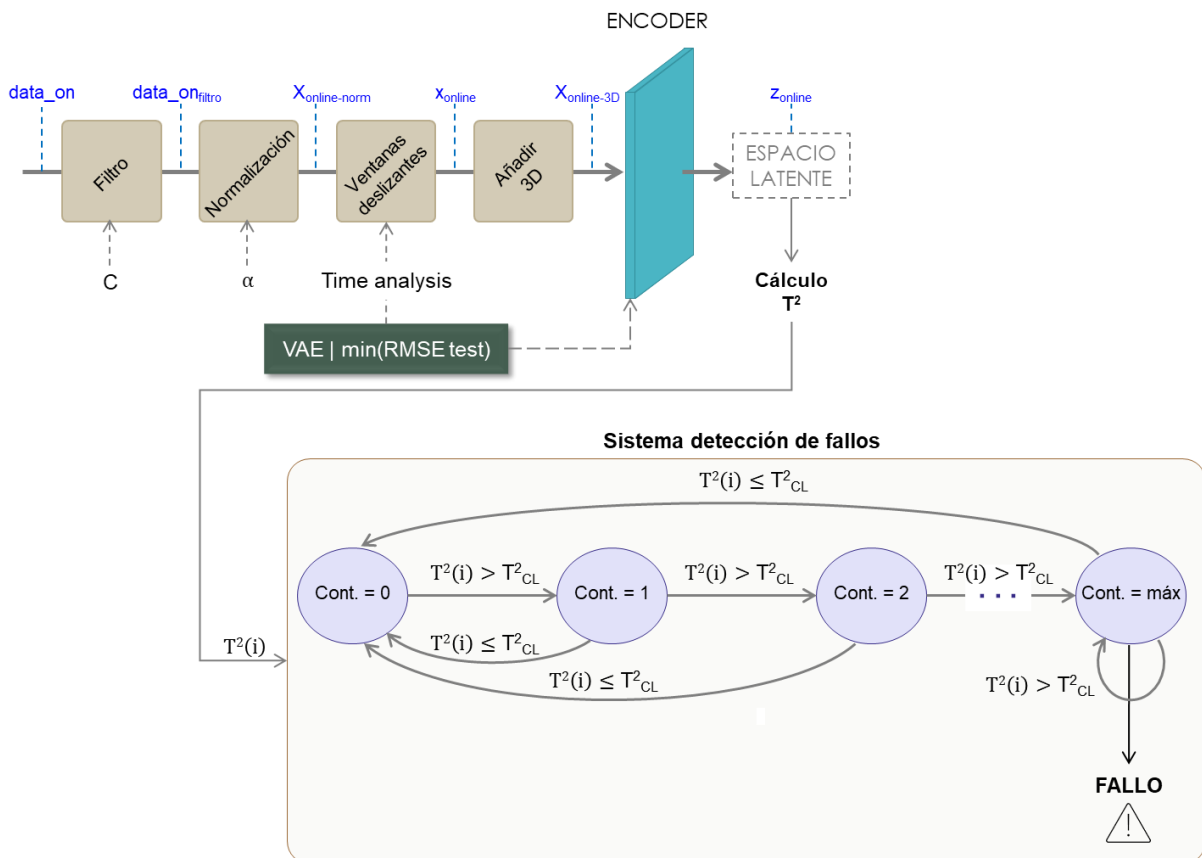


Figura 17. Diagrama del proceso *online*, método centralizado

Se emplea  $T^2$  como parámetro de control estadístico del proceso multivariado. En el caso que el índice  $T^2$  exceda el umbral de control establecido en el proceso *offline*,  $T^2(i) > T_{CL}^2$ , en el número de veces consecutivas establecido en el proceso *offline*, '*max\_desvíos*', se considera que se ha producido un fallo. En caso contrario, el proceso se encuentra en un estado normal de funcionamiento.

### 3.2.2.1 Filtrado y normalización de datos

El proceso *online* comienza con el pretratamiento y normalización de los datos de entrada eliminando las variables constantes del proceso *offline* (C) y escalando los valores de entrada al rango [0,1] mediante la normalización sigmoide utilizando el factor de ponderación  $\alpha$  calculado anteriormente.

### 3.2.2.2 Extracción de características mediante VAE

Los datos normalizados pasan a través del *encoder* del modelo VAE seleccionado en el proceso *offline* ( $vae_{offline}$ ) para la predicción de las características del espacio latente (**Algoritmo 5**).

Previamente, siguiendo los pasos del proceso *offline*, los datos se transforman por el método de ventanas deslizantes con paso 1 y tamaño de la frecuencia temporal de análisis correspondiente al  $vae_{offline}$  y se les añade una tercera dimensión de ceros ( $x_{online-3D}$ ) para poder aplicar la red neuronal entrenada.

#### Algoritmo 5. Extracción de características mediante VAE

**Input:**  $x_{online-norm}$ ,  $vae_{offline}$

**Output:**  $z_{online}$

1. **Cargar**  $vae_{offline}$
2.  $x_{online-3D} \leftarrow$  Dividir  $x_{offline-norm}$  en ventanas deslizantes tamaño  $vae_{offline}\{time\_analysis\}$  y añadir 3<sup>a</sup> dimensión de ceros.
3.  $z_{online} \leftarrow$  **Predicción** *encoder*  $vae_{offline}(x_{online-3D})$

### 3.2.2.3 Cálculo estadístico $T^2$

A partir de la predicción de la salida del *encoder* (espacio latente), se determina para cada ventana de observaciones el valor del estadístico  $T^2$  siguiendo lo expuesto en el apartado 2.1.2.1 y aplicando el procedimiento definido en el **Algoritmo 6**. El estadístico  $T^2$  se construye a partir de las características principales del proceso y los vectores de la media y matriz de covarianza obtenidos en el entrenamiento del modelo (**Algoritmo 3**).

**Algoritmo 6. Cálculo estadísticos de control proceso online**

**Input:**  $z_{\text{online}}, \bar{z}, \Sigma_z^{-1} // \bar{z}, \Sigma_z^{-1}$  obtenidos en el proceso offline con datos de entrenamiento

**Output:**  $T^2(i)$

1. **Cargar** media,  $\bar{z}$ , e inversa de la matriz de covarianza,  $\Sigma_z^{-1}$ , del espacio latente
2. **Calcular** estadístico de control  $T^2$  para cada ventana de datos:

$$T^2(i) = \left( z_{\text{online}}^{(i)} - \bar{z} \right)^T \cdot \Sigma_z^{-1} \cdot \left( z_{\text{online}}^{(i)} - \bar{z} \right), \text{ con } i = 1, \dots, (t' - \text{time}_{\text{analysis}} + 1)$$

3.2.2.4 Detección de fallos

Una vez calculado  $T^2$  para cada una de las ventanas de observaciones, se comprueba si su valor es superior al umbral determinado en el proceso *offline*.

En caso afirmativo, se considera que es posible que se haya producido un fallo solo en caso de que el número consecutivo de veces que se detecta el desvío supera el mínimo establecido, '*max\_desvíos*'. De no producirse dicho número mínimo consecutivo de desvíos, no se considera como fallo (**Algoritmo 7**).

**Algoritmo 7. Detección de fallos**

**Input:**  $T^2(i), T_{CL}^2, \text{max\_desvíos}$

**Output:** faults // Vector de observaciones en las que se detecta fallo

1. **Inicializar contador**, cont = 0
2. **for** i in range(longitud( $T^2$ )) **do**
  - if**  $T_i^2 > T_{CL}^2$  **then** // Posible fallo.
    - cont** += 1
    - if** cont  $\geq$  max\_desvíos **then**
      - Añadir** i a vector[faults] // instante de tiempo con fallo
    - end if**
  - else** // No fallo
    - Reiniciar** cont = 0
  - end if**
- end for**

Este proceso permitirá en un futuro al personal responsable del control de la planta detectar y actuar con mayor rapidez ante posibles fallos en el proceso, tomando las acciones correctivas que se estimen oportunas.

### 3.3 DETECCIÓN DISTRIBUIDA DE FALLOS

En el apartado 2.1.3 se plantea la detección de fallos distribuida como solución para afrontar la alta dimensionalidad y carga computacional de los procesos industriales complejos. Para la descomposición de las variables de control de la planta de estudio se utiliza la técnica mRMR.

Los procesos *offline* y *online* del sistema de detección de fallos distribuido se fundamentan en los algoritmos descritos en el control centralizado (apartado 3.2). La diferencia radica en que, en lugar de considerar la planta como un único bloque, se establece un modelo VAE para los distintos grupos de variables obtenidos tras la aplicación de mRMR. El sistema de detección de fallos se aplica individualmente para cada grupo:

► **Proceso *offline*:**

1. Filtrado de datos
2. Distribución de variables por grupos mediante mRMR (**NUEVO**)
3. Normalización
4. Entrenamiento iterativo y selección del VAE
5. Determinación umbral de control estadístico  $T_{CL}^2$

► **Proceso *online*:**

6. Filtrado, selección de variables (**NUEVO**) y normalización de datos
7. Extracción de características mediante VAE
8. Cálculo estadístico  $T^2$
9. Detección de fallos

Como se demostrará más adelante, la detección del fallo en un grupo de variables no implica que la localización de la falta sea en una de dichas variables, pudiendo estar ocasionado en cualquier otra parte del proceso. Es decir, esta metodología servirá para la detección, pero no para la identificación de fallos.

#### 3.3.1. Proceso *offline*

Como se indica anteriormente, la particularidad del proceso *offline* del método distribuido frente al descrito en el caso centralizado (apartado 3.2.1) es que, tras el filtrado de datos, se aplica la técnica mRMR para la descomposición adecuada del proceso en grupos teniendo en consideración la relevancia y redundancia de las variables de control.

En la Figura 18 se muestra la estructura de funcionamiento del proceso *offline* propuesto para el control distribuido del proceso:

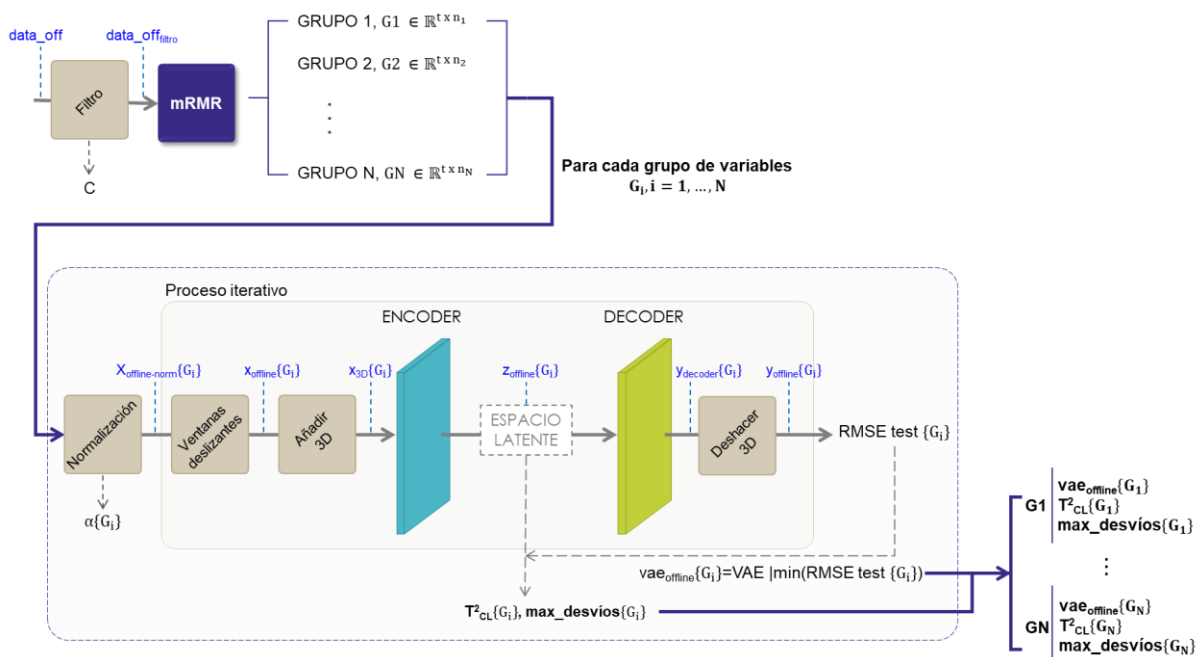


Figura 18. Diagrama del proceso *offline*, método distribuido

Tras la distribución de variables, el proceso *offline* definido en el método centralizado se aplica de manera independiente para uno de los N grupos de variables. Como resultado, se obtienen tantos VAE y  $T_{CL}^2$  como grupos de variables haya.

### 3.3.1.1 Distribución de variables por grupos mediante mRMR

Dada la alta dimensionalidad que puede alcanzar el sistema de control de una EDAR y con el fin de comparar el método multivariable centralizado del distribuido, se aplica la técnica de mRMR para la asociación de variables en grupos estableciendo como objetivo de salida de mRMR la misma dimensión que los datos de entrada.

La técnica mRMR del **Algoritmo 8** calcula el valor de máxima relevancia y mínima relevancia de las características de entrada respecto a una variable objetivo, comenzando por la variable '1' del fichero.

En cada iteración se identifican las variables que mayor correlación tienen con la variable objetivo como aquellas que su valor de mRMR supera un cierto umbral – *var\_NoRelevant*. La primera variable por debajo del umbral pasará a ser la variable objetivo del siguiente grupo de variables a analizar.

Se ha establecido el umbral como la media de los valores resultantes del cálculo de mRMR a partir del cual se distingue entre variables correlacionadas con la objetivo y aquellas que no guardan relación con la misma.

### Algoritmo 8. mRMR

**Input:** data\_off<sub>filtro</sub>

**Output:** variablesGrupos // N grupos var. de máx relevancia y var. redundantes asociadas

1. **Copiar**  $x \leftarrow \text{data\_off}_{\text{filtro}}$
2. **Crear** lista variablesPpales = ['1']
3. **for** variable in variablesPpales **do**
  - if** n° variables(x)  $\geq 2$  **then**
    - Copiar**  $y \leftarrow \text{data\_off}_{\text{filtro}}[:, \text{variable}]$
    - Eliminar** variable objetivo de  $x \leftarrow x[:, \sim \text{variable}]$
    - Calcular** res\_mRMR  $\leftarrow$  regresión mRMR siendo: características = x, variable objetivo = y, dimensión salida = dimensión data\_off<sub>filtro</sub>; método relevancia = f-statistic; método redundancia = correlación de Pearson
    - Establecer** umbral = media(res\_mRMR)
    - var\_NoRelevant  $\leftarrow \text{list}(i \in x \mid \text{res\_mRMR}(i) > \text{umbral})$  // Var. redundantes
    - Añadir** var\_NoRelevant a variablesGrupos{variable}
    - Eliminar** var\_NoRelevant de x,  $x \leftarrow x[:, \sim \text{var\_NoRelevant}]$
    - Añadir**  $j \in x \mid \text{res\_mRMR}(j) = \max(\text{res\_mRMR} < \text{umbral})$  a variablesPpales // Variable objetivo del siguiente grupo
  - end if**
- end for**

Una vez determinadas las agrupaciones de variables, los pasos a seguir en el sistema propuesto serán los mismos que los definidos en el método centralizado. La única diferencia radica en los datos de entrada que, en el caso distribuido, corresponderán a las variables del grupo de estudio en cuestión.

### 3.3.2. Proceso online

Para cada grupo de variables se evalúa el sistema de detección temprana de fallos utilizando los mismos datos históricos del proceso con fallo que se emplean en el método centralizado para su posterior comparación.

En la Figura 19 se muestra la estructura de funcionamiento del proceso *online* propuesto para el control distribuido del proceso:



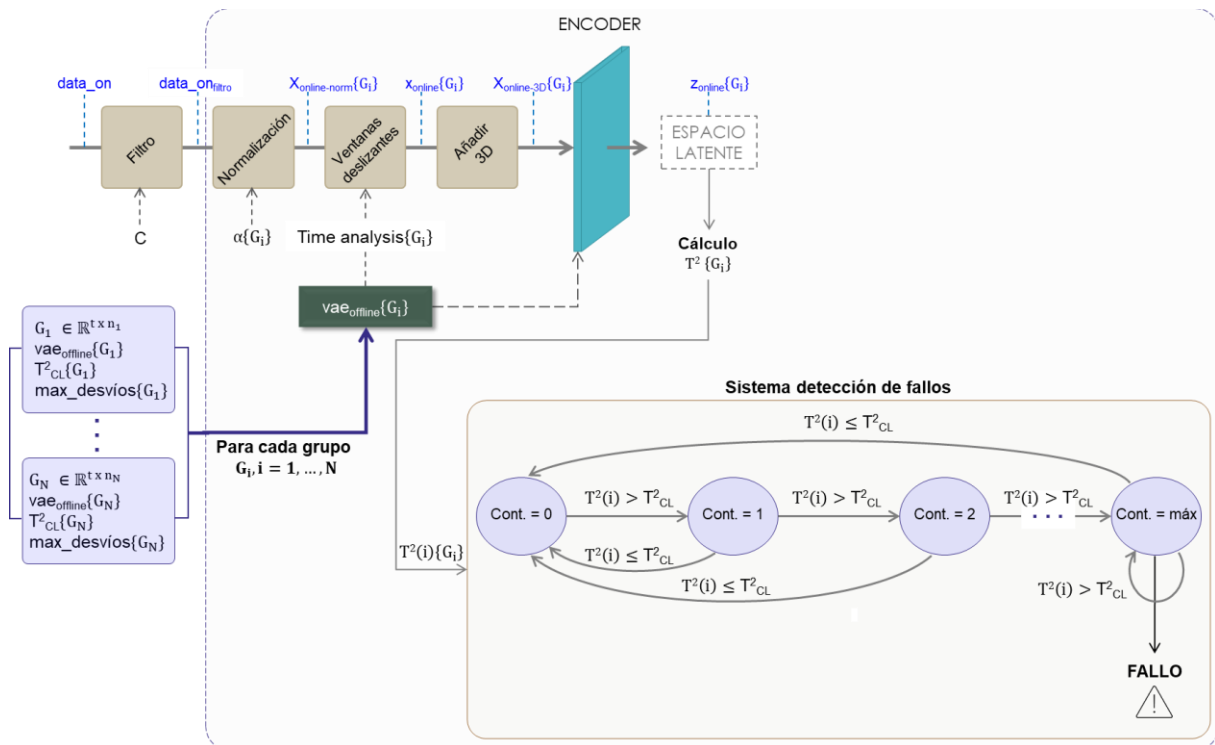


Figura 19. Diagrama del proceso *online*, método distribuido

El proceso *online* comienza con el pretratamiento y normalización de los datos de entrada siguiendo lo aplicado en el proceso *offline*. Tras el filtrado de datos y previo a la normalización, se seleccionan las variables incluidas dentro del grupo de variables sobre el cual se aplicará el modelo del VAE correspondiente. Este paso se realizará para cada una de las N agrupaciones de variables.

A continuación, se sigue el mismo procedimiento que en el método centralizado, obtenido finalmente un gráfico de control por cada uno de los N grupos de variables.

Así, será de interés analizar si existen agrupaciones de variables que mejor o más rápido detecten el fallo.

## 4. EXPERIMENTACIÓN Y RESULTADOS

En el presente Capítulo se aplica la metodología de detección temprana de fallos basada en el modelo de VAE mínimo, desarrollada en el Capítulo 3, al caso de estudio de una EDAR. La variedad y complejidad de los procesos que se llevan a cabo en una EDAR, incluyendo procesos mecánicos, químicos y biológicos, y los estrictos controles de calidad del vertido, hacen aún más interesante y necesario el desarrollo de nuevos procesos de detección temprana de fallos mediante técnicas de DL.

Se describe la planta utilizada como caso de estudio, la metodología experimental aplicada y el análisis de los resultados obtenidos, todo ello con el fin de evaluar la metodología desarrollada en base al VAE mínimo y comparar el método de control distribuido frente al centralizado.

En el Anexo B se indica el lenguaje de programación y librerías empleadas para la experimentación.

Para la ejecución del **Algoritmo 1**, que requiere cierta potencia computacional y, por tanto, no es viable realizarlo con un ordenador de uso doméstico, se ha utilizado un sistema clúster de máquinas con 252 GB de memoria RAM y una frecuencia de procesador de 1,5-3,5 GHz, facilitado por la Universidad de Valladolid.

### 4.1 CASO DE ESTUDIO

Para la evaluación de la metodología detección temprana de fallos basada en VAE propuesta se ha utilizado un set de datos obtenido mediante la simulación del funcionamiento de una EDAR en una situación de operación normal, sin fallos, y una situación de operación con perturbaciones que producen fallos en el proceso.

Para ello se ha empleado el modelo de referencia utilizado para el estudio de estrategias de control de EDAR conocido como *Benchmark Simulation Model No 2* (BSM2) [72]. La Figura 20 corresponde a la descripción general de este modelo y los 20 puntos de control del proceso:

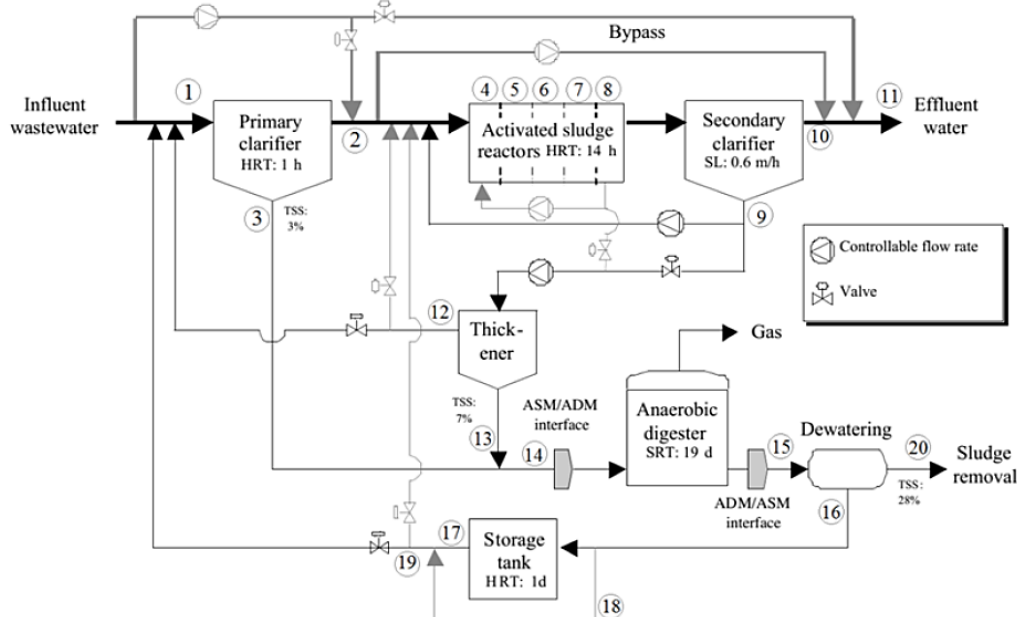


Figura 20. Diagrama del proceso BSM2 [72], [73]

En concreto, se analiza un total de 141 variables de las principales características de seguimiento del tratamiento de las aguas residuales y que definen su calidad por variables fisicoquímicas, así como el propio flujo y temperatura de proceso.

En concreto, las variables analizadas y que compondrán el set de datos empleado para la evaluación metodología propuesta son:

- DQO (demanda química de oxígeno)
- O<sub>2</sub> (oxígeno disuelto)
- Sales (alcalinidad [HCO<sub>3</sub><sup>-</sup>])
- N (nitratos y nitritos)
- SS (sólidos en suspensión)
- Caudal
- Temperatura
- Kla (coeficiente de transferencia de oxígeno)

A excepción de Kla, las otras siete variables o características del flujo del proceso se analizan a la entrada y/o salida de las distintas etapas del proceso, según se refleja en la Figura 20, lo que permite evaluar la eficiencia de cada una de estas etapas en términos de depuración de la corriente.

La simulación realizada en el modo de operación sin fallos registra durante un total de 301 días, con un paso temporal de 15 minutos (28.900 registros), el valor de las diferentes variables de control del proceso. De este modo, se considera que la planta ha funcionado bajo control en condiciones ambientales cambiantes (estacionalidad). Esta base de datos será la empleada en los procesos *offline* descritos.

A continuación, se han realizado 8 simulaciones en las que se ha introducido un fallo en alguna de sus variables. Los fallos se simulan para distintos instantes de tiempo y de distinta una magnitud, ambos prefijados por el usuario. Cada una de estas simulaciones se realizan para un periodo de 3 días 15 min, con el mismo paso temporal de registro de 15 minutos (total, 289 registros).

Los fallos simulados en el proceso se han introducido en las siguientes variables:

- Fallo 1: SS afluente
- Fallo 2: SS reactor 1
- Fallo 3: Caudales Q<sub>r</sub> y Q<sub>w</sub> (r: espesador de lodos y w: deshidratación)
- Fallo 4: Caudales decantador primario
- Fallo 5: Caudales tanque almacenamiento
- Fallo 6: Kla
- Fallo 7: O<sub>2</sub> disuelto (reactor 1)
- Fallo 8: Sales reactor 1

Estos 8 ficheros de datos con fallo serán utilizados en los procesos *online* descritos para la evaluación de la eficacia y rapidez del sistema de detección de fallos basado en VAE propuesto.

En la siguiente tabla se indican los instantes de fallo simulados y su amplitud:

Tabla 2. Tiempo y amplitud simulación fallos

Fallo	Instante fallo	Amplitud del fallo
1	39	3%
2	49	42%
3	192	6500%
4	58	0,5%
5	71	0,09%
6	68	2%
7	10	30%
8	39	30%

En el Anexo C. Simulación fallos EDAR se representan los gráficos de las variables del proceso donde se han simulado los fallos, comparando la situación de operación normal frente a la situación con cada uno de los 8 fallos indicados anteriormente.

Es importante señalar que la amplitud de fallo simulada en algunos de los casos listados anteriormente es tan pequeña que apenas resulta perceptible en los ficheros. Esto permitirá analizar si el sistema funciona también en estas situaciones.

## 4.2 METODOLOGÍA EXPERIMENTAL

Para la aplicación de la metodología descrita anteriormente es necesario preestablecer algunos parámetros en base al caso de estudio y que serán comunes para la aplicación centralizada y distribuida.

### Entrenamiento VAE

Para la definición y entrenamiento del VAE (**Algoritmo 1**) se definen en la Tabla 3 los valores de los distintos parámetros e hiperparámetros entre los que iterar ya que, de lo contrario, se estaría ante un número infinito de opciones. Estos valores se han seleccionado utilizando como referencia valores empleados por otros autores en AE o VAE que utilizan como set de datos datos históricos de EDAR o procesos de tipología similar como la base de datos *Tennessee Eastman Process*.

Por otra parte, se ha establecido como objetivo una reducción de la dimensionalidad del espacio latente del 50 o 75% respecto la dimensión de los datos de entrada. De este modo, la dimensión del espacio latente sobre el cual se calcula el estadístico  $T^2$  y se aplica la detección de fallos se reduce considerablemente respecto la entrada, con la consecuente mejora del rendimiento de ejecución.

En cuanto a la frecuencia temporal de análisis (ventanas de entrada) se establece en 4 o 6 dado que los registros del proceso en el caso de aplicación son cuatrorarios y así se limita la frecuencia de análisis a un máximo de 1 h y 1 h 30 min, respectivamente.

Tabla 3. Hiperparámetros y parámetros en el diseño del VAE

Hiperparámetro	Valores	
Tamaño del espacio latente	50%input, 75% input	
Tamaño del lote ( <i>batch size</i> )	32, 64, 128	[ [74], [75], [76] ]
Tasa de aprendizaje ( <i>learning rate</i> )	0,001; 0,0001	[ [74], [75], [76] ]
Épocas	100, 200	[ [61], [74], [76] ]
Parámetro	Valores	
Filtro	32, 64, 128	[ [74], [77] ]
<i>Kernels</i> o matrices de pesos	3	[ [56], [74], [77] ]
<i>Strides</i> o desplazamiento del filtro	1	[ [56], [77] ]
Optimizador	'Adam'	[ [30], [61], [74], [76] ]
Función de activación	'ReLU'	[ [30], [56], [61] ]
Frecuencia temporal	4, 6	

#### Determinación umbral de control estadístico

Para la determinación del umbral de control estadístico (**Algoritmo 3**) se estudiará la eficacia de establecer  $T_{CL}^2$  como el percentil 95 o 99 de los  $T^2$  calculados a partir de los datos de entrenamiento en el proceso *offline*, seleccionando finalmente el valor a emplear en función de los resultados que se obtengan.

Un percentil más estricto permite reducir la tasa de falsas alarmas, pero también puede recaer una detección de fallo ineficiente o tardía al pasar desapercibidos incrementos leves de  $T^2$ .

#### Configuración sistema detección de fallos

Como se ha descrito en el apartado 3.2.1.4, se ha de configurar el número consecutivo de desvíos de  $T^2$  para la consideración de fallo: '*max\_desvíos*'. Se establece en este estudio un valor de '*max\_desvíos*' de 2, es decir, que se produzcan al menos dos desviaciones consecutivas de  $T^2$  por encima del umbral para considerarlo fallo.

### 4.3 ANÁLISIS DE LOS RESULTADOS

La evaluación del sistema que se describe a continuación comienza con el uso del fichero de datos del modelo BSM2 durante 300 días de simulación.

No obstante, como se demostrará más adelante, al disponer de ficheros con fallo de solo los 3 primeros días de simulación y dado que el VAE aprende el patrón de los datos de la EDAR en la zona estable, es decir, a partir de los 119 días de simulación, se repetirá el proceso para centrar el aprendizaje de la red neuronal en estos 3 primeros días y así poder validar posteriormente el sistema.

#### 4.3.1. Diagnóstico centralizado

##### 4.3.1.1 Proceso offline

#### A. Simulación 300 días

##### ❑ Filtrado de datos y normalización

Mediante el pretratamiento de los datos de entrada, de las 141 variables de entrada se descartan 14 variables o características del proceso que permanecen constantes a lo largo del tiempo de muestreo.

Las 127 variables restantes se escalan al rango [0;1] por normalización sigmoide.

##### ❑ Entrenamiento iterativo y selección de VAE

Normalizados los datos de entrada, se procede al entrenamiento y test de los modelos VAE generados con las distintas combinaciones de parámetros e hiperparámetros recogidos en la Tabla 3, generando un total 144 modelos distintos de VAE (**Algoritmo 1**) para el control centralizado requiriendo para ello un tiempo de ejecución de 3 d 23 h 4 min 0,19 s.

Siguiendo el criterio establecido de RMSE test mínimo, la combinación de parámetros e hiperparámetros seleccionada para el caso de estudio de la diagnóstico centralizado se recoge en la Tabla 4. No se recogen aquellos parámetros e hiperparámetros que se han mantenido constantes y, por tanto, no varían en los distintos modelos generados (kernel: 3x3, stride: 1).

Tabla 4. Parametrización modelo VAE centralizado RMSE test mínimo, 300 días

MÉTODO CENTRALIZADO: PARAMETRIZACIÓN VAE						
Frecuencia temporal	Tasa aprendizaje	Épocas	Filtros	Tamaño espacio latente	Batch	RMSE test
4	0,001	100	64	95 (↓75%)	32	$3,74 \cdot 10^{-2}$

Para verificar el correcto comportamiento del VAE, se muestra en los siguientes gráficos algunas de variables del proceso utilizando el fichero de entrenamiento (operación bajo control) y reflejando la entrada (azul) y salida (rojo) del VAE:

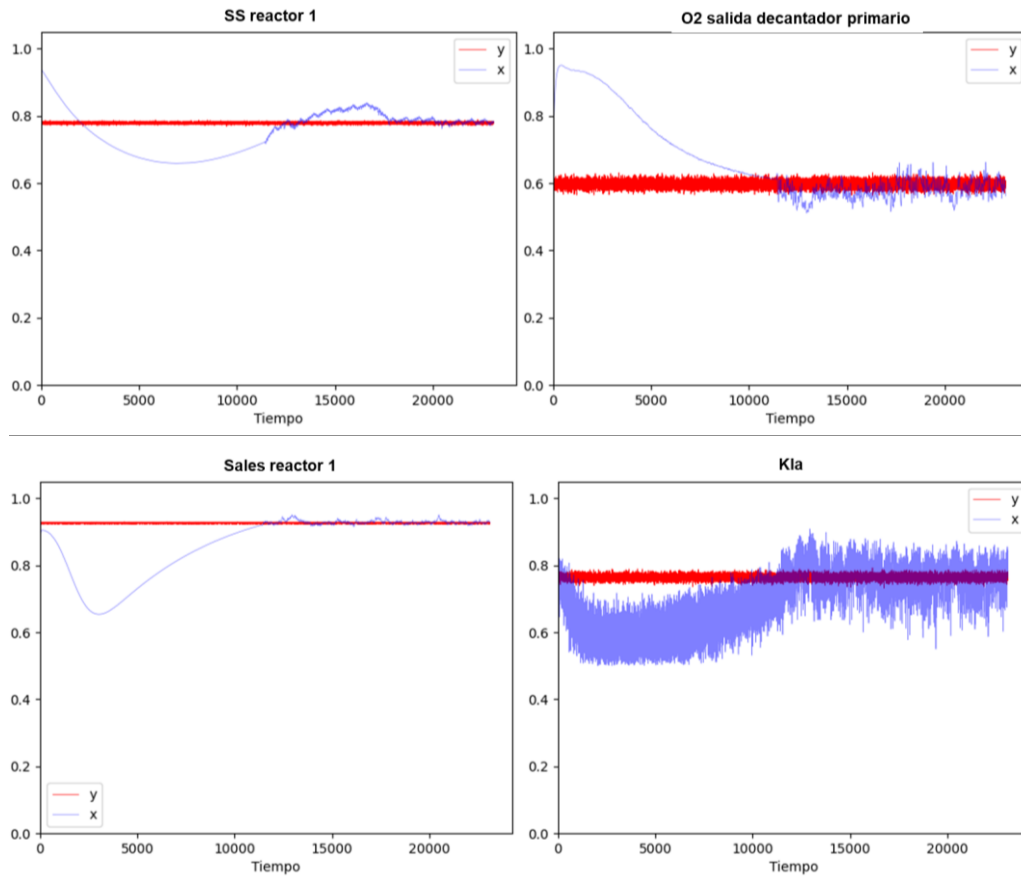


Figura 21. Validación entrenamiento VAE centralizado, 300 días. Comparativa entrada y salida VAE con datos entrenamiento (fichero sin fallos)

En los gráficos de la Figura 21 se observa que la red neuronal aprende el patrón de funcionamiento de la planta en la **zona estable** (a partir del registro 11.424), pero no el periodo transitorio.

Esto se debe a que está trabajando con el **modelo de VAE mínimo**, que cuenta únicamente con una capa oculta lo que limita la complejidad del aprendizaje de la red y las relaciones entre variables.

Por ello, el modelo olvida muy rápido durante el entrenamiento conservando sólo el aprendizaje sobre los últimos datos introducidos en el VAE, siendo en este caso la zona del estacionario. Este problema se solucionaría incrementando el número de capas intermedias del VAE y el tamaño de estas (número neuronas).

En consecuencia, y añadido a que los ficheros con fallo disponibles corresponden a la zona del transitorio y el modelo no se ajusta a dicho perfil, se opta por la aplicación del sistema de detección de fallos en el estudio del transitorio. Esto implica entrenar nuevamente el modelo reduciendo la entrada a los **3 primeros días de simulación**.

Además, por lo general, los sistemas de control se centran en la operación del proceso en el estacionario y no en los arranques de procesos. Es en estos arranques donde las variables pueden dispararse ocasionando picos y valles acentuados que podrían ocasionar graves problemas al sistema en general, como temperaturas excesivamente elevadas o concentraciones extremas que deterioren los equipos y conducciones, acaben con el cultivo bacteriano u otros.

## B. Simulación 3 días

### ❑ Filtrado de datos y normalización

Tras el pretratamiento de los datos de entrada y previo a la normalización, se seleccionan los 289 primeros registros del fichero (3 días).

### ❑ Entrenamiento iterativo y selección de VAE

A continuación, se procede al entrenamiento y test de los modelos VAE. En esta ocasión, el tiempo de ejecución del programa se reduce significativamente al haberse reducido a una centésima parte la dimensión de la entrada, pasando a requerir 10 h 47 min 1,62 s, es decir, casi 9 veces menos que con los 300 días de registro.

La combinación de parámetros e hiperparámetros seleccionada para el caso de estudio de 3 días del modelo centralizado se recoge en la siguiente tabla:

Tabla 5. Parametrización modelo VAE centralizado RMSE test mínimo, 3 días

MÉTODO CENTRALIZADO: PARAMETRIZACIÓN VAE						
Frecuencia temporal	Tasa aprendizaje	Épocas	Filtros	Tamaño espacio latente	Batch	RMSE test
4	0,001	200	32	95 (↓75%)	64	$3,71 \cdot 10^{-2}$

Se muestran en los siguientes gráficos algunas de variables del proceso utilizando el fichero de entrenamiento (operación bajo control) durante los 3 primeros días y reflejando la entrada (azul) y salida (rojo) del VAE:

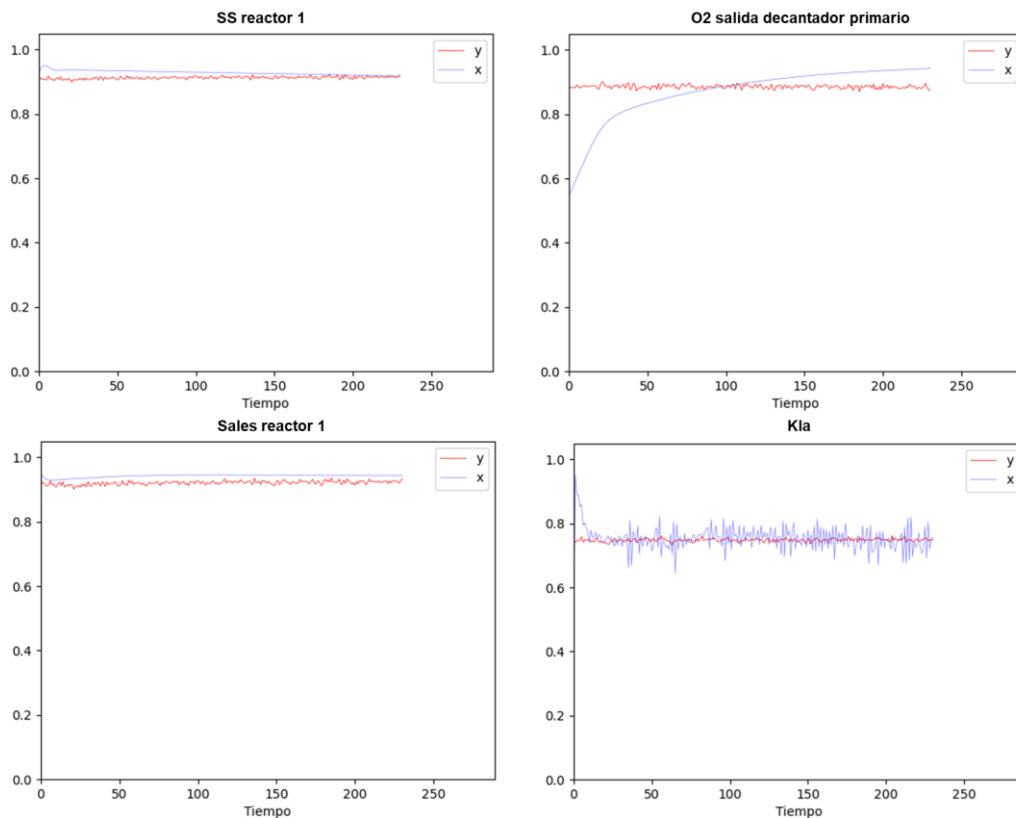


Figura 22. Validación entrenamiento VAE centralizado, 3 días. Comparativa entrada y salida VAE con datos entrenamiento (fichero sin fallos)



Se comprueba que el perfil utilizando únicamente los 3 primeros días de simulación difiere significativamente al observado en la Figura 21 en la zona transitoria.

En el caso de 3 días, **con la nueva parametrización el modelo se ajusta al patrón de funcionamiento en todo el periodo simulado**, con mejor resultado en los casos en los que la curva temporal muestra una variación lineal (ej.: SS o sales reactor 1) u oscila entre una constante (ej.: K1a). Como se indicó anteriormente, esto se debe a que al tener por objeto el estudio del VAE mínimo en el sistema propuesto, el aprendizaje sobre modelos complejos es limitado.

#### ❑ Determinación de $T_{CL}^2$

Utilizando el modelo  $VAE_{offline}$  seleccionado y los datos de entrenamiento del fichero de datos de EDAR en operación normal (sin fallo), se calcula el estadístico  $T^2$  para con las características del espacio latente (**Algoritmo 3**).

El valor de  $T_{CL}^2$  se determinará estudiando los casos del percentil 95 y 99 de los valores  $T^2$  calculados para determinar cuál es más favorable para el caso de estudio.

Tabla 6. Determinación umbrales de control estadístico método centralizado

Método	$T_{CL,95}^2$	$T_{CL,99}^2$
Centralizado	216,40	239,82

Con el fichero de datos sin fallo utilizado para el entrenamiento y test del modelo VAE, se aplica la metodología del proceso *online* de detección de fallos para determinar la tasa de falsas alarmas que se producen como consecuencia de utilizar un método estadístico, utilizando  $T_{CL}^2$  del 95 y 99% y con un criterio de 2 desvíos de  $T^2$  consecutivos para considerarlo fallo:

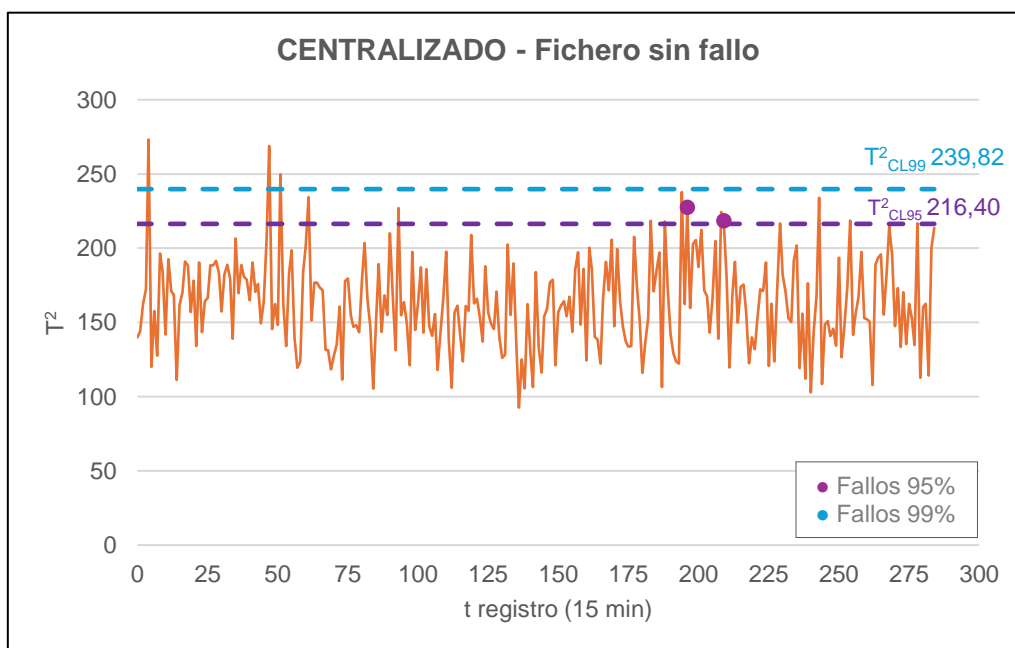


Gráfico 1. Umbral superior control  $T_{CL}^2$  método centralizado

Se comprueba que con el umbral del percentil 99%, al ser más restrictivo, utilizando el fichero de datos sin fallo no se registran falsas alarmas.

Por el contrario, con el percentil del 95% se obtiene un total de 2 falsas alarmas, que corresponde a una tasa del 0,70%, valor que se considera aceptable para el sistema desarrollado.

En consecuencia, se empleará el umbral  $T_{CL}^2$  obtenido con el **percentil del 95%** en el proceso online, para no despreciar pequeños incrementos de  $T^2$  que pudieran pasar desapercibidos con el percentil más restrictivo del 99%.

#### 4.3.1.2 Proceso online

Una vez configurado el modelo VAE y el sistema de detección de fallos, se procede a evaluar su funcionamiento (proceso *online*). En el caso de estudio de la EDAR se analizan los 8 ficheros que recogen los fallos antes señalados.

Teniendo en cuenta que cada fallo corresponde a variables distintas y se han simulado de distinta magnitud e instante de tiempo, para cada fallo estudiado y cada umbral  $T_{CL}^2$  se analizan, entre otros, los siguientes puntos:

- Falsas alarmas (fallos detectados antes del instante de fallo real)
- Tiempo registro primera detección de fallo tras instante de fallo real y magnitud  $T^2$  correspondiente.
- Tiempo de demora entre la primera y segunda alarma de fallo

### **Fallo 1: SS afluyente**

#### ► Simulación:

El fallo 1 (sólidos en suspensión corriente entrada EDAR) se observa a partir del registro de control número 39, correspondiente al tiempo de simulación 9 h 45 min. La amplitud del fallo introducida es del 0,03%.

#### ► Sistema detección de fallos:

El sistema de detección fallos centralizado no es capaz de detectar el fallo 1 ya que no se produce ninguna alarma a partir del instante en el que se introduce el fallo. Esto puede deberse a que la amplitud del fallo es muy pequeña y no supone un incremento del estadístico  $T^2$  por encima del umbral establecido.

Por otra parte, en el Gráfico 2 se comprueba que se produce una falsa alarma en el registro 27, lo cual es aceptable teniendo en cuenta la tasa de falsas alarmas del método centralizado antes señaladas del 0,70%.

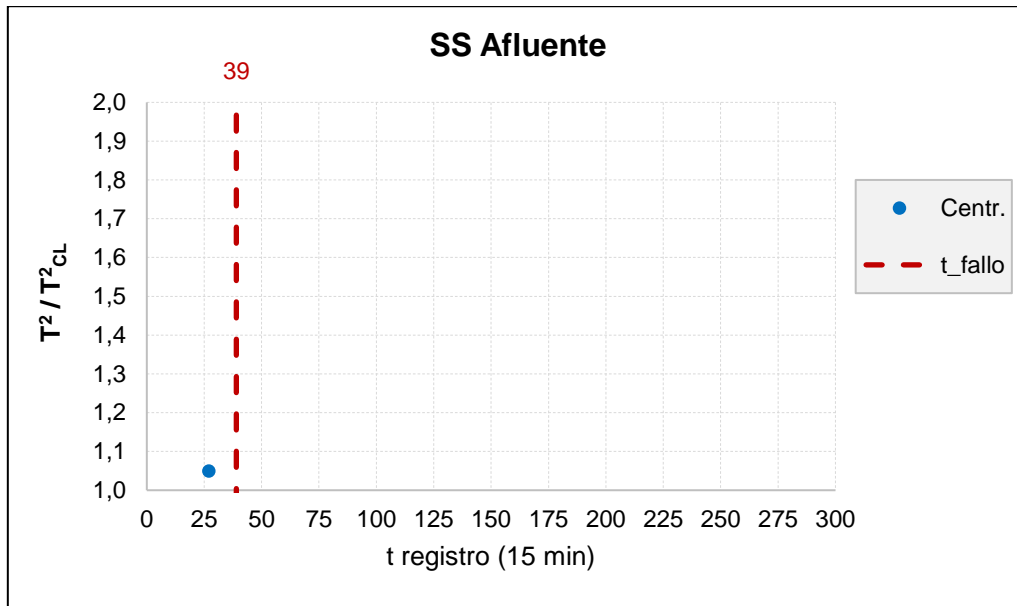


Gráfico 2. Control centralizado. Fallo 1

## **Fallo 2: SS reactor 1**

### ► Simulación:

El fallo 2 (sólidos en suspensión reactor 1) se observa gráficamente en el registro de control número 49, correspondiente al tiempo de simulación 12 h 15 min. La amplitud del fallo es del 42%.

### ► Sistema detección de fallos:

El sistema de control centralizado detecta el fallo introducido en los SS del reactor 1 con un retardo de 26 registros, es decir, 6 h 30 min, con un porcentaje de desvío de  $T^2$  respecto  $T_{CL}^2$  del 1%. En ese instante, la diferencia en la variable SS reactor 1 entre la situación con fallo y sin fallo es del 18%. No obstante, no puede considerarse que la detección del fallo sea eficaz dado que el siguiente fallo no se registrará hasta transcurrir 6 h 30 min adicionales.

Al igual que sucede con el fallo 1, se comprueba en el Gráfico 3 que se produce una alarma antes de producirse el fallo, aunque se considera aceptable y dentro de la tasa de falsas alarmas esperada.

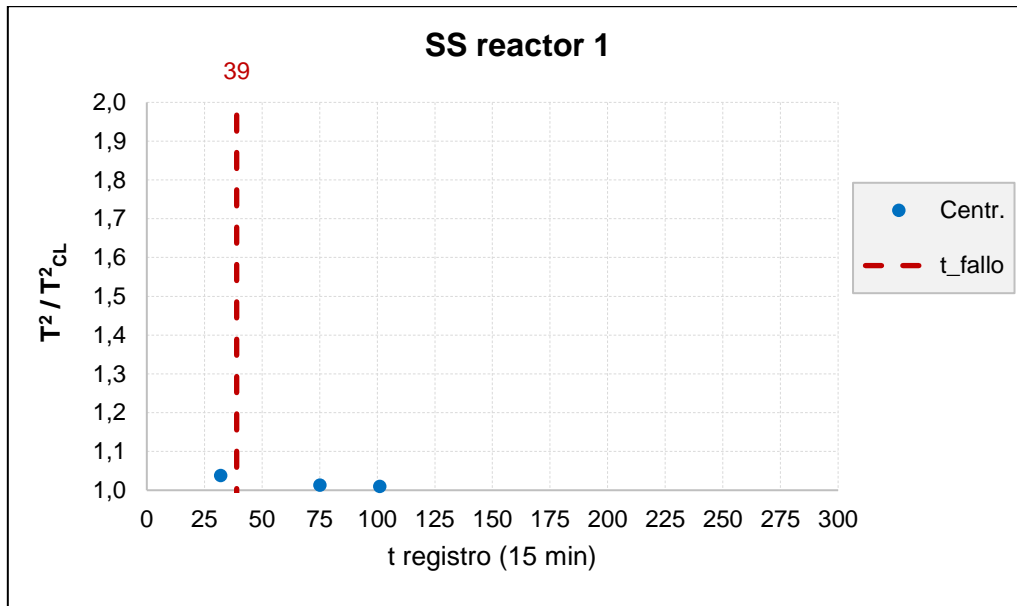


Gráfico 3. Control centralizado. Fallo 2

### **Fallo 3: Caudales Q<sub>r</sub> y Q<sub>w</sub>**

#### ► Simulación:

El fallo 3 (caudales espesador de lodos y tanque de deshidratación) se observa gráficamente en el registro de control número 192, correspondiente al tiempo de simulación 48 h (día 2), instante a partir del cual se reduce ambos caudales hasta 65 veces respecto la situación sin fallo.

#### ► Sistema detección de fallos:

En el caso del fallo 3, donde la magnitud del fallo es muy significativa y se produce de forma instantánea (salto escalón), el sistema es capaz de detectar el fallo con rapidez, a los en 45 min de producirse el fallo. En este instante, el valor por unidad de  $T^2$  es 1,20.

Tras esta primera alarma, se producen 48 alarmas adicionales, la siguiente a los 30 min desde la primera, y alcanzándose el máximo de  $T^2$  a las 12 h 30 min de producirse el fallo alcanzando un valor por unidad de 1,73.

En este caso no se ha registrado ninguna falsa alarma.

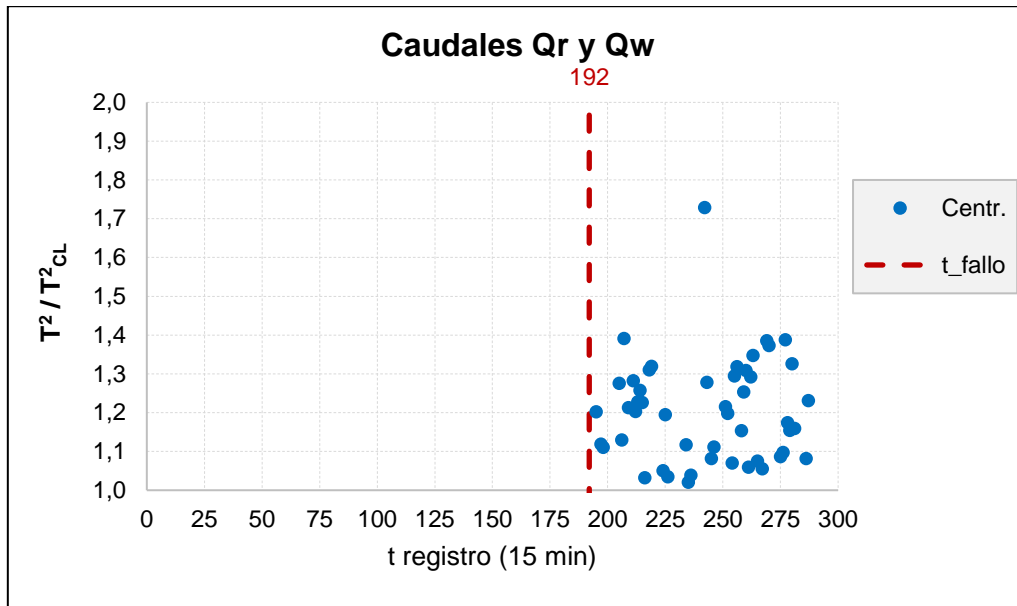


Gráfico 4. Control centralizado. Fallo 3

#### **Fallo 4: Caudales decantador primario**

► Simulación:

El fallo 4 (caudales decantador primario) se observa gráficamente en el registro de control número 58, correspondiente al tiempo de simulación 14 h 30 min. La amplitud del fallo es del 0,5%.

► Sistema detección de fallos:

Al igual que sucedía con el fallo 1, el sistema tampoco puede considerarse eficaz en la detección en el caso del fallo en los caudales del decantador primario ya que, si bien se registra 1 alarma de fallo, ésta no se produce hasta transcurrir más de 2 días.

En consecuencia, no puede asociarse esta alarma al fallo. El hecho de que no sea eficaz puede deberse a que la amplitud del fallo es de solo el 0,5%.

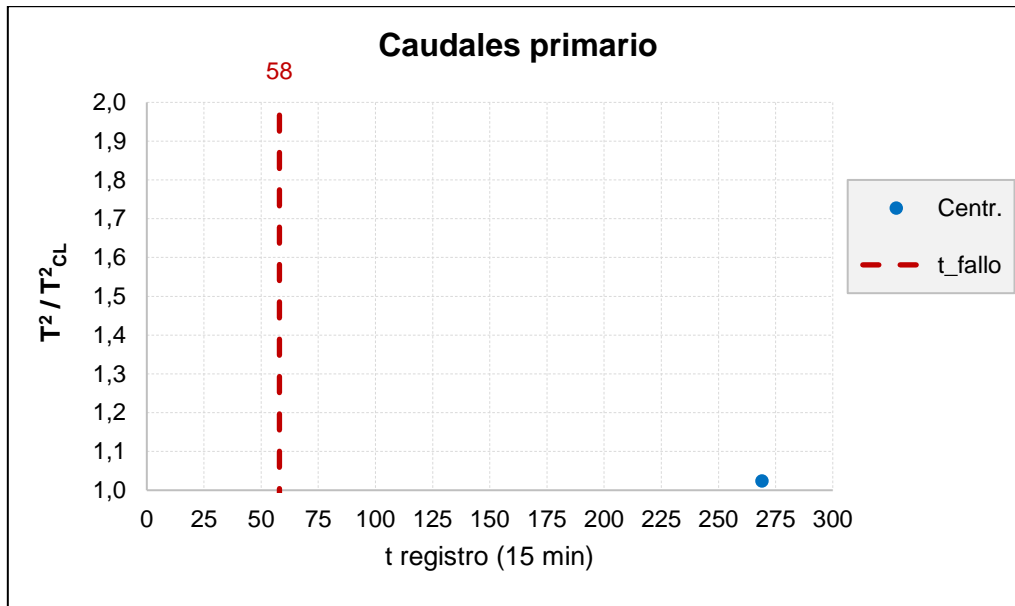


Gráfico 5. Control centralizado. Fallo 4

### **Fallo 5: Caudales tanque almacenamiento**

#### ► Simulación:

El fallo 5 (caudales de bypass y salida+bypass del tanque de almacenamiento) se introduce teóricamente en el registro de control número 71, correspondiente al tiempo de simulación 17 h 45 min, con una amplitud del fallo del 0,09%.

Sin embargo, esta amplitud no es suficiente como para que el fallo afecte al proceso por lo que este fichero, a efectos del sistema, debería comportarse igual que el fichero sin fallo.

#### ► Sistema detección de fallos:

Del estudio del fichero del fallo 5 cabría esperar que el sistema no detectase ningún fallo, o de hacerlo, son en casos puntuales dispares que puedan asociarse a falsas alarmas.

Es precisamente lo que se demuestra en el Gráfico 6, donde pueden observarse un total de 4 alarmas fallos, una de ellas antes de producirse teóricamente el fallo, y 3 adicionales tras producirse éste. No obstante, la primera alarma tras el fallo no se produce hasta haber transcurrido 17 h 15 min del proceso lo que demuestra que no puede asociarse al supuesto fallo del registro 71.

Habrà de transcurrir casi 3 h para producirse la siguiente alarma de fallo, por lo que no puede afirmarse que estas alarmas tengan relación al ser tan distantes en tiempo.

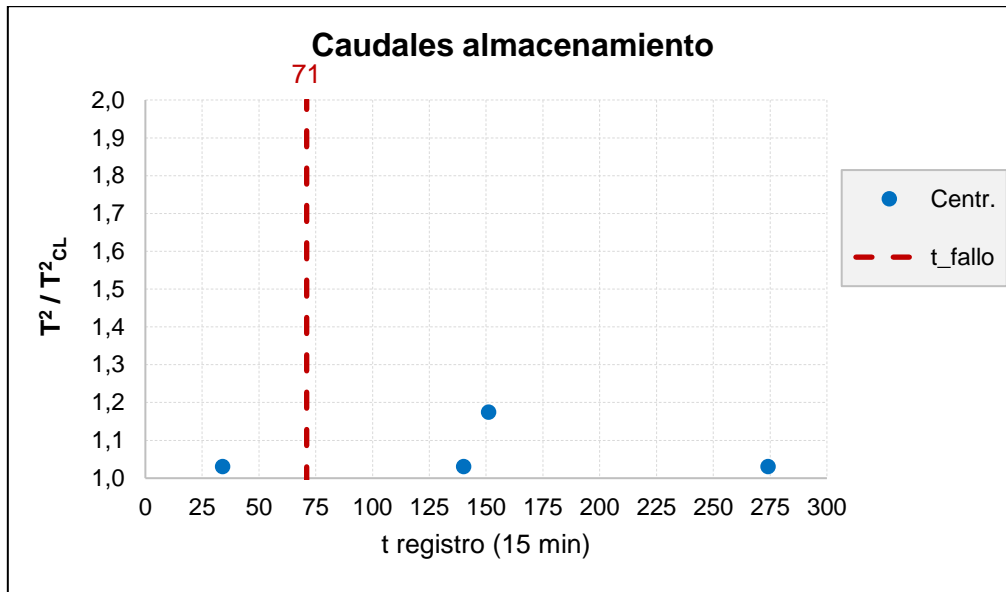


Gráfico 6. Control centralizado. Fallo 5

**Fallo 6: Kla**► Simulación:

El fallo 6 (coeficiente de transferencia de oxígeno) se observa gráficamente en el registro de control número 68, correspondiente al tiempo de simulación 17 h. La amplitud del fallo es del 2%.

► Sistema detección de fallos:

Una vez más se comprueba que el sistema centralizado no es eficaz en la detección de fallos de pequeña amplitud como es el fallo 6. En el Gráfico 7 se comprueba que no se registran alarmas de fallo tras comenzar el fallo 6 y solo se produce una falsa alarma en un instante de tiempo próximo a las falsas alarmas registradas en los casos de estudio de fallo 1, 2 y 4.

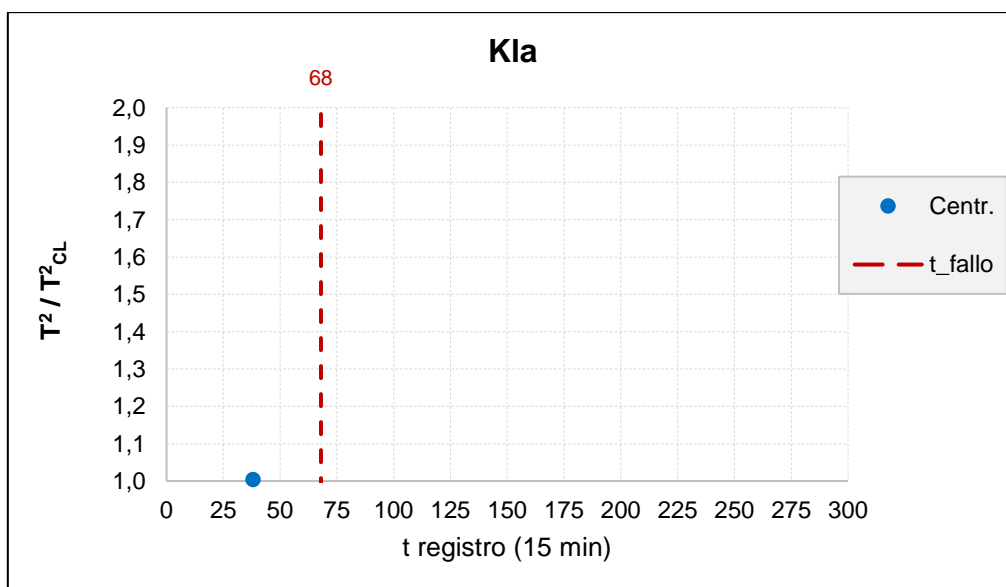


Gráfico 7. Control centralizado. Fallo 6

### **Fallo 7: O<sub>2</sub> disuelto**

► Simulación:

El fallo 7 (concentración de oxígeno disuelto) se observa gráficamente en el registro de control número 8, a las 2 h 30 min de simulación. La amplitud del fallo es del 30%.

► Sistema detección de fallos:

En el caso de estudio de anomalía en la concentración del O<sub>2</sub> disuelto (fallo 7) el sistema no detecta el fallo hasta transcurridas 18 h desde que comienza a producirse el fallo, con un desfase de  $T^2$  del 9,3% respecto el umbral. En este instante, la diferencia en la variable del O<sub>2</sub> disuelto en el reactor 1 entre la situación con fallo y sin fallo es del 78%.

La siguiente alarma no se produce hasta pasados 8 h 45 min, por lo que no puede relacionarse con la primera dada la diferencia temporal entre ambas y, en conclusión, no puede afirmarse que el sistema sea eficaz en la detección del fallo.

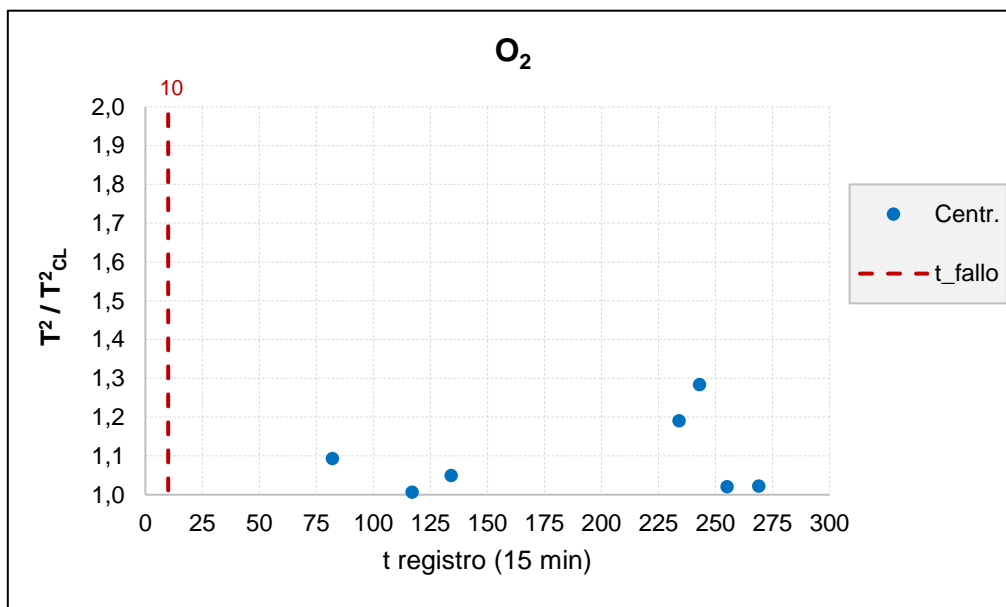


Gráfico 8. Control centralizado. Fallo 7

### **Fallo 8: Sales reactor 1**

► Simulación:

El fallo 8 (concentración de sales disueltas reactor 1) se observa gráficamente en el registro de control número 39, correspondiente al tiempo de simulación 9 h 45 min. La amplitud del fallo es del 4%.

► Sistema detección de fallos:

Finalmente, cabe destacar el caso de estudio del fallo 8 la cual puede asimilarse al fallo 3 analizado anteriormente. En ambos casos, se observa fácilmente en la gráfica de  $T^2/T_{CL}^2$  el instante a partir del cual el sistema comienza a detectar el fallo.



La diferencia radica en que para el caso particular del fallo 8, este instante se produce habiendo transcurrido 14 h 30 min desde que se introdujo el fallo, muy elevado para poder considerarlo detección temprana, con un porcentaje de desvío de  $T^2$  respecto el umbral en dicho registro de 26,9%.

Aunque el salto del fallo 8 también se produce de forma abrupta (escalón), continúa aumentando progresiva de manera que en el instante en el que se registra la primera alarma de fallo la diferencia entre la situación con fallo y sin fallo es del 72%.

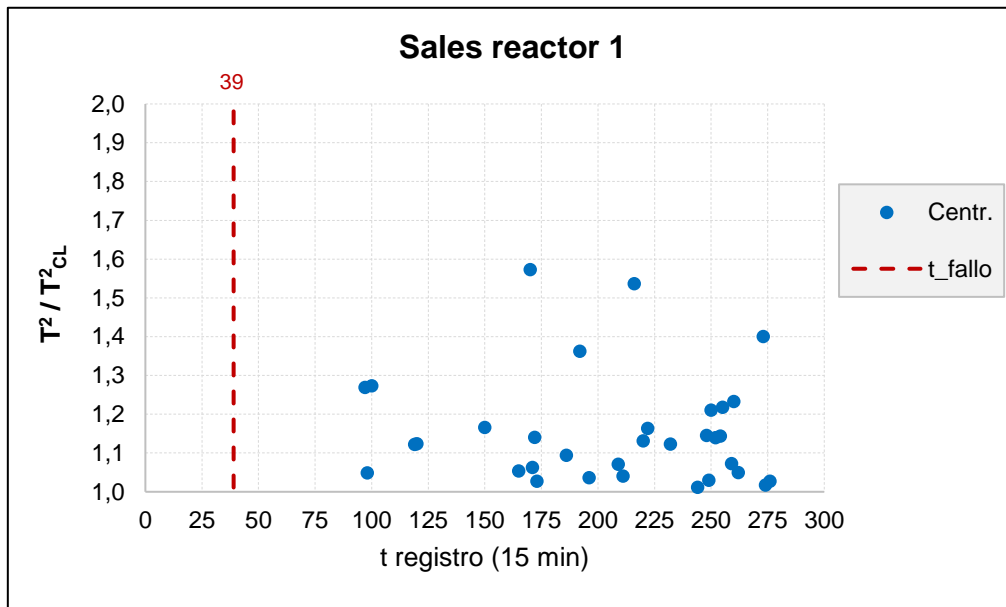


Gráfico 9. Control centralizado. Fallo 8

### 4.3.2. Diagnósis distribuida

#### 4.3.2.1 Proceso offline

#### A. Simulación 300 días

##### Asociación de variables mediante mRMR

La diagnósis distribuida se basa en la reducción de la dimensionalidad del proceso mediante la técnica mRMR, cuya aplicación se desarrolla en el **Algoritmo 8**.

Tras el descarte de las variables que permanecen constantes a lo largo del tiempo de muestreo, se procede a la aplicación de la técnica mRMR con objeto de distribuir en grupos las 127 variables restantes.

Como resultado, se obtiene un total de **9 agrupaciones** compuesta cada una de ellas por una variable principal y las variables que se correlacionan con la misma. Los grupos resultantes no son homogéneos en cuanto a su dimensionalidad, siendo el número total de variables por agrupación entre 2 y 62.

Se indica a continuación el número de variables que componen cada una de las agrupaciones:

Tabla 7. Grupos de variables resultantes de la técnica mRMR. Proceso offline

Descripción	Nº Variables	Descripción	Nº Variables
GRUPO 1	7	GRUPO 6	13
GRUPO 2	62	GRUPO 7	5
GRUPO 3	22	GRUPO 8	2
GRUPO 4	9	GRUPO 9	2
GRUPO 5	5		

Debido a que el cálculo de control estadístico  $T^2$  requiere una dimensionalidad mínima del espacio latente del VAE de 2 variables, no es posible aplicar el estudio a los grupos 8 y 9 al tener, ambos, 2 variables en su configuración de entrada lo que implica 1 característica en el espacio latente (reducción del 50%).

Así, se decide asociar estos dos grupos de variables en un único grupo (en adelante, "GRUPO 10") de modo que su tamaño en el espacio latente sea, al menos, de 2 variables y permita aplicar el método de control estadístico descrito.

En el Anexo D. Aplicación mRMR se recoge la relación de variables que componen cada uno de los grupos.

#### □ Entrenamiento iterativo y selección de VAEs

Para cada grupo normalizado de variables se sigue el mismo proceso de compilación, entrenamiento y test de los modelos VAE configurados con las distintas combinaciones de parámetros e hiperparámetros recogidos en la Tabla 3 (**Algoritmo 1**), generando un total 144 modelos de VAE por cada uno de los 8 grupos de variables.

En la ejecución del **Algoritmo 1** se observa una de las ventajas del método de control distribuido frente al centralizado: el tiempo de ejecución. Cada grupo de variables requiere menor tiempo de ejecución al ser inferior la dimensionalidad de la entrada al modelo. A modo de ejemplo, el grupo 2 – grupo con mayor número de variables (62) – requiere un total de 3 d 8 h 28 min 35,63 s para ejecutarse, lo que supone un 15% menos que en el método centralizado.

Si se compara con el grupo 1 (7 variables) la diferencia es lógicamente mayor, siendo el tiempo necesario para entrenar los 144 VAE un 87% inferior al centralizado.

No obstante, en términos globales no puede afirmarse lo anterior. Ejecutando de forma consecutiva cada grupo de variables, el tiempo total necesario es superior al caso descentralizado. Contabilizando de los grupos 1, 2 y 3, que suman 91 de las 127 variables (72%), el tiempo de ejecución asciende a 7 d 3 h 17 min 0,84 s, es decir, casi el doble que en el método centralizado.

En la siguiente tabla se recoge la parametrización seleccionada para el VAE de cada una de las 8 agrupaciones siguiendo el criterio establecido de RMSE de test mínimo. No se reflejan el valor de kernel (3x3) y stride (1) por no variar entre los distintos modelos generados.

Tabla 8. Parametrización modelo VAE descentralizado RMSE test mínimo, 300 días

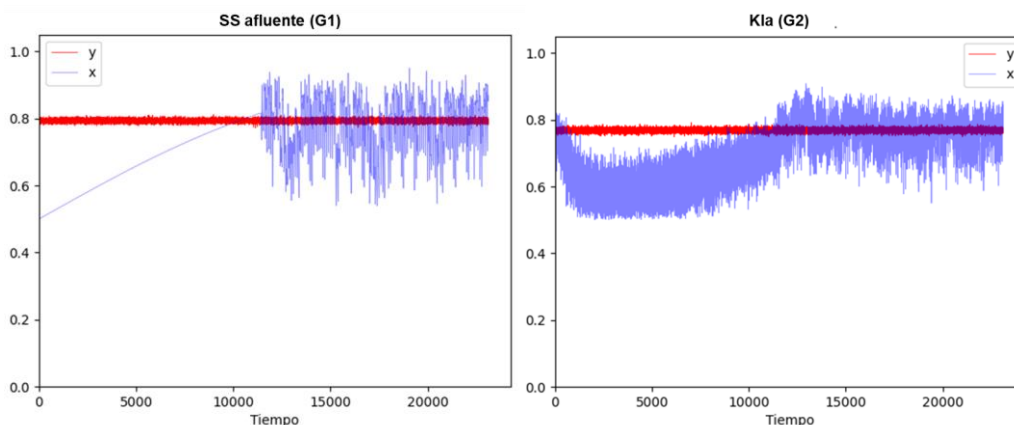
MÉTODO DESCENTRALIZADO – PARAMETRIZACIÓN VAE								
Grupo	Nº Var.	Frecuencia temporal	Tasa aprendizaje	Épocas	Filtros	Tamaño espacio latente	Batch	Error test
1	7	6	0,001	100	128	50%	32	$7,52 \cdot 10^{-2}$
2	62	6	0,0001	200	128	75%	32	$3,18 \cdot 10^{-2}$
3	22	6	0,001	200	64	50%	128	$4,01 \cdot 10^{-2}$
4	9	4	0,0001	200	128	75%	64	$2,21 \cdot 10^{-2}$
5	5	6	0,0001	200	32	75%	128	$7,00 \cdot 10^{-3}$
6	13	4	0,0001	200	64	75%	32	$3,78 \cdot 10^{-2}$
7	5	4	0,001	100	64	75%	32	$2,54 \cdot 10^{-2}$
10*	4	4	0,0001	200	32	75%	128	$2,54 \cdot 10^{-2}$

(\*) Combinación de los grupos 8 y 9

A la vista de los resultados, se demuestra que no es posible prefijar una configuración de parámetros e hiperparámetros válida para todas las agrupaciones de variables de la EDAR objeto de estudio. Esto implica que **el modelo de VAE no es escalable** entre los distintos grupos del proceso.

Cada modelo se configura con diferentes valores para obtener el RMSE de test mínimo de los calculados, cuyo valor promedio en el caso de estudio descentralizado es de  $3,01 \cdot 10^{-2}$ .

Para verificar el comportamiento de cada VAE seleccionado, se muestra en los siguientes gráficos alguna variable del proceso incluida en cada grupo, utilizando el fichero de entrenamiento y reflejando la entrada (azul) y salida (rojo) del VAE:



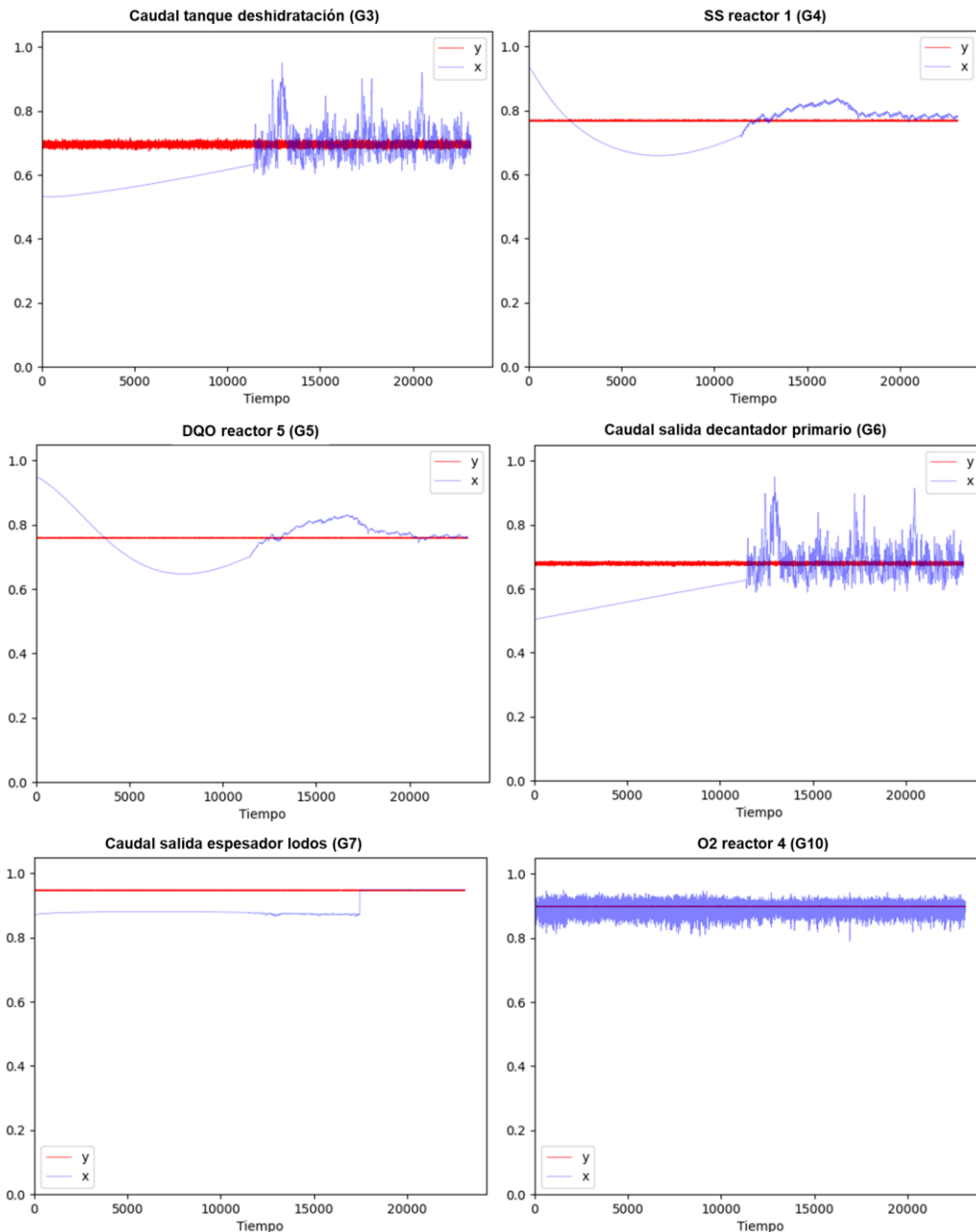


Figura 23. Validación entrenamiento VAEs distribuido, 30 días. Comparativa entrada y salida VAEs con datos entrenamiento (archivo sin fallos)

Los gráficos de la Figura 23 muestran un **comportamiento similar**, aunque en ocasiones con menos ruido, **que el obtenido en el sistema centralizado**. En aquellas variables cuyo perfil difiere en el transitorio y el estacionario, el patrón aprendido por el modelo es únicamente del estacionario.

Esto se debe a que, si bien la dimensionalidad de la entrada es menor, el modelo VAE sigue siendo el más básico posible, lo que limita el aprendizaje a los últimos datos entrenados. Al igual que en el caso anterior, sería necesario añadir capas intermedias adicionales y de mayor tamaño para mejorar el VAE y su capacidad de reconstrucción de la entrada.

Así, se opta por entrenar nuevamente los modelos, pero utilizando únicamente los **3 primeros días de simulación**, ajustándolo al mismo periodo del que se dispone en los ficheros de datos con fallo.

## B. Simulación 3 días

### □ Filtrado de datos y normalización

Tras el pretratamiento de los datos de entrada y la selección de las variables del grupo mRMR de estudio, se extraen los primeros 289 registros del fichero (3 días), se normalizan al rango [0;1] por normalización sigmoide.

### □ Entrenamiento iterativo y selección de VAE

A continuación, se procede al entrenamiento y test de los modelos VAE generados con las distintas combinaciones de parámetros e hiperparámetros recogidos en la Tabla 3, generando un total 144 modelos distintos de VAE (**Algoritmo 1**).

Comparando los métodos distribuidos con 300 y 3 días de registros, el tiempo de ejecución del programa del segundo se reduce significativamente al haberse reducido a una centésima parte la dimensionalidad de la entrada.

A modo de ejemplo, para el grupo 1 (7 variables) el tiempo necesario es 8 veces inferior, pasando a tan solo 1 h 35 min 5,21 s. En el caso de grupos con mayor número de variables la diferencia se acentúa aún más. En el caso de los grupos 2 y 3 (62 y 22 variables) el tiempo se reduce en casi 19 y 12 veces, respectivamente.

La combinación de parámetros e hiperparámetros seleccionada para cada grupo en el caso de estudio de 3 días de simulación se recoge en la siguiente tabla:

Tabla 9. Parametrización modelo VAE descentralizado RMSE test mínimo, 3 días

MÉTODO DESCENTRALIZADO – PARAMETRIZACIÓN VAE								
Grupo	Nº Var.	Frecuencia temporal	Tasa aprendizaje	Épocas	Filtros	Tamaño espacio latente	Batch	Error test
1	7	6	0,0001	200	32	50%	32	$1,07 \cdot 10^{-1}$
2	62	6	0,001	200	128	50%	32	$2,74 \cdot 10^{-2}$
3	22	6	0,001	100	64	75%	64	$2,97 \cdot 10^{-2}$
4	9	4	0,001	100	128	75%	32	$1,39 \cdot 10^{-2}$
5	5	6	0,001	200	64	75%	32	$4,50 \cdot 10^{-3}$
6	13	6	0,001	100	128	75%	32	$4,30 \cdot 10^{-2}$
7	5	6	0,001	200	64	75%	32	$2,35 \cdot 10^{-2}$
10*	4	4	0,001	200	128	75%	32	$2,15 \cdot 10^{-2}$

(\*) Combinación de los grupos 8 y 9

Con el nuevo entrenamiento se obtiene un RMSE de test promedio de  $3,12 \cdot 10^{-2}$ . En los siguientes gráficos se muestran las mismas variables que en la Figura 23 pero con el entrenamiento de 3 días:

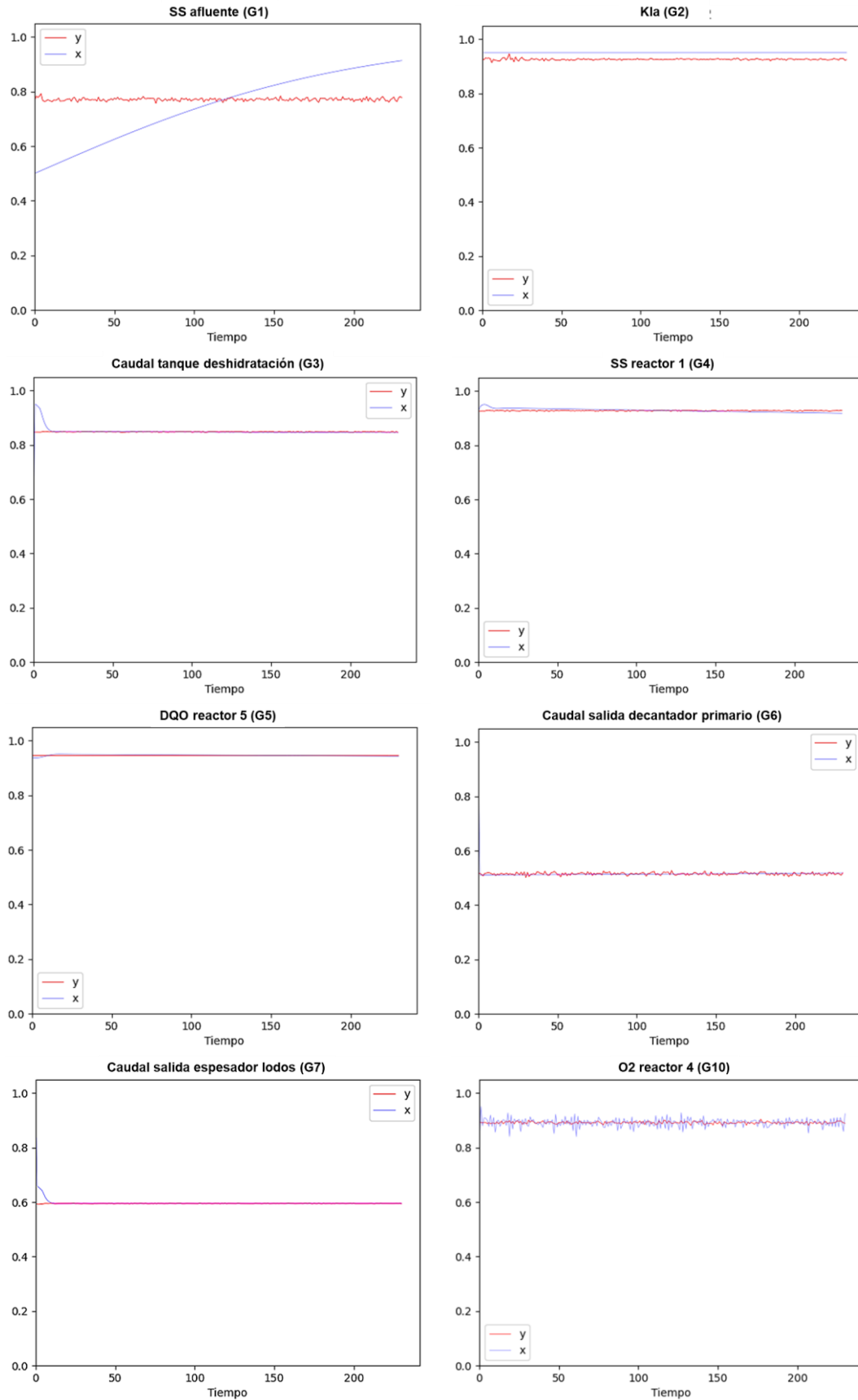


Figura 24. Validación entrenamiento VAEs distribuido, 3 días. Comparativa entrada y salida VAEs con datos entrenamiento (archivo sin fallos)

Poniendo el foco del entrenamiento en los 3 primeros días de simulación del proceso de la EDAR se observa cómo el modelo se ajusta con mayor precisión y menor ruido en los casos en los que la variable mantiene un comportamiento lineal mientras que en el caso que la complejidad del perfil es mayor, el VAE no es capaz de ajustar la salida del *decoder* a la entrada del *encoder*.

Esto se observa claramente en la variable de SS del afluyente, correspondiente al grupo G1 y cuyo error de test es superior al del resto de grupos estudiados (ver Tabla 9).

#### □ Determinación de $T_{CL}^2$

Una vez seleccionado el modelo VAE para cada uno de los grupos obtenidos tras la aplicación de la técnica mRMR, se procede a determinar el umbral de control estadístico  $T_{CL}^2$ , el valor de la media y la matriz de covarianza de las características del espacio latente, todo ello determinado con los datos de entrenamiento (sin fallo) utilizados para el estudio de los modelos VAE.

En la Tabla 10 se recoge el valor de  $T_{CL}^2$  para cada uno de los 8 grupos de variables calculado como el percentil 95 y 99 de los valores de  $T^2$ :

Tabla 10. Determinación umbrales de control estadístico método distribuido

Grupo	$T_{CL,95}^2$	$T_{CL,99}^2$
1	9,80	15,95
2	88,32	190,06
3	26,19	34,71
4	9,16	13,33
5	5,97	12,94
6	17,42	31,21
7	6,04	11,42
10*	7,50	10,26

En los siguientes gráficos se recoge el perfil de  $T^2$  que se obtiene al emplear para cada uno de los VAE el fichero de datos sin fallo, analizando el número de falsas alarmas con el mismo criterio de consideración de fallo de 2 desvíos de  $T^2$  consecutivos:

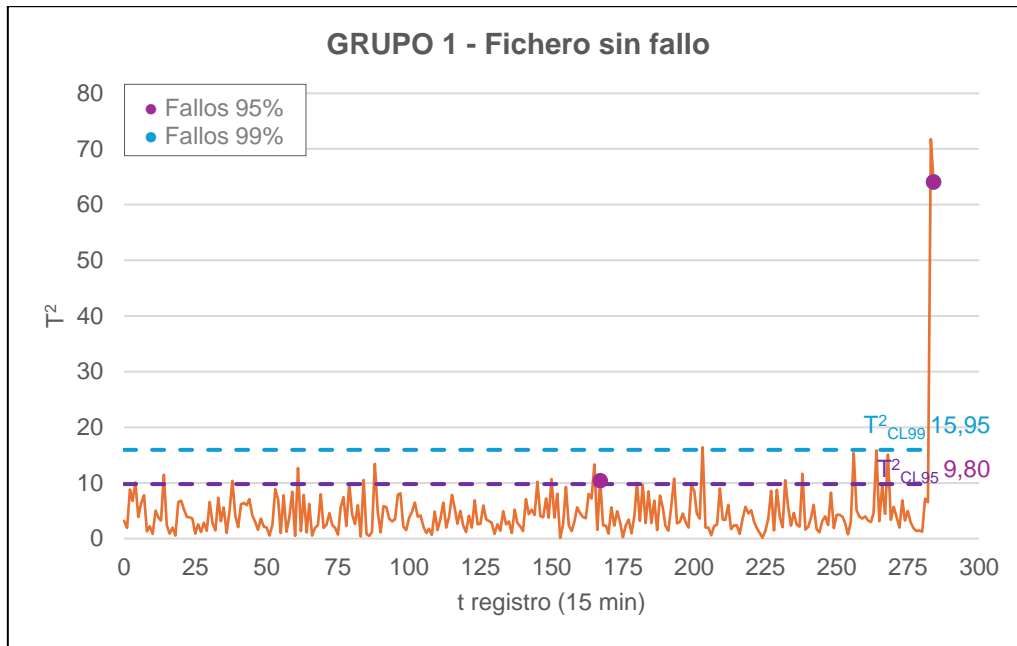


Gráfico 10. Umbral superior control  $T_{CL}^2$  método distribuido, grupo 1

En el caso del grupo G1, utilizando el umbral del percentil 99 tan solo se produce una falsa alarma en el registro  $t=284$ , coincidiendo con incremento abrupto de  $T^2$ . En el caso del percentil del 95, son 2 el número total de falsas alarmas, lo que supone una tasa del 0,70%, valor que se considera aceptable para el sistema desarrollado.

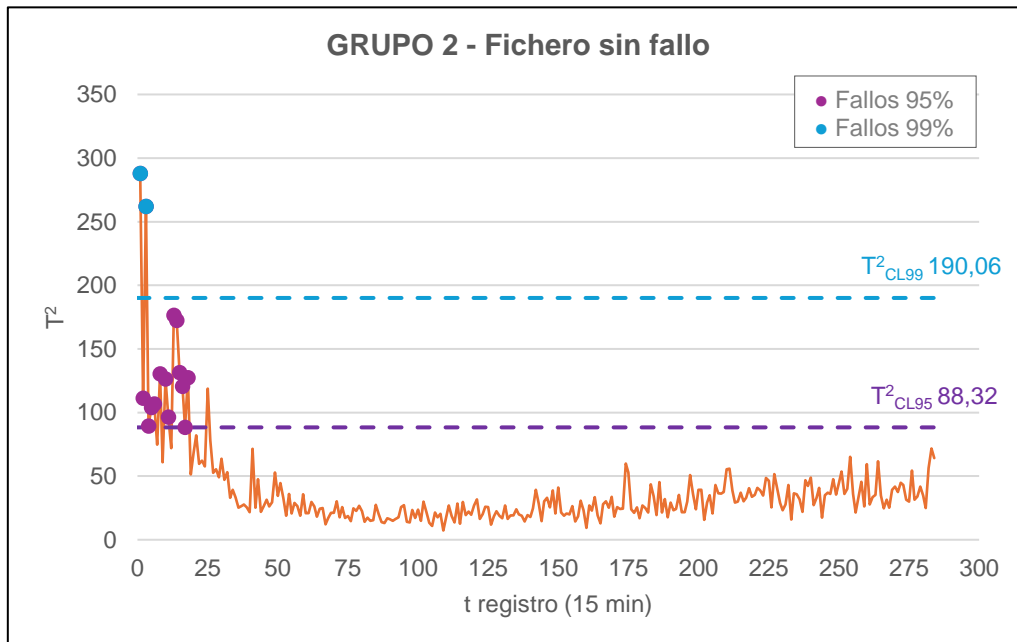


Gráfico 11. Umbral superior control  $T_{CL}^2$  método distribuido, grupo 2

El perfil de  $T^2$  del grupo G2 es un caso particular ya que hasta el registro 25 el valor de  $T^2$  supera el umbral reiteradamente, produciéndose hasta 15 falsas alarmas para el percentil 95 (2 con el percentil 99), lo que supone una tasa del 5,3%. Este detalle podrá suponer falsas alarmas en el proceso *online* en este periodo de tiempo.



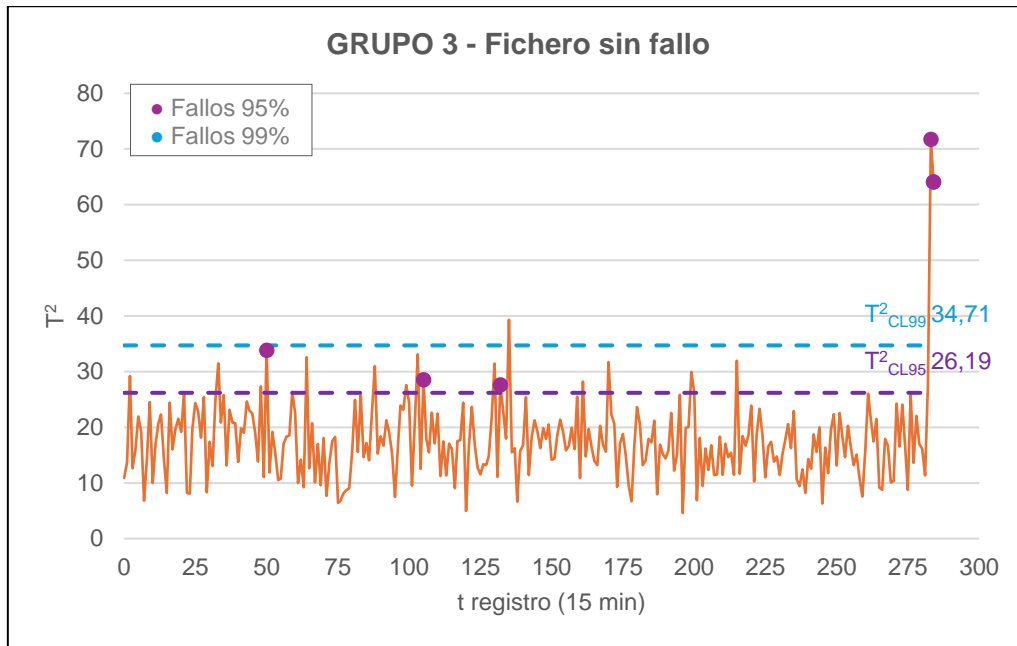


Gráfico 12. Umbral superior control  $T_{CL}^2$  método distribuido, grupo 3

En el caso del grupo G3, utilizando el umbral del percentil 99 se produce una falsa alarma en el registro  $t=284$ , coincidiendo con incremento abrupto de  $T^2$  observado anteriormente en el G1. En el caso del percentil del 95, se produce un total de 5 falsas alarmas, lo que supone una tasa del 1,8%.

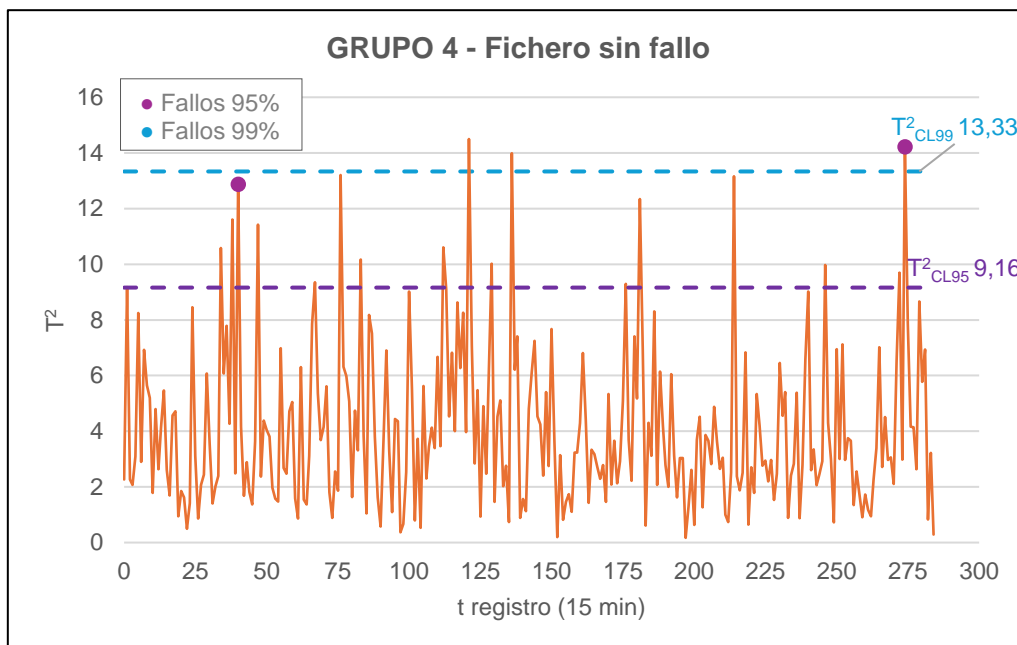


Gráfico 13. Umbral superior control  $T_{CL}^2$  método distribuido, grupo 4

En el caso del grupo G4, utilizando el umbral del percentil 99 no se registran falsas alarmas. Solo con el umbral del percentil 95 se producen 2 falsas alarmas (0,70%).

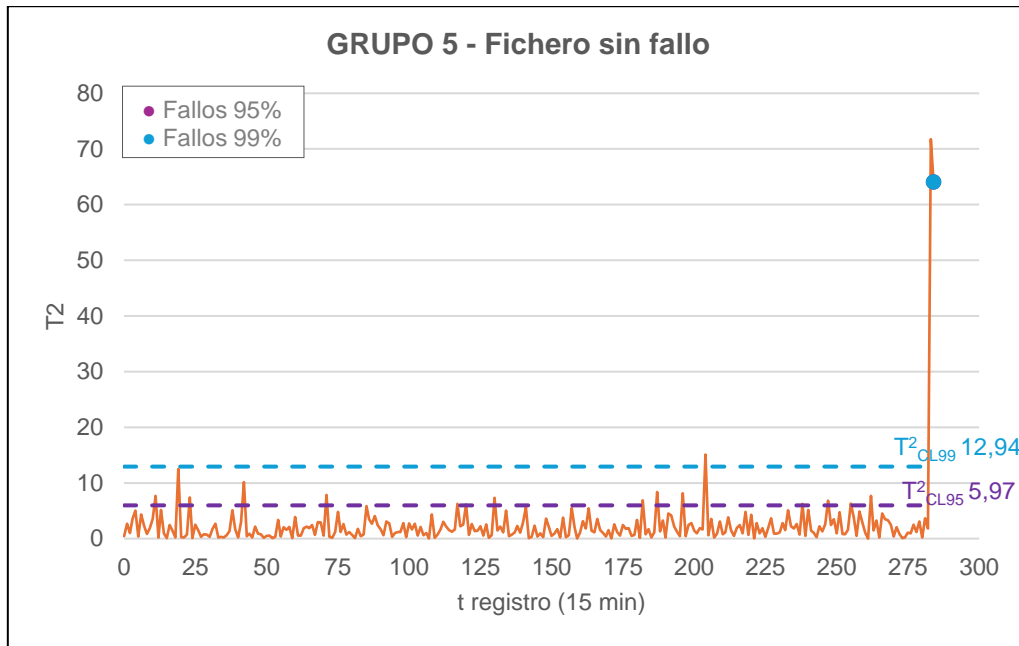


Gráfico 14. Umbral superior control  $T_{CL}^2$  método distribuido, grupo 5

Para el caso del grupo G5 se registra la misma falsa alarma tanto con el umbral 99 como con el 95 (0,35%) en el registro  $t=284$ , instante en el que se observa el mismo incremento abrupto de  $T^2$  que se remarca anteriormente en los grupos G1 y G3.

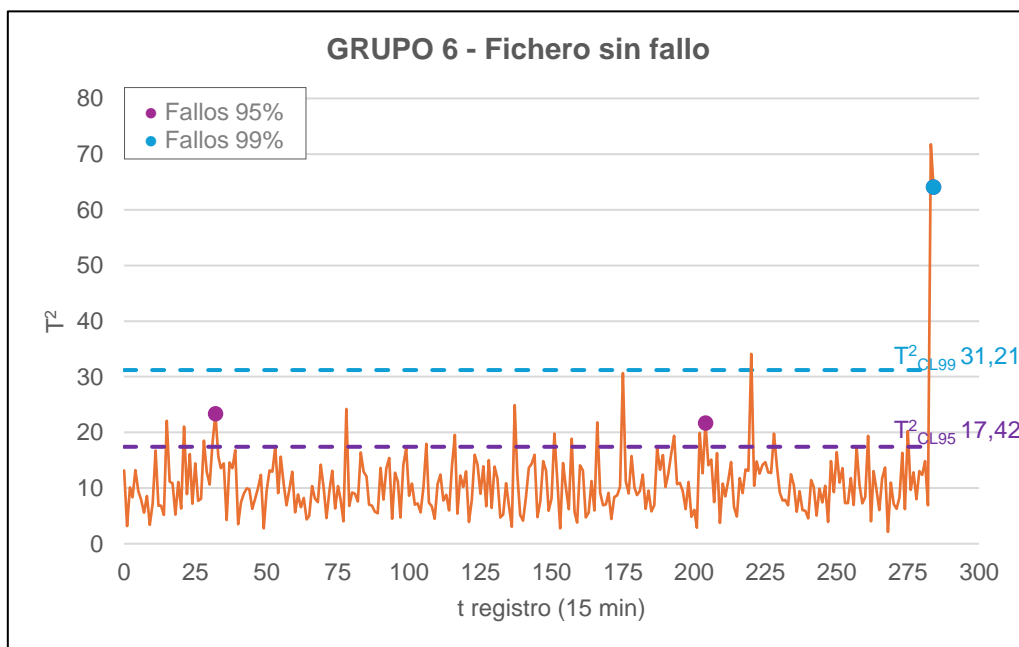


Gráfico 15. Umbral superior control  $T_{CL}^2$  método distribuido, grupo 6

En el caso del grupo G6, utilizando el umbral del percentil 99 se produce una falsa alarma en el registro  $t=284$ , coincidiendo con incremento abrupto de  $T^2$  indicado anteriormente. En el caso del percentil del 95, se produce un total de 3 falsas alarmas, lo que supone una tasa del 1,1%.

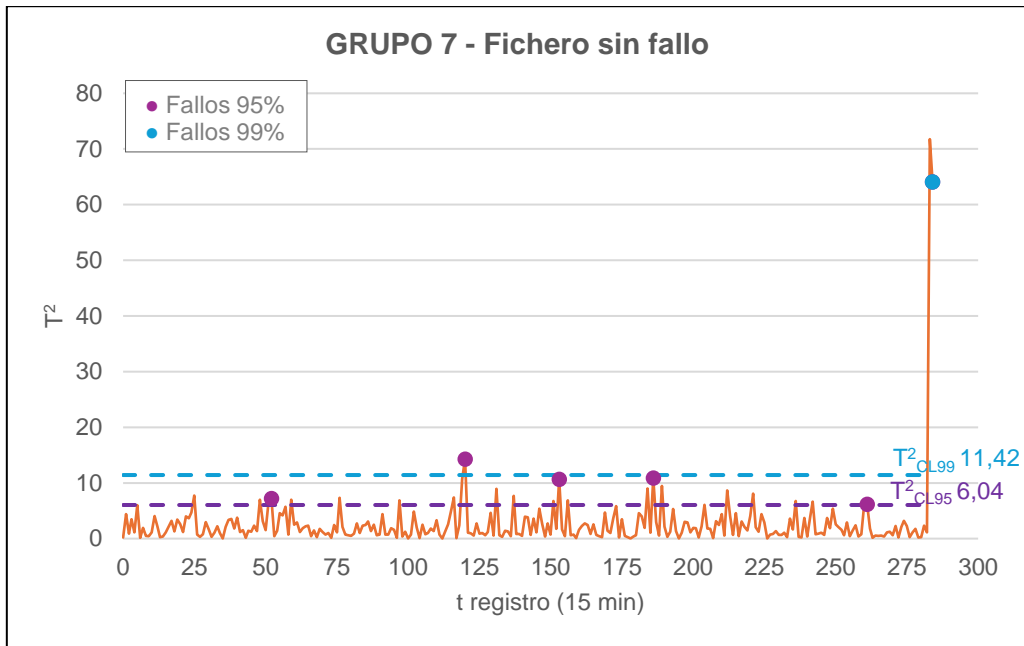


Gráfico 16. Umbral superior control  $T_{CL}^2$  método distribuido, grupo 7

En el grupo G7, al igual que en casos anteriores, utilizando el umbral del percentil 99 se produce una falsa alarma en el registro  $t=284$ . Por su parte, empleando el umbral del percentil del 95, se producen hasta 6 falsas alarmas (tasa del 2,1%).

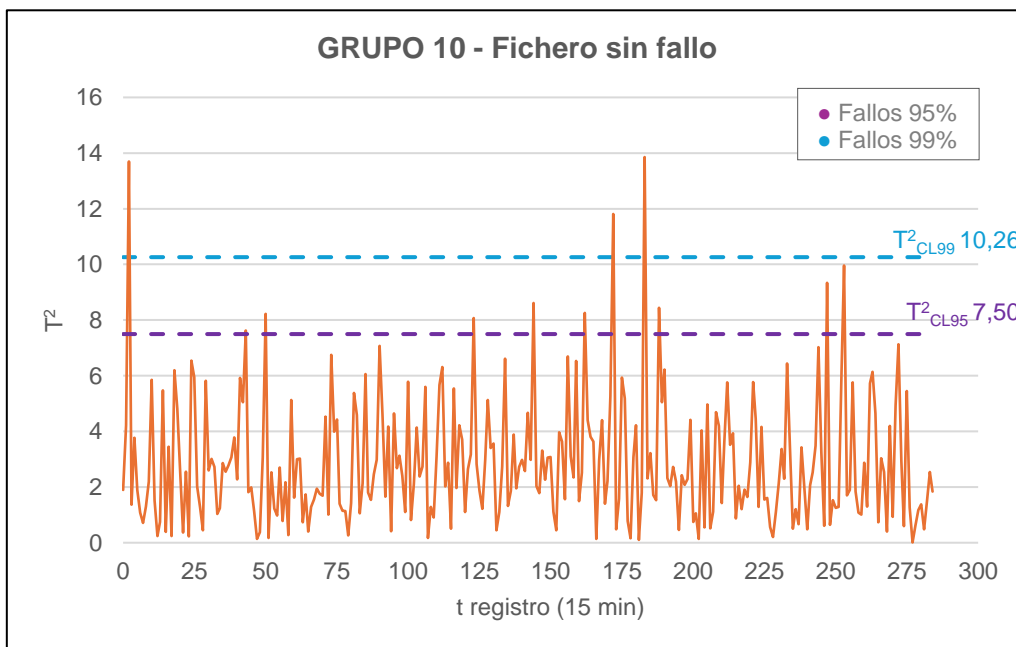


Gráfico 17. Umbral superior control  $T_{CL}^2$  método distribuido, grupo 10

Finalmente, para el grupo denominado G10, siguiendo el mismo procedimiento que en los casos anteriores no se producen falsas alarmas con el fichero sin fallos.

A la vista de los resultados se confirma que en el método distribuido también resulta ser el umbral del percentil 95 el preferible para el estudio del proceso *online*.

#### 4.3.2.2 Proceso online

Con objeto de analizar la eficacia y rapidez del sistema de detección de fallos desarrollado para el caso de estudio distribuido, se procede a aplicar el proceso *online* haciendo uso de los modelos VAE seleccionados en el proceso *offline*, así como los umbrales de control estadístico  $T_{CL}^2$  y los parámetros de media y matriz de covarianza necesarios para el cálculo del estadístico  $T^2$  de Hotelling.

Para poder hacer una comparativa posterior con el caso de estudio centralizado, analizan los mismos 8 ficheros que recogen los fallos antes señalados. Se analizarán los siguientes puntos:

- Falsas alarmas (fallos detectados antes del instante de fallo real)
- Tiempo registro primera detección de fallo tras instante de fallo real y magnitud  $T^2$  correspondiente.
- Grupo de variables que detecta el primer instante de fallo
- Tiempo de demora entre la primera y segunda alarma de fallo

Los gráficos que se muestran a continuación representan las alarmas de fallo registradas por cada modelo de VAE de los distintos grupos de variables. El eje de abscisas representa los registros de las variables de control (registros cada 15 min) y el eje de ordenadas representan el valor por unidad del estadístico  $T^2$ .

De esta forma es posible comparar tanto la rapidez con la que el sistema es capaz de detectar un fallo, la magnitud que supera el estadístico de control al umbral y el grupo de variables que lo detecta.

Se comprobará que los resultados difieren positivamente de los vistos en el caso centralizado, aunque el sistema tiene aún potencial de mejora ya que no todos los casos de fallo dan los mismos resultados.

Se recuerda que las alarmas del grupo G2 hasta el registro 25 pueden llevar a resultados erróneos dada la peculiaridad del perfil del estadístico  $T^2$  en este periodo para el dicho grupo de variables (ver Gráfico 11), motivo por el cual se representan en los gráficos siguientes de manera fácilmente distinguible.

#### **Fallo 1: SS afluyente**

##### ► Simulación:

El fallo 1 (sólidos en suspensión corriente entrada EDAR) se observa a partir del registro de control número 39, correspondiente al tiempo de simulación 9 h 45 min. La amplitud del fallo introducida es del 0,03%.

La variable SS afluyente se encuentra en el grupo de variables de mRMR G1 si bien este fallo se detecta en los SS de los flujos de salida del decantador primario (G1).

##### ► Sistema detección de fallos:

La primera alarma de fallo que emitiría el sistema de detección fallos distribuido para el fallo 1 se produce con una demora de 30 min (G1) y una desviación de  $T^2$  del 0,6%. La siguiente alarma de fallo se produce al transcurrir 1 h 45 min. En los siguientes

30 min se siguen registrando alarmas de fallo de magnitud más significativa y esta vez ocasionada por el G7.

En lo sucesivo se seguirán emitiendo alarmas de fallo, hasta un total de 28 alarmas, siendo los grupos que mayor número de fallos registran los G3, G6 y G7.

En el Gráfico 18 se comprueba que todos los grupos de variables detectarían el fallo 1, aunque con diferente intensidad y distinto retardo, aunque no todas estas alarmas pueden considerarse relevantes ya que algunas se producen con bastante diferencia temporal, pudiendo deberse a falsas alarmas propias del método estadístico.

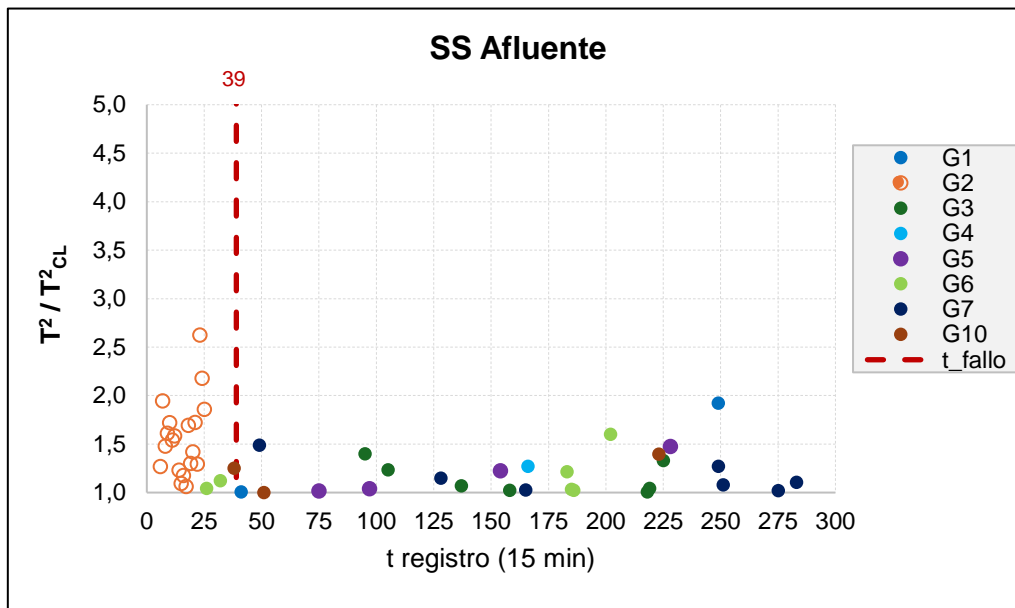


Gráfico 18. Control distribuido. Fallo 1

Por otra parte, es importante señalar que antes de producirse verdaderamente el fallo se registran hasta 3 falsas alarmas (sin contar con las falsas alarmas de G2).

En conclusión, se obtiene que, a diferencia del control centralizado, en el caso distribuido el sistema sí es capaz de detectar el fallo, con un retardo de 30 min.

## **Fallo 2: SS reactor 1**

### ► Simulación:

El fallo 2 (sólidos en suspensión reactor 1) se observa gráficamente en el registro de control número 49, correspondiente al tiempo de simulación 12 h 15 min. La amplitud del fallo es del 42%.

La variable SS reactor 1 se encuentra en el grupo de variables de mRMR G4.

### ► Sistema detección de fallos:

En el estudio del fallo 2, el resultado es que el sistema detecta la anomalía con 5 h 15 min de retardo (G6) y una desviación de  $T^2$  del 19,3% por encima del umbral. En la siguiente 1h 30 min se producen otras dos alarmas de fallo, una de ellas en el

siguiente registro de la primera y en el mismo grupo de variables, razón por la cual se consideran que estas alarmas pueden relacionarse con el fallo introducido.

Sin embargo, el resto de alarmas de fallo se producen con cierta distancia temporal entre ellas, lo que puede hacer dudar de si se trata de alarmas por fallo o por el carácter estadístico del método. Es importante destacar el comportamiento de  $T^2$  del G2 que, a partir del registro 221 supera el umbral de control hasta en 38 ocasiones alcanzando un máximo de  $T^2$  de 24,6 veces el umbral.

A diferencia del fallo 1, en esta ocasión no todos los grupos detectan la falla, siendo el G6 que más alarmas emite por detrás del G2.

En cuanto a las falsas alarmas, ignorando aquellas correspondientes al G2 antes del registro 25, se producen 2 registros de fallo antes de introducirse éste, lo cual se considera aceptable.

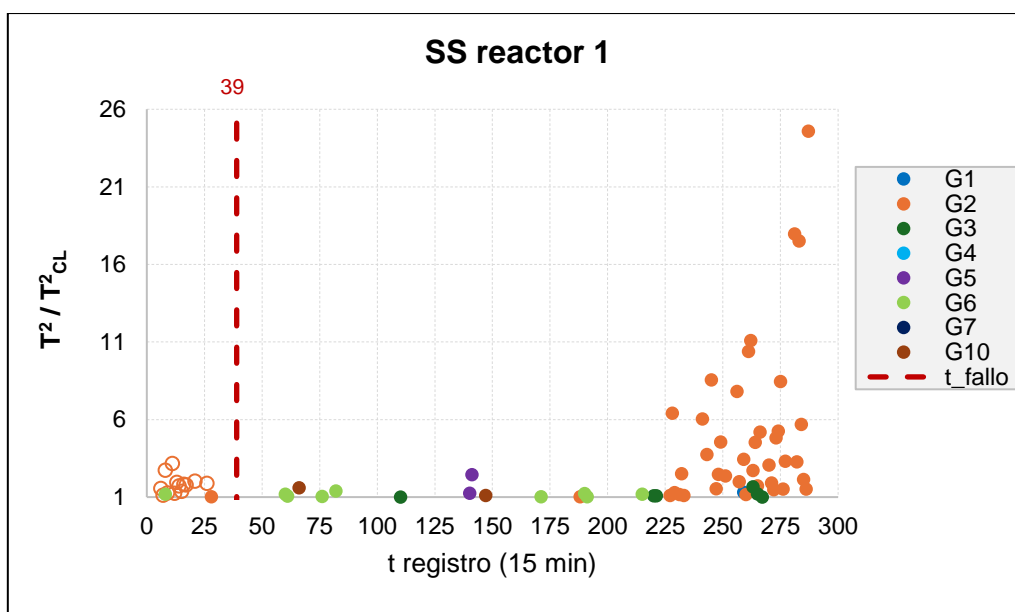


Gráfico 19. Control distribuido. Fallo 2

### **Fallo 3: Caudales Qr y Qw**

#### ► Simulación:

El fallo 3 (caudales espesador de lodos y tanque de deshidratación) se observa gráficamente en el registro de control número 192, correspondiente al tiempo de simulación 48 h (día 2), instante a partir del cual se reduce ambos caudales hasta 65 veces respecto la situación sin fallo.

Las variables correspondientes al fallo 3 se encuentran distribuidas en los grupos de variables de mRMR número 3, 6 y 7.

#### ► Sistema detección de fallos:

Al igual que sucedía en la monitorización centralizada, el sistema de monitorización y detección fallos distribuido es ciertamente eficaz con el caso de estudio del fallo 3. Gráficamente se observa que el fallo se detecta pasada 1 h desde que se produce el

fallo por los grupos G2 y G6. En este instante, el valor por unidad de  $T^2$  es 3,07 y 1,26 respectivamente.

Tras esta primera alarma, se producen 92 alarmas adicionales en el G2, todas ellas consecutivas, alcanzándose el máximo de  $T^2$  a las 20 h 45 min de producirse el fallo alcanzando un valor por unidad de 147,24.

En este caso se registra un total de 6 alarmas, ocasionadas por distintos grupos de variables y que, en consecuencia, se consideran dentro de la tasa de error esperada. Adicionalmente, se producen también alarmas de fallo por distintos grupos a partir del registro 192. No obstante, no de todas ellas puede asegurarse que se deban verdaderamente al fallo dada la dispersión entre alarmas del mismo grupo.

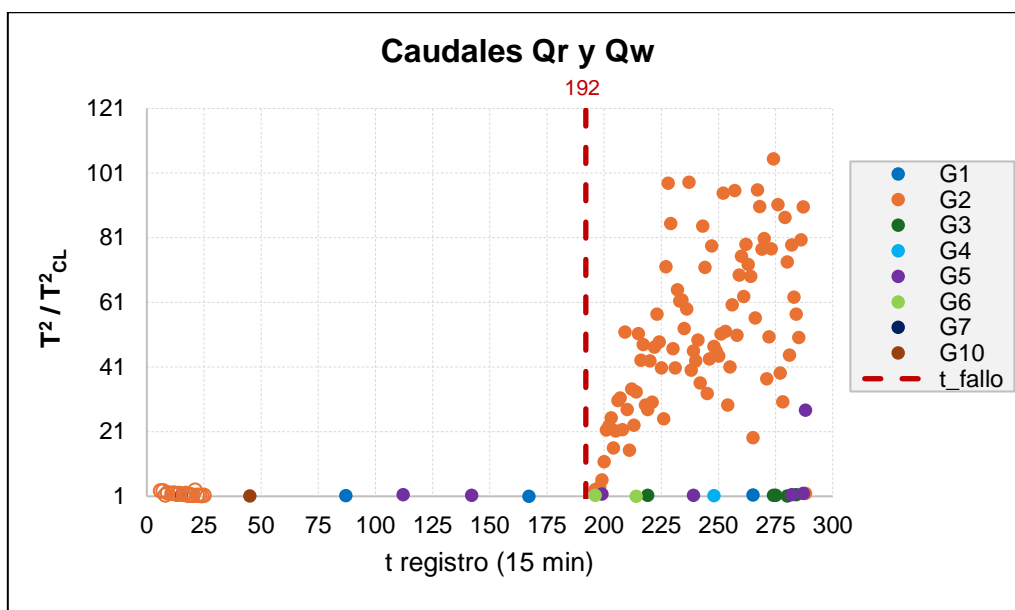


Gráfico 20. Control distribuido. Fallo 3

#### **Fallo 4: Caudales decantador primario**

##### ► Simulación:

El fallo 4 (caudales decantador primario) se observa gráficamente en el registro de control número 58, correspondiente al tiempo de simulación 14 h 30 min. La amplitud del fallo es del 0,5%.

Las variables relativas al caudal del decantador primario se encuentran en el grupo de variables de mRMR G6.

##### ► Sistema detección de fallos:

En el estudio de la eficacia del sistema con el fichero de fallo 4 se obtiene que la primera alarma de detección del fallo se produce con un retardo de 6 h 45 min (G3), con un valor de  $T^2$  de 1,02. La siguiente alarma no se produce hasta transcurrir 8 h 30 min, razón por la cual no puede considerarse que ambas alarmas estén relacionadas entre si y, por tanto, tampoco con el fallo introducido.

Las alarmas que se emiten posteriormente en el resto de grupos siguen un comportamiento similar al del G3, por lo que en esta ocasión se concluye que el sistema no es capaz de detectar el fallo 4. Esto puede deberse a que, al igual que en la monitorización centralizada, la amplitud del fallo no es suficiente como para que pueda ser detectada por el sistema mediante el estadístico  $T^2$ .

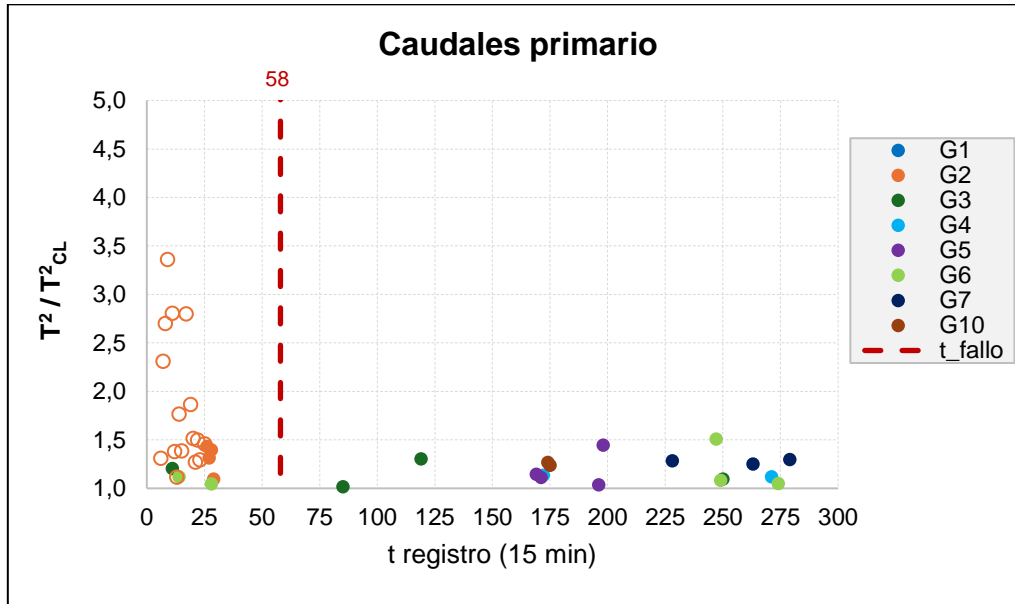


Gráfico 21. Control distribuido. Fallo 4

En cuanto a las falsas alarmas, en el Gráfico 21 se observan las mismas falsas alarmas del G2 antes del fallo que se detectan en otros casos de estudio, y otras 3 adicionales correspondientes a los grupos de variables 3 y 6. Si se considera que las alarmas posteriores al instante del fallo también son falsas alarmas, entonces estamos ante una tasa máxima del 7,4%, siendo el máximo por grupo del 1,8%.

### **Fallo 5: Caudales tanque almacenamiento**

#### ► Simulación:

El fallo 5 (caudales de bypass y salida+bypass del tanque de almacenamiento) se introduce teóricamente en el registro de control número 71, correspondiente al tiempo de simulación 17 h 45 min, con una amplitud del fallo del 0,09%. Sin embargo, esta amplitud no es suficiente como para que el fallo afecte al proceso por lo que este fichero a efectos del sistema debería comportarse igual que el fichero sin fallo.

Respecto a las variables caudal de bypass y caudal salida+bypass del tanque de almacenamiento, éstas se encuentran en el grupo de variables de mRMR G3.

#### ► Sistema detección de fallos:

La situación del caso del fallo 5 es similar a la observada en el fallo 4. Por un lado, si bien la primera alarma de fallo se produce tras 1 h 45 min desde producirse el fallo (G3), no puede asociarse al teórico fallo introducido en el sistema ya que tienen que pasar hasta 17 h para que se produzca la siguiente alarma.



Las siguientes alarmas que se producen en otros grupos de variables se comportan de modo similar, con una distancia temporal entre ellas tal que puedan considerarse como falsas alarmas propias del método estadístico utilizado (percentil 95).

Por tanto, el resultado del fallo 5 es el esperado y, al ser tan poco significativa la amplitud del fallo introducida, el sistema se comporta de forma similar a la situación del fichero sin fallo.

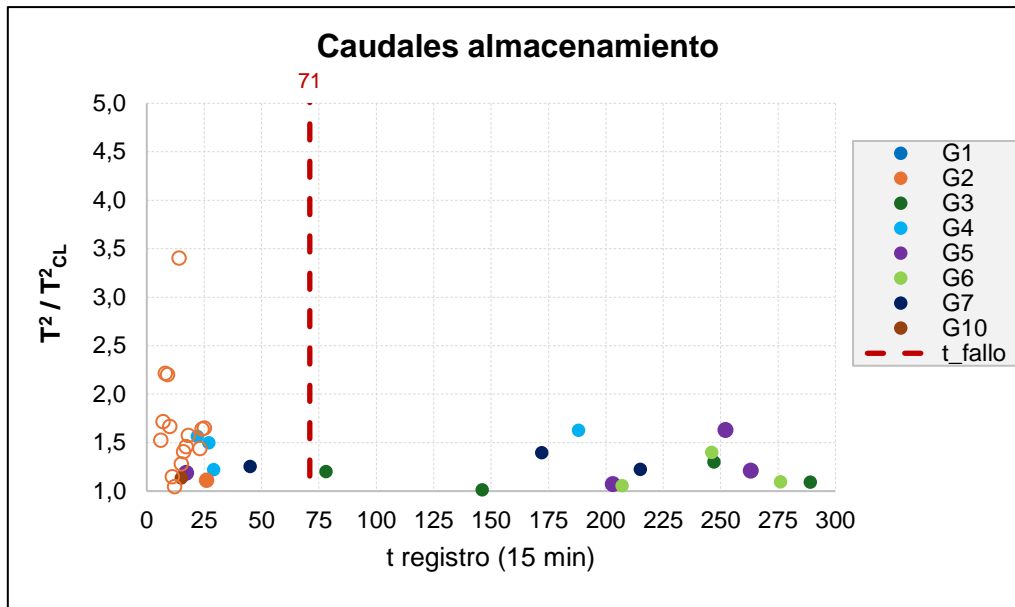


Gráfico 22. Control distribuido. Fallo 5

Por otra parte, en el Gráfico 18 se comprueba que antes del fallo se producen hasta 6 falsas alarmas sin contabilizar las propias del G2, lo cual se considera dentro de lo normal teniendo en cuenta que se producen en distintos grupos de variables.

### **Fallo 6: Kla**

#### ► Simulación:

El fallo 6 (coeficiente de transferencia de oxígeno) se observa gráficamente en el registro de control número 68, correspondiente al tiempo de simulación 17 h. La amplitud del fallo es del 2%.

La variable Kla se encuentra en el grupo de variables de mRMR G2.

#### ► Sistema detección de fallos:

En el caso del fallo en Kla, el comportamiento del sistema es similar al visto en los casos de fallo 4 y 5, donde el no puede considerarse eficaz en la detección de fallos de pequeña amplitud. Aunque la primera alarma de fallo se produce a los 30 min de introducirse el fallo, tendrán que pasar 11 h hasta la siguiente alarma. Así, éstas y las sucesivas alarmas no pueden asociarse al fallo dada la demora en que se producen.

Por tanto, se concluye que el sistema tanto en la monitorización centralizada como distribuida, no es capaz de detectar el fallo simulado en la variable de Kla, posiblemente debido a su pequeña amplitud.

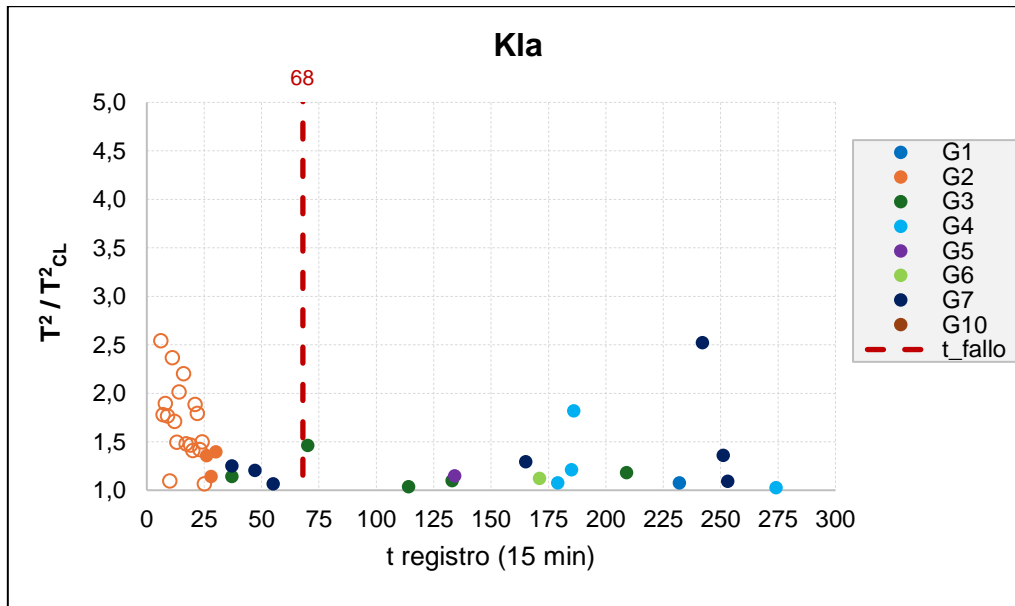


Gráfico 23. Control distribuido. Fallo 6

### **Fallo 7: O<sub>2</sub> disuelto**

#### ► Simulación:

El fallo 7 (concentración de oxígeno disuelto) se observa gráficamente en el registro de control número 10, a las 2 h 30 min de simulación. La amplitud del fallo es del 30%.

La perturbación en la variable O<sub>2</sub> se introduce en el O<sub>2</sub> de la corriente del decantador primario y reactor 1 las cuales se encuentran en los grupos de variables de mRMR número 2 y 3, respectivamente.

#### ► Sistema detección de fallos:

El caso del fallo en el O<sub>2</sub> disuelto no se puede evaluar dado que éste se produce precisamente en el periodo de tiempo donde se espera obtener las falsas alarmas del G2.

Además, en el Gráfico 24 se comprueba que el comportamiento del sistema vuelve a ser parecido al mostrado en los casos anteriores en los que se concluye que el sistema no es eficaz en la detección del fallo.

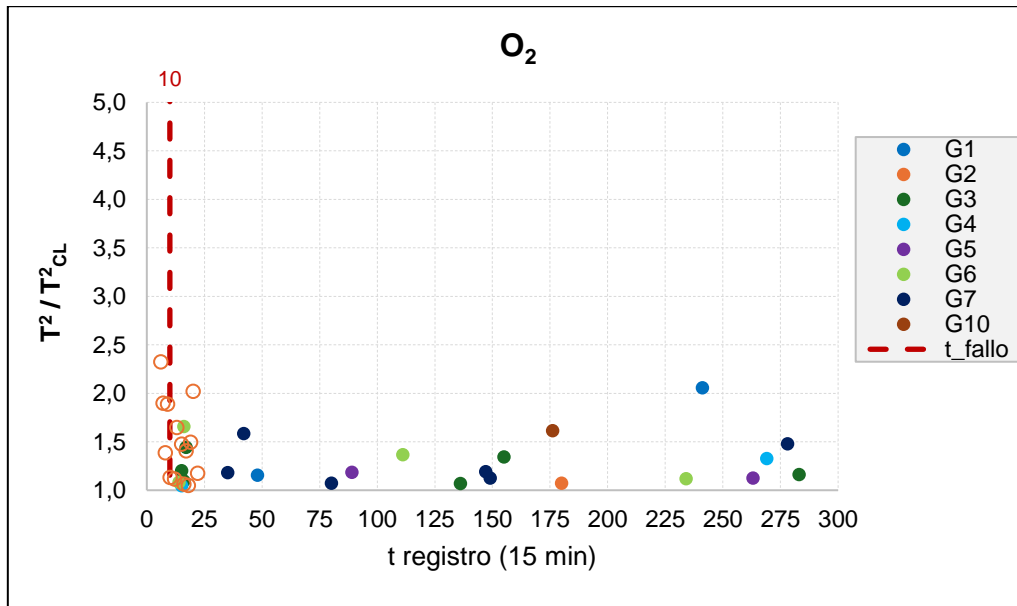


Gráfico 24. Control distribuido. Fallo 7

### **Fallo 8: Sales reactor 1**

#### ► Simulación:

El fallo 8 (concentración de sales disueltas reactor 1) se observa gráficamente en el registro de control número 39, correspondiente al tiempo de simulación 9 h 45 min. La amplitud del fallo es del 30%.

La variable sales reactor 1 se encuentra en el grupo de variables de mRMR G2.

#### ► Sistema detección de fallos:

Por último, el caso de estudio del fallo 8 en la diagnosis distribuida es muy similar al estudiado en la centralizada. En este caso, se observa claramente en la gráfica de  $T^2/T_{CL}^2$  el instante a partir del cual el sistema comienza a detectar el fallo.

Aunque el grupo G6 emite una alarma de fallo a 1 h 30 min desde que se introdujo el fallo, con un porcentaje de desvío de  $T^2$  respecto el umbral en dicho registro de 80,2%, el momento significativo de detección corresponde al G2 en el registro 65, es decir, tras 6 h 30 min de producirse el fallo. A partir de este instante, se emiten sucesivas alarmas de fallo, aumentando significativamente el desvío de  $T^2$  respecto el umbral hasta alcanzar un máximo de desvío del 4935%.

Además, en este caso de estudio no se producen alarmas antes del fallo, a excepción de las esperadas del G2.

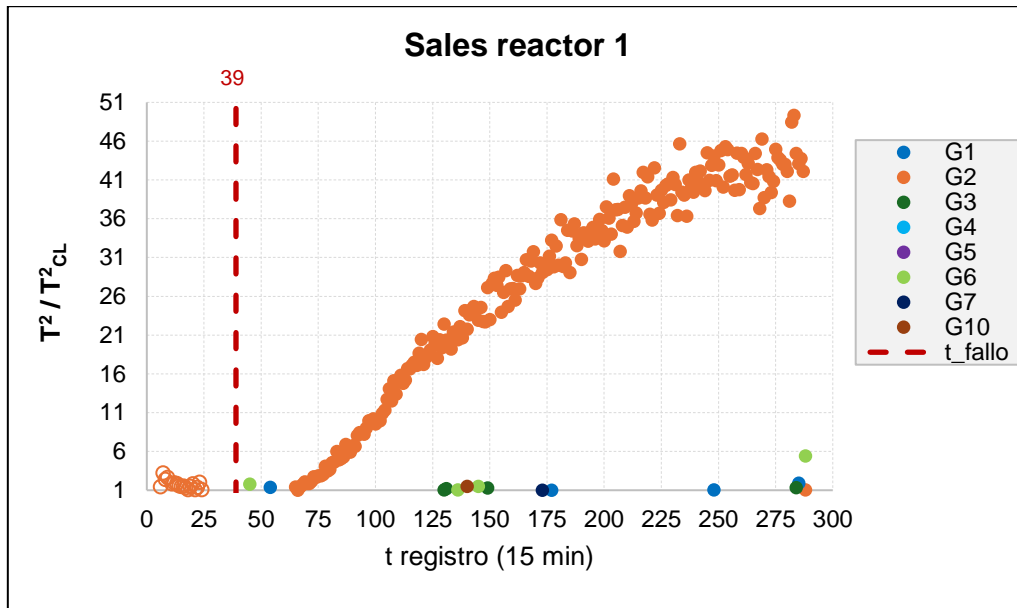


Gráfico 25. Control distribuido. Fallo 8

## 4.4 COMPARATIVA DE RESULTADOS

### 4.4.1. Proceso offline

En la aplicación del proceso *offline* del sistema de detección de fallos desarrollado a los métodos de control centralizado y distribuido se analiza, por un lado, el tiempo de ejecución requerido para el entrenamiento y selección de los VAE y, por otro, la relación entre la dimensionalidad de la entrada y el patrón aprendido por el modelo.

Respecto al **tiempo de ejecución**, como se ha indicado anteriormente, éste tiene una relación directa con el tamaño de la muestra de entrada al VAE. Una reducción del número de registros o del número de variables incide directamente en el tiempo de procesamiento requerido.

Así, utilizar registros de 3 días frente a 300 mejora significativamente el tiempo de ejecución de las distintas etapas del proceso *offline*, del orden de 9 veces en el caso centralizado y hasta 19 veces en los grupos con mayor número de variables del distribuido (Tabla 11).

Tabla 11. Comparativa tiempo de ejecución proceso *offline*

Método / Grupo	Nº Variables	Tiempo ejecución 300 días	Tiempo ejecución 3 días	Comparativa 300 d / 3 d
Centralizado	127	3 d 23 h 4 min 0,19 s	10 h 47 min 1,62 s	8,82
Distribuido / G1	7	12 h 35 min 7,85 s	1 h 35 min 5,21 s	7,94
Distribuido / G2	62	3 d 8 h 28 min 35,63 s	4 h 12 min 28,54 s	19,12
Distribuido / G3	22	3 d 6 h 13 min 17,36 s	6 h 46 min 45,88 s	11,54

Comparando el tiempo de ejecución requerido por ambos métodos, la ventaja del método distribuido es que cada grupo de variables requiere menor tiempo de ejecución al ser inferior la dimensionalidad de la entrada al modelo.

Aunque en términos globales es mayor el tiempo requerido por el método distribuido que el centralizado, esto sucedería únicamente en la primera ocasión que se defina el sistema para el proceso completo de la EDAR. A partir de entonces, el método distribuido presenta una mayor robustez pues el VAE correspondiente a cada grupo puede actualizarse y mejorarse de manera individual sin interferir en el resto del sistema de detección de fallos. Además, estas mejoras serán mucho más rápidas y requerirán menores recursos computacionales que si se actualizase el VAE del caso centralizado.

De este modo, en caso de cambios de equipos, reparaciones, cambios en las condiciones de operación de alguna parte del proceso u otro relacionado con la operación y mantenimiento de la planta, es mucho más ágil la readaptación del sistema de control en el caso distribuido. Este hecho resulta de gran interés en EDAR al ser sistemas cuyo perfil de funcionamiento puede variar con la estacionalidad y pueda requerir de constantes ajustes.

Por otra parte, en cuanto a la **dimensionalidad de la entrada y el patrón aprendido**, se ha observado que con tan solo 3 días de registros del funcionamiento en condiciones normales el modelo de VAE mínimo es capaz de aprender el perfil del proceso y detectar fallos en el proceso *online*.

Esto es especialmente relevante en modelos de aprendizaje profundo dado que se demuestra que no es necesario tener datos históricos de largos periodos de tiempo para ajustar el sistema de detección de fallos. Con solo 3 días de funcionamiento el sistema demuestra ser eficaz en la detección de anomalías de cierta magnitud, con mayor sensibilidad en el caso distribuido.

Así, volviendo a la operación y mantenimiento de una EDAR, se podrían reajustar los VAE del sistema de manera continua con pocos días de datos históricos del funcionamiento del proceso e implementar, en consecuencia, adaptaciones periódicas con un volumen de datos recientes relativamente pequeño.

Finalmente es importante destacar el **comportamiento del VAE mínimo** en el aprendizaje del proceso. Con 300 días de datos de registro de la EDAR, **el modelo conserva sólo el aprendizaje sobre los últimos datos introducidos en el VAE**. A priori puede suponer un problema ya que limita la complejidad del perfil capaz de reconstruir, pero supone también una ventaja en cuanto a las actualizaciones del proceso, recordando únicamente el modo de operación más actual de la EDAR.

#### 4.4.2. Proceso online

A la vista de los resultados del proceso *online* es posible afirmar que la aplicación del sistema de detección de fallos mediante un **control distribuido es más eficaz que el método centralizado** en el presente caso de estudio ya que es capaz de detectar mayor número de casos de fallo simulados.

Este hecho puede relacionarse con la técnica de control estadístico empleada. En la matriz de covarianza del método centralizado podría haber relaciones entre variables que oculten los fallos que se den en otras de menor peso en la matriz.

Para facilitar la comparativa de los resultados obtenidos en los casos que el sistema sí ha sido capaz de detectar el fallo, se resume en la Tabla 13 el instante en que se produce el fallo y su amplitud junto con el grupo que antes detecta dicho fallo (caso distribuido), la rapidez de la detección y el valor del estadístico  $T^2$  frente al umbral.

Tabla 12. Comparativa resultados detección de fallo. Fallos detectados

Fallo	Variable de fallo	Fallo (t)	Amplitud fallo	Grupo detección	Tiempo detección	$T^2/T_{CL}^2$ (p.u.)
1	SS Afluente	9 h 45 min	3%	Centr.	-	-
				G1	30 min	1,01
2	SS reactor 1	12 h 15 min	42%	Centr.	-	-
				G6	5 h 15 min	1,19
3	Caudales Qr y Qw	48 h 00 min	6500%	Centr.	45 min	1,20
				G2/G6	1 h 00 min	3,07/1,26
8	Sales reactor 1	9 h 45 min	30%	Centr.	14 h 30 min	1,27
				G6	1 h 30 min	1,80

Los casos más demostrativos de detección del fallo se dan con los fallos 3 y 8. En ambos casos, gráficamente se puede identificar el instante a partir del cual el sistema detecta el fallo y que se mantiene emitiendo sucesivas alarmas de fallo.

Por otra parte, los resultados demuestran que, en el caso de producirse un fallo en el proceso, el grupo de variables que primero detecta el fallo no refleja dónde se está produciendo el fallo. Se confirma así que el sistema desarrollado serviría para la detección, pero no para la identificación del fallo.

Respecto a la velocidad de detección del fallo, tanto el método distribuido como el centralizado **detectan el fallo en 1h o menos**, a excepción del fallo 8 donde la diferencia es mucho mayor siendo la detección más temprana en el caso distribuido (1 h 30 min) que en el centralizado (14 h 30 min). Así, y particularmente en la diagnosis distribuida, el tiempo de detección se considera aceptable ya que en algunos casos tan solo se necesitan 2 registros desde que se produce el fallo para que éste sea detectado por el sistema.

En cuanto a la amplitud del desvío de  $T^2$  frente a  $T_{CL}^2$ , se observa cierta relación entre la amplitud del fallo y la magnitud de desvío del estadístico en el primer instante de detección del fallo. Por lo general, a mayor amplitud de fallo, mayor es la proporción en que de  $T^2$  supera el umbral.

No obstante, **ambos métodos muestran no ser eficaces en la detección de fallos de pequeña amplitud**, como ocurre con los fallos 4, 5 y 6. Tampoco resultan eficaces en la detección del fallo del  $O_2$  disuelto, lo cual puede deberse al peso que tenga esta variable en la matriz de covarianza de los distintos grupos en los que se encuentra.

Tabla 13. Comparativa resultados detección de fallo. Fallos no detectados

Fallo	Variable de fallo	Fallo (t)	Amplitud fallo	Grupo detección	Tiempo detección	$T^2/T_{CL}^2$ (p.u.)
4	Caudales primario	14 h 30 min	0,5%	Centr.	-	-
				-	-	-
5	Caudales almacenamiento	17 h 45 min	0,09%	Centr.	-	-
				-	-	-
6	Kla	17 h 00 min	2%	Centr.	-	-
				-	-	-
7	O <sub>2</sub>	2 h 30 min	30%	Centr.	-	-
				-	-	-

Para solventar este problema habría que centrarse en la mejora del VAE, incrementando el número de capas intermedias en *encoder* y *decoder* y el tamaño de las mismas. De este modo, el modelo, más complejo, podría reconstruir el perfil del proceso con un ajuste más preciso permitiendo posteriormente la detección de fallos de menor amplitud. Sin embargo, teniendo como referencia el caso distribuido de este estudio, el incremento de la sensibilidad de detección de fallo también podría ocasionar un mayor número de falsas alarmas.

En la Tabla 14 se recoge el número de alarmas emitidas antes del fallo por el método centralizado y distribuido en cada caso estudiado, así como la tasa de falsas alarmas que supone. En los casos del método distribuido también se incluyen las falsas alarmas correspondiente al G2.

Tabla 14. Análisis resultados detección de fallo en proceso online. Método distribuido

Fallo	Variable de fallo	Fallo (t)	Amplitud fallo	Alarmas antes fallo	Tasa de falsas alarmas
1	SS Afluyente	9 h 45 min	3%	1	0,35%
				22	7,61%
2	SS reactor 1	12 h 15 min	42%	1	0,35%
				15	5,19%
3	Caudales Qr y Qw	48 h 00 min	6500%	-	-
				24	8,30%
4	Caudales primario	14 h 30 min	0,5%	1	0,35%
				23	7,96%
5	Caudales almacenamiento	17 h 45 min	0,09%	1	0,35%
				22	7,61%
6	Kla	17 h 00 min	2%	1	0,35%
				8	2,77%
7	O <sub>2</sub>	2 h 30 min	30%	1	0,35%
				5	1,73%
8	Sales reactor 1	9 h 45 min	30%	-	-
				18	6,23%

En el caso centralizado a penas se producen falsas alarmas o de producirse son irrelevantes para el sistema (tasa inferior al 1%). En el caso del control distribuido el valor de la tasa de falsas alarmas es superior, aunque siempre por debajo del 10%.

Esto se debe principalmente al perfil de  $T^2$  del grupo G2, donde se ocasionan numerosas falsas alarmas en los primeros 25 registros. Este hecho implica que no sea evaluable el fallo 7 en el método distribuido, el cual se produce dentro de este periodo temporal de falsas alarmas.

No obstante, la tasa de falsas alarmas se reduciría significativamente en caso de ignorarse las propias del G2 en ese primer periodo de tiempo o mejorando el VAE de dicho grupo haciéndolo más complejo y pudiendo ajustarse mejor al perfil de las variables del proceso propias de dicho grupo. Teniendo en cuenta lo anterior, se concluye que **la tasa de falsas alarmas es aceptable en ambos métodos ya que, aun contabilizando las del G2, es siempre inferior al 10%.**



## 5. ESTUDIO ECONÓMICO

---

El estudio económico que se desarrolla en este capítulo se trata de un análisis de los recursos que han sido utilizados para el desarrollo de este Trabajo.

Únicamente se considerarán los costes directamente asignables al desarrollo de este proyecto, es decir, derivados de los recursos humanos y materiales utilizados, o lo que es lo mismo, relativos a la mano de obra y la amortización de equipos.

Este análisis económico permite conocer el coste de desarrollo del proyecto como estudio preliminar a lo que sería un encargo del mismo por un cliente que desee desarrollar la aplicación para la mejora del control de proceso de sus propias instalaciones. Así, en este estudio solo se considerarán como costes indirectos los suministros de electricidad, calefacción e internet, no considerando otros costes que hubieran sido propios de una empresa tales como gastos de administración, marketing, alquiler, etc.

Para calcular el coste total del desarrollo del trabajo, se describirán las fases que han sido llevadas a cabo, desde la comprensión y conocimiento de las herramientas y técnicas a emplear hasta el análisis de resultados y finalización del proyecto.

### 5.1 FASES DEL PROYECTO

El proyecto ha sido desarrollado en 6 etapas de trabajo secuenciales, que incluyen desde la fase de análisis y comprensión del problema hasta la aplicación de la metodología desarrollada para su evaluación en la detección de fallos.

Es importante señalar que, si bien las etapas que a continuación se describen son extrapolables al desarrollo de la metodología propuesta en una aplicación industrial diferente, no es posible afirmar que la duración y coste del proyecto sea el mismo pues se trata de una metodología y un modelo específico para la aplicación propuesta, cuya complejidad depende en gran medida de la labor de investigación y documentación de referencia disponible.

#### ► **Fase 1. Análisis de la propuesta**

La primera etapa consiste en el análisis y comprensión de las necesidades del proyecto. Se definen el alcance, los objetivos, técnicas y herramientas a utilizar para el desarrollo del trabajo.

Se establece la planificación de los trabajos y las siguientes fases a desarrollar. En el caso de una aplicación futura, se definirían en esta fase el equipo del proyecto, los tiempos de ejecución y recursos consumidos estimados para cada actividad.

#### ► **Fase 2. Investigación y documentación**

Una vez definido el alcance y los objetivos del proyecto, se realiza una búsqueda y análisis de documentación relacionada con la aplicación de redes neuronales y, en particular, AE y VAE, en procesos de la misma tipología o similar al caso de estudio, que en este proyecto corresponde con una EDAR.

Esta fase será clave tanto para el desarrollo del modelo como para la evaluación de los resultados.

### ► **Fase 3. Diseño de la herramienta**

Como paso previo al diseño del modelo, se han empleado recursos en ampliación del conocimiento del lenguaje de programación y sus librerías en la aplicación de las técnicas concretas del proyecto, es decir, mRMR y modelos VAEs.

A continuación, se describe a alto nivel o arquitectónico la estructura del programa, definiendo los distintos módulos que lo compondrán y la relación entre ellos.

Se describe en esta etapa el diseño de la metodología de la herramienta en pseudocódigo y se acotan los parámetros e hiperparámetros de entrada a evaluar para la búsqueda del modelo más ajustado.

### ► **Fase 4. Programación**

Una vez planteada la arquitectura de la herramienta, se procede al desarrollo del programa del sistema de detección de fallos propuesto.

El desarrollo del programa se realiza de forma genérica, de modo que pueda ser extrapolado a múltiples bases de datos. En el caso concreto de este trabajo, el código inicial fue desarrollado y probado para la base de datos TEP (*Tennessee Eastman plant*) y posteriormente se adaptó a la base de datos de la EDAR simulada.

El desarrollo del programa se realiza por fases. De este modo, se testean los diferentes bloques a medida que se van desarrollando, comprobándose que los componentes y módulos funcionan y se interrelacionan de forma correcta y en cumplimiento de los requisitos. Esto ha permitido localizar y corregir los defectos e ineficiencias del programa paulatinamente antes de tener el programa completo.

Además, esta fase de desarrollo y testeo del programa ha permitido optimizar los recursos computacionales y minimizar retrasos en las fases siguientes.

### ► **Fase 5. Implantación de la metodología**

A continuación, se procede a aplicar la metodología desarrollada a la fuente de datos objeto del proyecto (EDAR). Para ello se requiere el uso equipos ofimáticos con mayor capacidad de procesamiento que los equipos de uso personal o doméstico.

Se ejecuta en primer lugar el proceso *offline*, con el objetivo de evaluar los resultados y seleccionar los parámetros de entrada más adecuados para la construcción del modelo. A continuación, se procede a la aplicación de la metodología del proceso *online*. Se evaluarán los resultados obtenidos para la validación del proyecto y su potencial de aplicación en EDAR existentes.

En caso de obtenerse resultados incongruentes en el proceso *offline*, se realizarán los ajustes necesarios (Fase 4) para la corrección de errores de programación antes de continuar con el proceso *online*.

### ► **Fase 6. Elaboración de la memoria**

Para concluir, se elabora esta memoria que incluye toda la documentación y resultados relacionados con la metodología propuesta. Esto incluye la definición del alcance y los objetivos del proyecto, el análisis teórico de las técnicas empleadas, el desarrollo de la metodología, la experimentación con el caso de EDAR, el estudio de los resultados y la definición de líneas futuras del proyecto.

Asimismo, se incluye en esta fase del proyecto la corrección por parte del tutor del documento y la validación final de las conclusiones alcanzadas.

## 5.2 PLANIFICACIÓN

El desarrollo de este proyecto se planifica en 12 meses. El diagrama de Gantt de la Figura 25 recoge las distintas tareas en las que se ha ejecutado el proyecto global, definiendo la duración de cada una de ellas en semanas y considerando una dedicación parcial al proyecto (estimado en jornadas de 4 h/día).

Tabla 15. Enumeración tareas fases desarrollo del proyecto

Tarea	
<b>1. Análisis de la propuesta</b>	
1.1	Definición alcance y objetivos
1.2	Selección técnicas y herramientas a utilizar
1.3	Planificación de los trabajos
<b>2. Investigación y documentación</b>	
2.1	Revisión bibliográfica mRMR
2.2	Revisión bibliográfica control estadístico multivariado
2.3	Revisión bibliográfica <i>Deep Learning</i> /VAE
2.4	Revisión bibliográfica VAE en EDAR
<b>3. Diseño de la herramienta</b>	
3.1	Ampliación conocimiento en lenguaje de programación
3.2	Descripción a alto nivel estructura del programa
3.3	Definición pseudocódigo
3.4	Definición parámetros/hiperparámetros iteración VAEs
<b>4. Programación</b>	
4.1	Programación proceso <i>offline</i> centralizado
4.2	Programación proceso <i>offline</i> distribuido
4.3	Programación proceso <i>online</i> centralizado
4.4	Programación proceso <i>online</i> distribuido
<b>5. Implantación de la metodología</b>	
5.1	Experimentación proceso <i>offline</i> centralizado
5.2	Experimentación proceso <i>offline</i> distribuido
5.3	Análisis resultados proceso <i>offline</i>
5.4	Experimentación proceso <i>online</i> centralizado
5.5	Experimentación proceso <i>online</i> distribuido
5.6	Análisis resultados proceso <i>online</i>
<b>6. Elaboración de la memoria</b>	
6.1	Elaboración de la memoria
6.2	Revisión y validación resultados

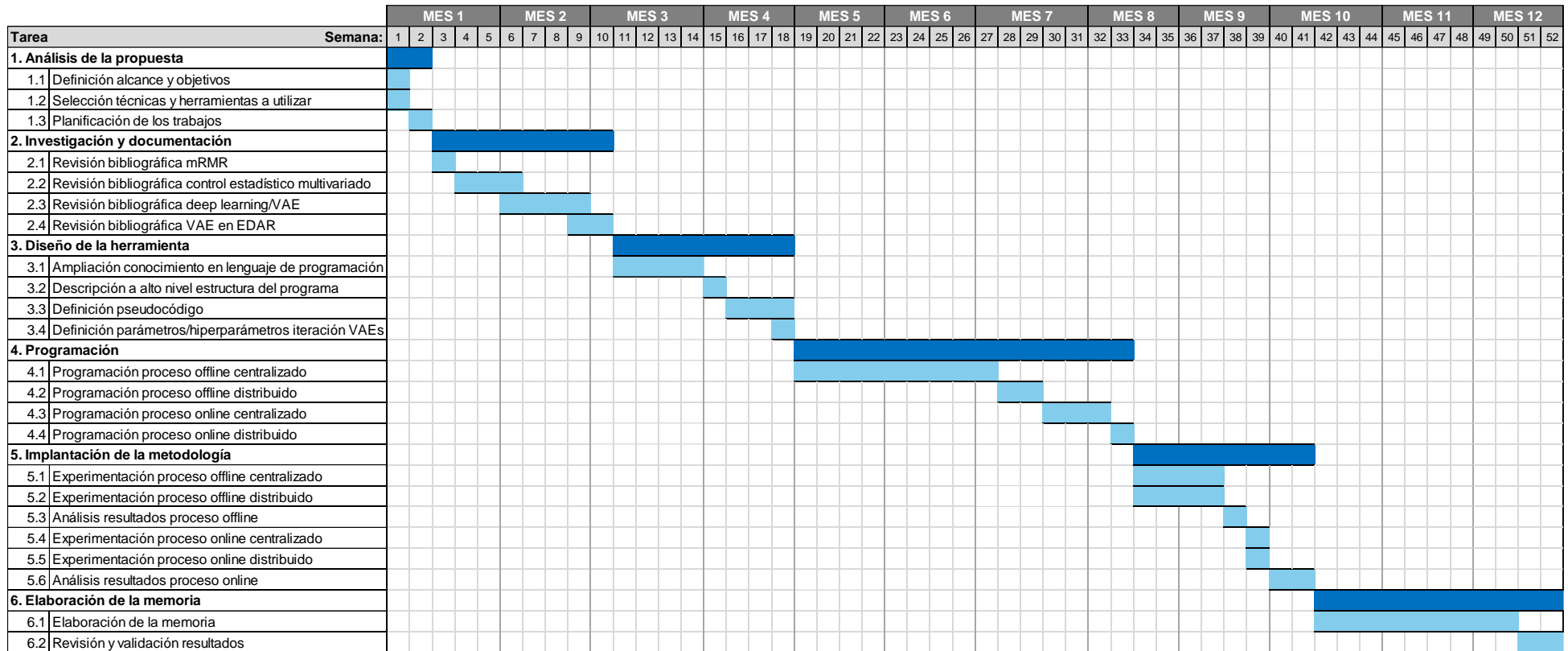


Figura 25. Diagrama de Gantt

## 5.3 ANÁLISIS ECONÓMICO

De cara a evaluar el coste total del proyecto, se distinguirán los costes directos (afectan directamente al producto o desarrollo del proyecto) y los costes indirectos (no directos pero asignables al coste final del proyecto).

Para la estimación de los costes se tendrán en cuenta la duración e implicación de los recursos, tanto humanos como materiales, en las distintas fases, así como los suministros necesarios para su desarrollo.

### 5.3.1. Costes directos

Para el cálculo de los costes directos de considerarán:

- Costes de mano de obra directa
- Amortización de equipos

#### 5.3.1.1 Mano de obra directa

En relación con los costes de mano de obra directa se estima una dedicación de jornada parcial al proyecto, es decir, 4 h diarias, contabilizando únicamente días laborales. De este modo, se tendrá un total de 20 h/semana.

Para las tareas 5.1 y 5.2 no se considerará coste de mano de obra directa (MOD) y para la tarea 6.2 únicamente se contabilizará una dedicación total de 16 h.

En cuanto el salario del trabajador que desarrolla el proyecto se tendrá en cuenta el salario bruto del mismo al ser este el coste que supone un empleado para una empresa, así como la seguridad social a cargo de la empresa (considerada en un 25%) y otros conceptos de recaudación conjunta (desempleo: 5,50%, fondo garantía salarial: 0,20%, formación profesional: 0,60%, cotización por Accidentes de Trabajo y Enfermedades Profesionales (AT y EP): 1,75%).

Tabla 16. Cálculo coste empleado para empresa (jornada completa)

Concepto	Base	Tipo	Aportación empresa
Salario bruto	35.000	100%	35.000,00 €
S.S. empresa		25%	8.750,00 €
Desempleo		5,50%	1.925,00 €
Fondo garantía salarial		0,20%	70,00 €
Formación profesional		0,60%	210,00 €
Cotización AT y EP		1,75%	612,50 €
<b>TOTAL EMPRESA/AÑO</b>			<b>46.567,50 €</b>

Dado que no se considera una dedicación total a este proyecto por parte del trabajador, se determinará el coste por día que supone para la empresa y posteriormente se extrapolará a los recursos requeridos por este proyecto:

Tabla 17. Cálculo jornadas efectivas al año de un trabajador

Jornadas laborales/año	260 días
Festivos anuales	-12 días
Vacaciones convenio	-23 días
Jornadas efectivas/año	225 días
<b>Coste por jornada efectiva</b>	<b>206,97 €/día</b>

Tabla 18. Cálculo coste mano de obra por fase de proyecto

Tarea	Duración (semanas)	MoD (días)	Dedicación	Coste MOD (€)
1. Análisis de la propuesta	2	10	50%	1.034,83
2. Investigación y documentación	8	40	50%	4.139,33
3. Diseño de la herramienta	8	40	50%	4.139,33
4. Programación	15	75	50%	7.761,25
5. Implantación metodología	8	20	50%	2.069,67
6. Elaboración de la memoria	11	49	50%	5.070,68
<b>TOTAL MOD PROYECTO</b>				<b>24.215,10</b>

### 5.3.1.1 Amortización de equipos

Para el cálculo de la amortización de los equipos ofimáticos utilizados para el desarrollo del proyecto se supondrá una amortización lineal de los mismos y una vida útil de 4 años, excepto en el caso de los periféricos que se aplicará una amortización lineal a 8 años.

Se considerarán únicamente los equipos personales utilizados y no se considerarán el sistema clúster de máquinas de la Universidad utilizado (considerándose como un alquiler a coste cero).

Tabla 19. Cálculo coste amortización equipos ofimáticos

Equipo	Ud.	Precio compra	Amortización	Coste amortiz./año	Coste amortiz./h
Ordenador portátil	1	800 €	4 años	200,00 €	0,0228 €/h
Pantalla auxiliar	1	130 €	4 años	32,50 €	0,0037 €/h
Periféricos	2	15 €	4 años	1,88 €	0,0002 €/h
<b>TOTAL</b>				<b>234,38 €</b>	<b>0,0268 €/h</b>

El uso de los equipos ofimáticos indicados superior al de recursos humanos en las tareas 4.1 y 4.2 ya que la simulación requiere de horas de ejecución sin necesidad de supervisión. En estas dos fases se determinará un uso de 10/h jornada de los equipos.

En base a lo anterior, el coste de amortización de equipos para cada una de las fases del proyecto es la siguiente:

Tabla 20. Cálculo coste amortización equipos por fase de proyecto

Tarea	Duración (semanas)	Uso equipos (horas)	Coste Amort. (€)
1. Análisis de la propuesta	2	40	1,07
2. Investigación y documentación	8	160	4,28
3. Diseño de la herramienta	8	160	4,28
4. Programación	15	630	16,86
5. Implantación metodología	8	80	2,14
6. Elaboración de la memoria	11	196	5,24
<b>TOTAL AMORTIZACIÓN EQUIPOS PROYECTO</b>			<b>33,87</b>

El total de los costes directos del proyecto asciende a 24.248,97 €.

### 5.3.2. Costes indirectos

Entre los costes indirectos solo se estimarán los derivados de gastos de suministros de electricidad, calefacción e internet, no considerando otros costes adicionales.

Como datos de partida se considerará:

- Precio electricidad promedio (coste fijo+variable): 0,15 €/kWh
- Demanda electricidad: 150 W
- Cuota mensual internet: 40 €/mes
- Precio gas natural promedio (coste fijo+variable): 0,05 €/kWh
- Consumo medio calefacción: 50 kWh/día, 6h/día

Tabla 21. Cálculo costes indirectos

Concepto	Cantidad	Coste unidad	Coste total
Electricidad	1.266 h x 150 W	0,15 €/kWh	28,49 €
Internet	936 h	0,056 €/h	52,00 €
Calefacción	320 h x 8,33 kWh/h	0,05 €/kWh	133,33 €
<b>TOTAL COSTES INDIRECTOS PROYECTO</b>			<b>213,82 €</b>

### 5.3.3. Coste total

El coste total que supone el proyecto resulta de la suma de los costes directos e indirectos anteriormente calculados:

Tabla 22. Cálculo coste total del proyecto

Concepto	Coste total
Costes directos	24.248,97 €
Costes indirectos	213,82 €
<b>Coste total proyecto</b>	<b>24.462,79 €</b>

### 5.3.4. Precio de venta

El coste total es la base de cálculo del precio de venta a cliente del proyecto.

Sobre el coste total se aplica un margen de beneficio. Cada empresa suele tener marcados unos márgenes de beneficio mínimos. En este caso, se aplicará un margen de beneficio del 21,6 % y el precio de venta con IVA incluido se refleja en la siguiente tabla:

Tabla 23. Precio de venta final

Coste total proyecto	24.462,79 €
Margen de beneficio (21,6%)	5.283,96 €
Precio venta sin IVA	29.746,75 €
IVA (21%)	6.246,82 €
<b>Precio de venta IVA incl.</b>	<b>35.993,57 €</b>



## 6. CONCLUSIONES Y LÍNEAS FUTURAS

---

### 6.1 CONCLUSIONES

La escasez del agua es uno de los desafíos más críticos que enfrenta la humanidad en la actualidad, agravado por el crecimiento poblacional y el cambio climático. Este problema subraya la imperativa necesidad de gestionar eficientemente el tratamiento de las aguas residuales en las EDAR.

Al mismo tiempo, el aumento en la complejidad de los procesos industriales y de la cantidad de datos disponibles en ellos han hecho que los sistemas de control tradicionales no sean viables.

En este contexto, surge la propuesta de este Trabajo Fin de Máster cuyo objetivo es desarrollar un sistema de detección temprana de fallos en una EDAR mediante el modelo de VAE mínimo, utilizando datos históricos del proceso no etiquetados, y comparar su eficacia y rendimiento según si se aplica en una aproximación centralizada o distribuida del control del proceso.

Utilizando como caso de estudio el modelo de EDAR de referencia BSM2 con datos del proceso cuartohorarios, se ha demostrado que el sistema propuesto es capaz de detectar fallos con desvío del 3% o superior, exceptuando fallos en la medida de O<sub>2</sub>, con un retardo de 1h o menos. En estos casos en los que se detecta el fallo, el control distribuido es, por lo general, más rápido en la detección que el centralizado.

Además, la eficacia para la detección de fallos de pequeña amplitud es también mayor en el caso distribuido que en el centralizado, como se ha comprobado en el estudio del fallo 1 (amplitud del 3%) donde sólo el primer método detecta el desvío.

No obstante, es importante señalar que fallos de pequeña amplitud o en la variable de O<sub>2</sub> disuelto no son detectados por el sistema en ninguna de las aproximaciones estudiadas. Esto podría solventarse aumentando la complejidad del VAE añadiendo capas intermedias adicionales e incrementando el tamaño de las mismas, permitiendo así un ajuste más fino del patrón aprendido por el VAE al perfil real del proceso.

Aumentar la sensibilidad también podría ocasionar una mayor tasa de falsas alarmas. La tasa de falsas alarmas obtenida es inferior al 10% en todos los casos – 1% si se desprecian las anomalías que detecta el grupo 2 del distribuido en los primeros registros. Por tanto, actualmente los resultados de falsas alarmas son muy positivos, tanto en la diagnosis centralizada, donde apenas se producen, como en la distribuida. Lo anterior indica que, pese a que al aumentar la complejidad del modelo pudiera incrementar esta tasa, cabe esperar que se mantenga dentro de un margen admisible.

Por otra parte, se destaca la relación entre la dimensionalidad de la entrada al proceso y los resultados del modelo de VAE mínimo estudiado. El tiempo de ejecución requerido para el entrenamiento de los modelos es en cómputo global inferior en el caso centralizado que en el distribuido. Sin embargo, al reducirse la dimensionalidad de cada grupo de variables en la aproximación distribuida, el entrenamiento individual de cada VAE requiere significativamente menos tiempo de ejecución. De este modo, la actualización y mejora del sistema de control es mucho más rápida y requiere menos recursos computacionales en el caso distribuido.

Finalmente es importante resaltar que el sistema de detección de fallos propuesto basado en un modelo VAE mínimo es capaz de aprender los patrones del proceso con solo tres días de datos históricos. A pesar de las restricciones inherentes a la simplicidad del modelo, éste ha mostrado una precisión suficiente para la detección efectiva de anomalías. Esta característica elimina la necesidad de grandes volúmenes de datos históricos para su entrenamiento, lo que mejora la eficiencia y la aplicabilidad del sistema en entornos con datos limitados y la actualización periódica del sistema.

En conclusión, el sistema propuesto ha demostrado su capacidad para detectar fallos de cierta magnitud con una velocidad de detección muy positiva en los casos de éxito en la detección, mostrando resultados más interesantes en el caso distribuido que en el centralizado. Además, el método de control distribuido ofrece mayores ventajas en la operación y mantenimiento del sistema, lo cual es de especial interés en EDAR al ser sistemas cuyo perfil de funcionamiento puede variar con la estacionalidad.

Estos resultados indican un potencial significativo para el desarrollo de aplicaciones futuras de control distribuido basado en VAEs. Estudiar modelos de VAE más complejos, con mayor número de capas ocultas, y ajustar los parámetros de entrenamiento del VAE son dos de las mejoras que permitirían mejorar el sistema y, particularmente, su precisión en el patrón de aprendizaje.

Con estos ajustes, el sistema podría ofrecer una herramienta robusta y eficiente para el control de fallos tanto en EDAR como otros entornos industriales, sin necesidad de disponer de grandes volúmenes de datos históricos del proceso y sin requerir un conocimiento experto del dominio para su definición, ventajas del DL frente a los métodos tradicionales de control de procesos.

## 6.2 LÍNEAS FUTURAS

Este trabajo me ha permitido profundizar en el ámbito de la IA y comprender en detalle el funcionamiento interno de las redes neuronales, específicamente los VAE. La aplicación de esta tecnología en una EDAR ha representado para mí una oportunidad única de aplicar conceptos actualmente en auge y de gran potencial como son las redes neuronales en un proceso característico de la ingeniería química, disciplina en la que inicialmente me formé.

Dado que el modelo VAE mínimo ha demostrado ser eficaz en muchos de los casos evaluados, sería interesante continuar su estudio en el control distribuido al ofrecer mayores ventajas para la operación y mantenimiento de la EDAR y desarrollar mejoras que permitan mejorar principalmente la sensibilidad de detección del sistema. Con ello se busca poder detectar cualquier tipo de fallo, incluidos aquellos de baja amplitud, manteniendo los tiempos de detección obtenidos en este trabajo y sin aumentar significativamente los recursos computacionales.

Para ello, se propone a continuación las líneas de trabajo futuro que se podrían seguir para tal fin u otras situaciones de estudio que podrían resultar también de interés:

### 1. Incrementar la complejidad de las redes neuronales del VAE

Una de las principales causas de la falta de eficacia en la detección de fallos se asocia a que el modelo utilizado de VAE es el más básico posible. Esto permite que el tiempo de ejecución y la potencia computacional requerida sea menor, reduciendo también el costo que supone para el negocio tanto por equipos como

por el almacenamiento de datos, pero también limita la complejidad del aprendizaje.

Así, se propone estudiar incrementar paulatinamente la complejidad del modelo hasta llegar a una situación que se considere adecuada para la detección de fallos de distinta tipología y distinta magnitud a un rendimiento y coste admisibles.

Hay que tener en cuenta que cada capa adicional supone reajustar la parametrización del modelo, lo cual hace necesario abordar el siguiente punto de propuestas de mejora.

## 2. Implementación nueva metodología de selección de parámetros e hiperparámetros del VAE

El sistema de detección se sustenta sobre el modelo VAE entrenado para el caso de estudio. Por ello, el primer paso para la mejora de este sería realizar un ajuste fino de los parámetros e hiperparámetros de entrenamiento del VAE para mejorar su precisión y capacidad de generalización. Esto podría implicar el uso de otras técnicas avanzadas no contempladas en este proyecto, como la búsqueda bayesiana o el uso de redes neuronales evolutivas.

## 3. Estudio del proceso en el régimen estacionario:

Una alternativa a estudiar antes de incrementar la complejidad de la red sería simular los distintos fallos en la EDAR del modelo BSM2 a partir del día 119, es decir, cuando entra en un régimen estacionario. De este modo, a priori podría no ser necesario aumentar la complejidad del VAE ya que se simplificaría el perfil de las distintas variables del proceso como se observa en los gráficos de los apartados 4.3.1.1. A y 4.3.2.1 A.

Estas líneas de acción están dirigidas a garantizar que el sistema propuesto no solo mantenga su capacidad actual de detección de fallos en tiempos aceptables, sino que también mejore su precisión y fiabilidad, permitiendo asegurar con alto grado de confianza la detección de fallos incluso en situaciones diferentes a las hasta ahora estudiadas.

## 7. BIBLIOGRAFÍA

---

- [1] UNESCO y ONU-Agua, «Informe Mundial de las Naciones Unidas sobre el Desarrollo de los Recursos Hídricos 2020: Agua y Cambio Climático», UNESCO, París, 2020. [En línea]. Disponible en: [www.unesco.org/open-access/](http://www.unesco.org/open-access/)
- [2] R. Martínez, N. Vela, A. el Aatik, E. Murray, P. Roche, y J. M. Navarro, «On the use of an IoT integrated system for water quality monitoring and management in wastewater treatment plants», *Water (Switzerland)*, vol. 12, n.º 4, abr. 2020, doi: 10.3390/W12041096.
- [3] MITERD, «Estación depuradora de aguas residuales (EDAR)». Accedido: 14 de mayo de 2023. [En línea]. Disponible en: <https://www.miteco.gob.es/es/agua/temas/saneamiento-depuracion/sistemas/edar/default.aspx>
- [4] *Real Decreto 509/1996, de 15 de marzo, de desarrollo del Real Decreto-ley 11/1995, de 28 de diciembre, por el que se establecen las normas aplicables al tratamiento de las aguas residuales urbanas*. Ministerio de Obras Públicas, Transportes y Medio Ambiente, 1996, pp. 12038-12041. Accedido: 14 de mayo de 2023. [En línea]. Disponible en: <https://www.boe.es/buscar/doc.php?id=BOE-A-1996-7159>
- [5] «Europe's €1.8 billion water reuse investment soars amid scarcity, regulation, and recovery», *Smart Water Magazine*, ago. 2023, Accedido: 25 de marzo de 2024. [En línea]. Disponible en: <https://smartwatermagazine.com/news/bluefield-research/europes-eu18-billion-water-reuse-investment-soars-amid-scarcity-regulation>
- [6] V. Sánchez, «Estación depuradora de agua residual: ¿Qué es? - Fundación Aquae». Accedido: 14 de mayo de 2023. [En línea]. Disponible en: <https://www.fundacionaquae.org/estacion-agua-depuradora>
- [7] I. Perez Colo, C. Saavedra Sueldo, M. De Paula, y G. G. Acosta, «Intelligent approach for the industrialization of deep learning solutions applied to fault detection», *Expert Syst Appl*, vol. 233, p. 120959, dic. 2023, doi: 10.1016/J.ESWA.2023.120959.
- [8] J. Qian, Z. Song, Y. Yao, Z. Zhu, y X. Zhang, «A review on autoencoder based representation learning for fault detection and diagnosis in industrial processes», *Chemometrics and Intelligent Laboratory Systems*, vol. 231, p. 104711, dic. 2022, doi: 10.1016/J.CHEMOLAB.2022.104711.
- [9] M. Alvi *et al.*, «Deep learning in wastewater treatment: a critical review», *Water Res*, vol. 245, p. 120518, oct. 2023, doi: 10.1016/J.WATRES.2023.120518.
- [10] J. P. Usuga Cadavid, S. Lamouri, B. Grabot, R. Pellerin, y A. Fortin, «Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0», *J Intell Manuf*, vol. 31, n.º 6, pp. 1531-1558, ago. 2020, doi: 10.1007/S10845-019-01531-7.
- [11] M. Khan, W. Chuenchart, K. C. Surendra, y S. Kumar Khanal, «Applications of artificial intelligence in anaerobic co-digestion: Recent advances and

- prospects», *Bioresour Technol*, vol. 370, p. 128501, feb. 2023, doi: 10.1016/J.BIORTECH.2022.128501.
- [12] S. Qiu *et al.*, «Deep Learning Techniques in Intelligent Fault Diagnosis and Prognosis for Industrial Systems: A Review», *Sensors 2023, Vol. 23, Page 1305*, vol. 23, n.º 3, p. 1305, ene. 2023, doi: 10.3390/S23031305.
- [13] G. Fu, Y. Jin, S. Sun, Z. Yuan, y D. Butler, «The role of deep learning in urban water management: A critical review», *Water Res*, vol. 223, p. 118973, sep. 2022, doi: 10.1016/J.WATRES.2022.118973.
- [14] MICIU, *Extracto de la Resolución de 6 de septiembre de 2019, de la Presidencia de la Agencia Estatal de Investigación, por la que se aprueba la convocatoria de tramitación anticipada para el año 2019 del procedimiento de concesión de ayudas a «Proyectos de I+D+i», en el marco del Programa Estatal de Generación de Conocimiento y Fortalecimiento Científico y Tecnológico del Sistema de I+D+i y del Programa Estatal de I+D+i Orientada a los Retos de la Sociedad, del Plan Estatal de I+D+i 2017-2020*. 2019, pp. 48757-48759.
- [15] MICIU, *Orden CNU/320/2019, de 13 de marzo, por la que se aprueban las bases reguladoras para la concesión de ayudas públicas en el marco del Programa Estatal de Generación de Conocimiento y Fortalecimiento Científico y Tecnológico del Sistema de I+D+i y en el marco del Programa Estatal de I+D+i Orientada a los Retos de la Sociedad, del Plan Estatal de Investigación Científica y Técnica y de Innovación 2017-2020, destinadas a organismos de investigación y de difusión de conocimientos*. 2019, pp. 29527-29570. [En línea]. Disponible en: <https://www.boe.es/eli/es/o/2019/03/13/cnu320>
- [16] Q. Jiang y X. Yan, «Dynamic CCA-Based Distributed Monitoring for Multiunit Non-Gaussian Processes», *IFAC-PapersOnLine*, vol. 51, n.º 21, pp. 347-352, ene. 2018, doi: 10.1016/J.IFACOL.2018.09.444.
- [17] J. Yang y L. Wang, «Nonlocal, local and global preserving stacked autoencoder based fault detection method for nonlinear process monitoring», *Chemometrics and Intelligent Laboratory Systems*, vol. 235, p. 104758, abr. 2023, doi: 10.1016/J.CHEMOLAB.2023.104758.
- [18] D. Slišković, R. Grbić, y Ž. Hocenski, «Multivariate statistical process monitoring», *Tehnički vjesnik*, vol. 19, n.º 1, pp. 33-41, mar. 2012.
- [19] W. Sun, Z. Zhou, F. Ma, J. Wang, y C. Ji, «Industrial Application of Data-Driven Process Monitoring with an Automatic Selection Strategy for Modeling Data», *Processes*, vol. 11, n.º 2, feb. 2023, doi: 10.3390/pr11020402.
- [20] A. Mouzakis, «Classification of fault diagnosis methods for control systems», *Measurement and Control (United Kingdom)*, vol. 46, n.º 10, pp. 303-308, dic. 2013, doi: 10.1177/0020294013510471/ASSET/IMAGES/LARGE/10.1177\_0020294013510471-FIG3.JPEG.
- [21] Z. Morrison, B. P. Russo, Y. Lian, y R. Kamalapurkar, «Fault Detection via Occupation Kernel Principal Component Analysis», *IEEE Control Syst Lett*, vol. 7, pp. 2695-2700, mar. 2023, doi: 10.1109/LCSYS.2023.3287568.
- [22] V. Venkatasubramanian, R. Rengaswamy, K. Yin, y S. N. Kavuri, «A review of process fault detection and diagnosis: Part I: Quantitative model-based

- methods», *Comput Chem Eng*, vol. 27, n.º 3, pp. 293-311, mar. 2003, doi: 10.1016/S0098-1354(02)00160-6.
- [23] Y. Pan, C. Yang, Y. Sun, R. An, y L. Wang, «Fault detection with principal component pursuit method», *J Phys Conf Ser*, vol. 659, p. 012035, nov. 2015, doi: 10.1088/1742-6596/659/1/012035.
- [24] V. Venkatasubramanian, R. Rengaswamy, y S. N. Kavuri, «A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies», *Comput Chem Eng*, vol. 27, n.º 3, pp. 313-326, mar. 2003, doi: 10.1016/S0098-1354(02)00161-8.
- [25] D.-H. Ham, «Modelling work domain knowledge with the combined use of abstraction hierarchy and living systems theory», doi: 10.1007/s10111-015-0338-y.
- [26] O. P. Gandhi y V. P. Agrawal, «FMEA—A diagraph and matrix approach», *Reliab Eng Syst Saf*, vol. 35, n.º 2, pp. 147-158, ene. 1992, doi: 10.1016/0951-8320(92)90034-I.
- [27] H. Bai, «Research on Network Equipment Fault Detection Based on Fault Tree Analysis», *Procedia Comput Sci*, vol. 228, pp. 271-280, ene. 2023, doi: 10.1016/J.PROCS.2023.11.031.
- [28] C. Liu, J. Bai, y F. Wu, «Fault Diagnosis Using Dynamic Principal Component Analysis and GA Feature Selection Modeling for Industrial Processes», *Processes 2022, Vol. 10, Page 2570*, vol. 10, n.º 12, p. 2570, dic. 2022, doi: 10.3390/PR10122570.
- [29] S. Yin, S. X. Ding, A. Haghani, H. Hao, y P. Zhang, «A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process», *J Process Control*, vol. 22, n.º 9, pp. 1567-1581, oct. 2012, doi: 10.1016/J.JPROCONT.2012.06.009.
- [30] S. Lee, M. Kwak, K. L. Tsui, y S. B. Kim, «Process monitoring using variational autoencoder for high-dimensional nonlinear processes», *Eng Appl Artif Intell*, vol. 83, pp. 13-27, ago. 2019, doi: 10.1016/J.ENGAPPAI.2019.04.013.
- [31] N. Migenda, R. Möller, y W. Schenck, «Adaptive dimensionality reduction for neural network-based online principal component analysis», *PLoS One*, vol. 16, p. e0248896, mar. 2021, doi: 10.1371/journal.pone.0248896.
- [32] F. Harrou, M. N. Nounou, H. N. Nounou, y M. Madakyaru, «PLS-based EWMA fault detection strategy for process monitoring», *J Loss Prev Process Ind*, vol. 36, pp. 108-119, jul. 2015, doi: 10.1016/J.JLP.2015.05.017.
- [33] Z. Zhao *et al.*, «Instance transfer partial least squares for semi-supervised adaptive soft sensor», *Chemometrics and Intelligent Laboratory Systems*, vol. 245, p. 105062, feb. 2024, doi: 10.1016/J.CHEMOLAB.2024.105062.
- [34] L. Shu, F. Lu, y Y. Chen, «Robust forecasting with scaled independent component analysis», *Financ Res Lett*, vol. 51, p. 103399, ene. 2023, doi: 10.1016/J.FRL.2022.103399.
- [35] S. Choi, «Independent Component Analysis», en *Encyclopedia of Biometrics*, S. Z. Li y A. Jain, Eds., Boston, MA: Springer US, 2009, pp. 735-741. doi: 10.1007/978-0-387-73003-5\_305.

- [36] Q. Jiang y B. Huang, «Distributed monitoring for large-scale processes based on multivariate statistical analysis and Bayesian method», *J Process Control*, vol. 46, pp. 75-83, oct. 2016, doi: 10.1016/J.JPROCONT.2016.08.006.
- [37] M. J. Fuente, M. Galende-Hernández, y G. I. Sainz-Palmero, «Data-based decomposition plant for decentralized monitoring schemes: A comparative study», *J Process Control*, vol. 135, p. 103178, mar. 2024, doi: 10.1016/J.JPROCONT.2024.103178.
- [38] A. Sánchez-Fernández, F. J. Baldán, G. I. Sainz-Palmero, J. M. Benítez, y M. J. Fuente, «Fault detection based on time series modeling and multivariate statistical process control», *Chemometrics and Intelligent Laboratory Systems*, vol. 182, pp. 57-69, nov. 2018, doi: 10.1016/J.CHEMOLAB.2018.08.003.
- [39] S. Heo y J. H. Lee, «Statistical Process Monitoring of the Tennessee Eastman Process Using Parallel Autoassociative Neural Networks and a Large Dataset», 2019, doi: 10.3390/pr7070411.
- [40] R. H. Martello, L. Ranzan, M. Farenzena, y J. O. Trierweiler, «Improving Autoencoder Training with novel Goal Functions based on Multivariable Control Concepts», *IFAC-PapersOnLine*, vol. 54, n.º 3, pp. 73-78, ene. 2021, doi: 10.1016/J.IFACOL.2021.08.221.
- [41] F. S. Zaidi, H. L. Dai, M. Imran, y K. P. Tran, «Analyzing abnormal pattern of hotelling T2 control chart for compositional data using artificial neural networks», *Comput Ind Eng*, vol. 180, p. 109254, jun. 2023, doi: 10.1016/J.CIE.2023.109254.
- [42] Z. Al-Ghazawi y R. Alawneh, «Use of artificial neural network for predicting effluent quality parameters and enabling wastewater reuse for climate change resilience – A case from Jordan», *Journal of Water Process Engineering*, vol. 44, p. 102423, dic. 2021, doi: 10.1016/J.JWPE.2021.102423.
- [43] Z. Wang *et al.*, «Effluent quality prediction of the sewage treatment based on a hybrid neural network model: Comparison and application», *J Environ Manage*, vol. 351, p. 119900, feb. 2024, doi: 10.1016/J.JENVMAN.2023.119900.
- [44] D. Garcia-Alvarez, M. J. Fuente, P. Vega, y G. Sainz, «Fault Detection and Diagnosis using Multivariate Statistical Techniques in a Wastewater Treatment Plant.», *IFAC Proceedings Volumes*, vol. 42, n.º 11, pp. 952-957, ene. 2009, doi: 10.3182/20090712-4-TR-2008.00156.
- [45] P. Chiñas-Sánchez, I. López-Juárez, y J. A. Vázquez-López, «Reconocimiento de variables multivariantes empleando el estadístico T2 Hotelling y MEWMA mediante las RNA's», *Ingeniería, investigación y tecnología*, vol. 15, n.º 1, pp. 125-138, 2014, [En línea]. Disponible en: [http://www.scielo.org.mx/scielo.php?script=sci\\_arttext&pid=S1405-77432014000100011&lng=es&nrm=iso](http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-77432014000100011&lng=es&nrm=iso). ISSN 2594-0732.
- [46] H. Albazzaz, X. Z. Wang, y F. Marhoon, «Multidimensional visualisation for process historical data analysis: a comparative study with multivariate statistical process control», *J Process Control*, vol. 15, n.º 3, pp. 285-294, abr. 2005, doi: 10.1016/J.JPROCONT.2004.06.007.
- [47] T. Biegel, N. Jourdan, C. Hernandez, A. Cviko, y J. Metternich, «Deep learning for multivariate statistical in-process control in discrete manufacturing: A case

- study in a sheet metal forming process», *Procedia CIRP*, vol. 107, pp. 422-427, ene. 2022, doi: 10.1016/J.PROCIR.2022.05.002.
- [48] M. Imran, H. L. Dai, F. S. Zaidi, X. Hu, K. P. Tran, y J. Sun, «Analyzing out-of-control signals of T2 control chart for compositional data using artificial neural networks», *Expert Syst Appl*, vol. 238, p. 122165, mar. 2024, doi: 10.1016/J.ESWA.2023.122165.
- [49] S. Ramírez-Gallego *et al.*, «Fast-mRMR: Fast Minimum Redundancy Maximum Relevance Algorithm for High-Dimensional Big Data», *International Journal of Intelligent Systems*, vol. 32, n.º 2, pp. 134-152, feb. 2017, doi: 10.1002/INT.21833.
- [50] M. Mandal y A. Mukhopadhyay, «An Improved Minimum Redundancy Maximum Relevance Approach for Feature Selection in Gene Expression Data», *Procedia Technology*, vol. 10, pp. 20-27, ene. 2013, doi: 10.1016/J.PROTCY.2013.12.332.
- [51] L. Yu y H. Liu, «Efficient Feature Selection via Analysis of Relevance and Redundancy», *Journal of Machine Learning Research*, vol. 5, pp. 1205-1224, dic. 2004.
- [52] C. Ding y H. Peng, «Minimum redundancy feature selection from microarray gene expression data», *J Bioinform Comput Biol*, vol. 03, n.º 02, pp. 185-205, abr. 2005, doi: 10.1142/S0219720005001004.
- [53] M. Radovic, M. Ghalwash, N. Filipovic, y Z. Obradovic, «Minimum redundancy maximum relevance feature selection approach for temporal gene expression data», *BMC Bioinformatics*, vol. 18, n.º 1, pp. 1-14, ene. 2017, doi: 10.1186/S12859-016-1423-9/FIGURES/6.
- [54] H. Peng, F. Long, y C. Ding, «Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy», *IEEE Trans Pattern Anal Mach Intell*, vol. 27, n.º 8, pp. 1226-1238, 2005, doi: 10.1109/TPAMI.2005.159.
- [55] S. Xie, Y. Zhang, D. Lv, X. Chen, J. Lu, y J. Liu, «A new improved maximal relevance and minimal redundancy method based on feature subset», *J Supercomput*, vol. 79, n.º 3, pp. 3157-3180, 2023, doi: 10.1007/s11227-022-04763-2.
- [56] Q. Song y P. Jiang, «A multi-scale convolutional neural network based fault diagnosis model for complex chemical processes», *Process Safety and Environmental Protection*, vol. 159, pp. 575-584, mar. 2022, doi: 10.1016/J.PSEP.2021.11.020.
- [57] K. Demertzis, S. Demertzis, y L. Iliadis, *A Selective Survey Review of Computational Intelligence Applications in the Primary Subdomains of Civil Engineering Specializations*. 2023. doi: 10.20944/preprints202301.0270.v1.
- [58] X. Zhao, L. Zhang, Y. Cao, K. Jin, y Y. Hou, «Anomaly Detection Approach in Industrial Control Systems Based on Measurement Data», *Information*, vol. 13, n.º Advances in Computing, Communication & Security, p. 450, sep. 2022, doi: 10.3390/info13100450.



- [59] L. Alzubaidi *et al.*, «Review of deep learning: concepts, CNN architectures, challenges, applications, future directions», *J Big Data*, vol. 8, n.º 1, p. 53, 2021, doi: 10.1186/s40537-021-00444-8.
- [60] M. M. Hamed, M. G. Khalafallah, y E. A. Hassanien, «Prediction of wastewater treatment plant performance using artificial neural networks», *Environmental Modelling & Software*, vol. 19, n.º 10, pp. 919-928, oct. 2004, doi: 10.1016/J.ENVSOFT.2003.10.005.
- [61] S. Plakias y Y. S. Boutalis, «A novel information processing method based on an ensemble of Auto-Encoders for unsupervised fault detection», *Comput Ind*, vol. 142, p. 103743, nov. 2022, doi: 10.1016/J.COMPIND.2022.103743.
- [62] A. H. Ba-Alawi, J. Loy-Benitez, S. Y. Kim, y C. K. Yoo, «Missing data imputation and sensor self-validation towards a sustainable operation of wastewater treatment plants via deep variational residual autoencoders», *Chemosphere*, vol. 288, p. 132647, feb. 2022, doi: 10.1016/J.CHEMOSPHERE.2021.132647.
- [63] C. Ji y W. Sun, «A Review on Data-Driven Process Monitoring Methods: Characterization and Mining of Industrial Data», *Processes 2022, Vol. 10, Page 335*, vol. 10, n.º 2, p. 335, feb. 2022, doi: 10.3390/PR10020335.
- [64] R. A. Khalil, N. Saeed, M. Masood, Y. M. Fard, M.-S. Alouini, y T. Y. Al-Naffouri, «Deep Learning in the Industrial Internet of Things: Potentials, Challenges, and Emerging Applications», *IEEE Internet Things J*, vol. 8, n.º 14, pp. 11016-11040, 2021, doi: 10.1109/JIOT.2021.3051414.
- [65] T. A. Pham, J. H. Lee, y C. S. Park, «MST-VAE: Multi-Scale Temporal Variational Autoencoder for Anomaly Detection in Multivariate Time Series», *Applied Sciences 2022, Vol. 12, Page 10078*, vol. 12, n.º 19, p. 10078, oct. 2022, doi: 10.3390/APP121910078.
- [66] D. D. ' Avino, D. Cozzolino, G. Poggi, y L. Verdoliva, «Auto-Encoders in Deep Learning—A Review with New Perspectives», *Mathematics 2023, Vol. 11, Page 1777*, vol. 11, n.º 8, p. 1777, abr. 2023, doi: 10.3390/MATH11081777.
- [67] D. P. Kingma y M. Welling, «Auto-Encoding Variational Bayes», dic. 2013, doi: 10.61603/ceas.v2i1.33.
- [68] J. Bobadilla, F. Ortega, A. Gutierrez, y Á. González-Prieto, «Deep variational models for collaborative filtering-based recommender systems», *Neural Comput Appl*, vol. 35, dic. 2022, doi: 10.1007/s00521-022-08088-2.
- [69] C. Huang, Y. Chai, Z. Zhu, B. Liu, y Q. Tang, «A Novel Distributed Fault Detection Approach Based on the Variational Autoencoder Model», *ACS Omega*, vol. 7, n.º 3, pp. 2996-3006, ene. 2022, doi: 10.1021/acsomega.1c06033.
- [70] F. Yu, J. Liu, y D. Liu, «Multimode Process Monitoring Based on Modified Density Peak Clustering and Parallel Variational Autoencoder», *Mathematics 2022, Vol. 10, Page 2526*, vol. 10, n.º 14, p. 2526, jul. 2022, doi: 10.3390/MATH10142526.
- [71] C. Peng, W. Kai, Z. Kun, y M. Fanchao, «Monitoring of wastewater treatment process based on multi-stage variational autoencoder», *Expert Syst Appl*, vol. 207, p. 117919, nov. 2022, doi: 10.1016/J.ESWA.2022.117919.

- [72] D. Vrecko, K. V Gernaey, C. Rosen, y U. Jeppsson, «Benchmark Simulation Model No 2 in Matlab-Simulink: towards plant-wide WWTP control strategy evaluation», *Water Science and Technology*, vol. 54, n.º 8, pp. 65-72, oct. 2006, doi: 10.2166/wst.2006.773.
- [73] A. Beneyto-Rodríguez, G. Sainz-Palmero, M. Galende-Hernández, y M. Fuente, *Sliding Time Window And Linguistic Ordered Weighted Averaging Based Deep Learning Pooling for Fault Diagnosis in a Wastewater Treatment Plant*. 2024. doi: 10.2139/ssrn.4804526.
- [74] N. Wang, H. Li, F. Wu, R. Zhang, y F. Gao, «Fault Diagnosis of Complex Chemical Processes Using Feature Fusion of a Convolutional Network», *Ind Eng Chem Res*, vol. 60, n.º 5, pp. 2232-2248, feb. 2021, doi: 10.1021/ACS.IECR.0C05739.
- [75] N. Amini y Q. Zhu, «Fault detection and diagnosis with a novel source-aware autoencoder and deep residual neural network», *Neurocomputing*, vol. 488, pp. 618-633, jun. 2022, doi: 10.1016/J.NEUCOM.2021.11.067.
- [76] I. Lomov, M. Lyubimov, I. Makarov, y L. E. Zhukov, «Fault detection in Tennessee Eastman process with temporal deep learning models», *J Ind Inf Integr*, vol. 23, p. 100216, sep. 2021, doi: 10.1016/J.JII.2021.100216.
- [77] H. Wu y J. Zhao, «Deep convolutional neural network model based chemical process fault diagnosis», *Comput Chem Eng*, vol. 115, pp. 185-197, jul. 2018, doi: 10.1016/J.COMPCHEMENG.2018.04.009.