



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería Mecánica

**Detección y diagnóstico de fallos mediante
técnicas basadas en datos aplicadas a un
colector de aguas**

Autor:

Elena Gómez, Alberto

Tutor:

**De la Fuente Aparicio,
María Jesús**

**Departamento de
Ingeniería de Sistemas y
Automática**

Valladolid, Julio 2022.



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES



ÍNDICE DE CONTENIDOS

RESUMEN	11
ABSTRACT.....	12
CAPÍTULO I: INTRODUCCIÓN Y OBJETIVOS	13
1.1 INTRODUCCIÓN.....	15
1.2. OBJETIVOS.....	16
1.3. ORGANIZACIÓN.....	16
CAPÍTULO II: MARCO TEÓRICO.....	19
2.1. CONTROL DE CALIDAD	21
2.2. CONTROL ESTADÍSTICO DE PROCESOS	23
2.2.1 VARIABILIDAD EN EL PROCESO DE PRODUCCION.....	23
2.2.2. CAUSAS DE LA VARIABILIDAD	25
2.2.3. GRAFICOS DE CONTROL	27
2.3. Análisis de Componentes Principales (PCA).....	29
2.3.1. Introducción.....	29
2.3.2. Objetivos del PCA.....	29
2.3.3. Desarrollo matemático del método PCA	30
2.3.4. Estadísticas para la monitorización.....	32
2.4. TÉCNICAS DE APRENDIZAJE AUTOMÁTICO.....	34
2.4.1. REDES NEURONALES ARTIFICIALES	35
2.4.2. APRENDIZAJE AUTOMÁTICO.....	37
2.4.3. AUTOENCODERS	38
2.4.4. ÁRBOLES DE DECISIÓN (RANDOM FOREST).....	41
2.4.5. MATRIZ DE CONFUSIÓN.....	44
CAPÍTULO III: COLECTOR DE AGUAS.....	47
3.1 CASO DE ESTUDIO	49
3.2 DESCRIPCIÓN DE LA PLANTA.....	49
3.3 VARIABLES Y DATOS DE LA PLANTA.....	51
CAPÍTULO IV: APLICACIÓN DESARROLLADA	55
4.1. PROCEDIMIENTO. DATOS DE FALLO.....	57



4.2. MODELOS PARA LA DETECCIÓN DE FALLOS	58
4.2.1. ANÁLISIS DE COMPONENTES PRINCIPALES (PCA).....	58
4.2.2. PCA NO LINEAL.....	67
4.2.3. PCA DEL RESIDUO.....	78
4.3. MODELOS PARA LA CLASIFICACIÓN DE FALLOS.....	86
4.3.1. MODELO DE RED NEURONAL MONOCAPA.....	86
4.3.2. MODELO DE RED NEURONAL MULTICAPA	94
4.3.3. MODELO DE ARBOLES DE DECISIÓN	101
CAPÍTULO V: CONCLUSIONES Y TRABAJO FUTURO	107
5.1. CONCLUSIONES	109
5.2. TRABAJO FUTURO	110
REFERENCIAS.....	113



ÍNDICE DE FIGURAS

Figura 1. Variaciones naturales y asignables.....	24
Figura 2. Tipos de salida del proceso.	25
Figura 3. Distribución de proceso simétrica.	25
Figura 4. Ejemplo de los efectos de las causas asignables sobre la distribución de un proceso.....	26
Figura 5. Ejemplo del efecto de las causas asignables sobre la distribución de un proceso.....	26
Figura 6. Ejemplo de diagrama de control para detección de anomalías.	27
Figura 7. Factores para calcular acotamientos 3σ para la gráfica R [6].	28
Figura 8. Representación de las m dimensiones del conjunto de datos.....	30
Figura 9. Relación entre la Inteligencia Artificial, Machine Learning y Deep Learning [11].	35
Figura 10. Ejemplo de neurona artificial con 5 entradas [12].....	36
Figura 11. Arquitectura de una Red Neuronal Feedforward con dos capas ocultas [14].	37
Figura 12. Arquitectura de un Autoencoder [16].	39
Figura 13. Sobreajuste de una red [17].	39
Figura 14. Combinación de 2 Autoencoders [18].	40
Figura 15. Esquema de la clasificación de las técnicas Data Mining [19].	41
Figura 16. Esquema del árbol de decisión [20].	43



Figura 17. Ejemplo de matriz de confusión con tasa de clasificación=98.4 %.[4]
.....45

Figura 18. Esquema de la planta de alcantarillado.....50

Figura 19 Diagrama de bloques simplificado de la planta de alcantarillado..50

Figura 20. Variables del proceso.52

Figura 21. Código del archivo para provocar fallos.53

Figura 22. Tabla de fallos provocados en la planta.57

Figura 23. **(a) Arriba.** Gráfico de control T^2 para el fallo en n_5 (offset=2). **(b) Abajo.** Gráfico de control Q para el fallo en n_5 (offset=2).60

Figura 24. Resultado paramétrico del fallo en n_5 (offset=2).61

Figura 25. **(a) Arriba.** Gráfico de control T^2 para el fallo en n_4 (offset=2,4). **(b) Abajo.** Gráfico de control Q para el fallo en n_4 (offset=2,4).62

Figura 26. Resultado paramétrico del fallo en n_4 (offset=2).62

Figura 27. Resumen de los 10 fallos provocados en la planta con umbral constante.....63

Figura 28. **(a) Arriba.** Gráfico de control T^2 para comportamiento normal. **(b) Abajo.** Gráfico de control Q para comportamiento normal.64

Figura 29. Esquema de funcionamiento de la ventana deslizante aplicada a un vector.65

Figura 30. **(a) Arriba.** Gráfico de control T^2 para el fallo en n_5 (offset=2,4). **(b) Abajo.** Gráfico de control Q para el fallo en n_5 (offset=2,4).66

Figura 31. Resultados de PCA adaptativo con fallo n_5 (offset=2,4).66

Figura 32. Resumen de los 10 fallos provocados en la planta con umbral adaptativo.....67



Figura 33. Arquitectura de la red final obtenida.....68

Figura 34. a) Arquitectura Autoencoder 1. b) Arquitectura Autoencoder 2. c) Arquitectura Autoencoder 3. d)Arquitectura Red..... 70

Figura 35. Gráficas de entrenamiento de Autoencoders y red..... 71

Figura 36. Tabla de resultados de PCA no lineal con umbral constante. 73

Figura 37. **(a) Arriba.** Gráfico de control T^2 para el fallo en q_7 (offset=2000). **(b) Abajo.** Gráfico de control Q para el fallo en q_7 (offset=2000). 74

Figura 38. **(a) Arriba.** Gráfico de control T^2 para el fallo en n_4 (offset=2,4). **(b) Abajo.** Gráfico de control Q para el fallo en n_4 (offset=2,4). 75

Figura 39. Tabla de resultados de PCA no lineal con umbral adaptativo. 76

Figura 40. **(a) Arriba.** Gráfico de control T^2 para el fallo en q_8 (offset=8500). **(b) Abajo.** Gráfico de control Q para el fallo en q_8 (offset=8500). 77

Figura 41. **(a) Arriba.** Gráfico de control T^2 para el fallo en n_4 (offset=2). **(b) Abajo.** Gráfico de control Q para el fallo en n_4 (offset=2). 78

Figura 42. Representación del entrenamiento de la red. 79

Figura 43. **(a) Arriba.** Gráfico de control T^2 para el funcionamiento normal con umbral constante. **(b) Abajo.** Gráfico de control Q funcionamiento normal con umbral constante. 80

Figura 44. **(a) Arriba.** Gráfico de control T^2 para funcionamiento normal con umbral adaptativo. **(b) Abajo.** Gráfico de control Q para funcionamiento normal con umbral adaptativo. 81

Figura 45. Resultados PCA con residuo y umbral constante. 82

Figura 46. **(a) Arriba.** Gráfico de control T^2 para el fallo n_4 (offset=2,4) con umbral constante. **(b) Abajo.** Gráfico de control Q para el fallo n_4 (offset=2,4) con umbral constante. 83



Figura 47. Resultados de PCA con residuo y umbral adaptativo84

Figura 48. **(a) Arriba.** Gráfico de control T^2 para el fallo n5 (offset=2) con umbral adaptativo. **(b) Abajo.** Gráfico de control Q para el fallo n5 (offset=2) con umbral adaptativo.....85

Figura 49. Matriz de Confusión del entrenamiento con 18 neuronas.....87

Figura 50. Matriz de confusión del entrenamiento con 4 neuronas.....88

Figura 51. Matriz de confusión del entrenamiento con 10 neuronas.88

Figura 52. Entrenamiento de la red.....89

Figura 53. Validación del fallo en q8.....90

Figura 54. Resumen de resultados para la clasificación de 1 fallo por parte del modelo monocapa.90

Figura 55. Validación del fallo en la pareja de fallos q8 y n5.....91

Figura 56. Validación del fallo en la pareja de fallos q8 y q7 con el modelo monocapa.....91

Figura 57. Validación del fallo en la pareja de fallos n4 y n5 con el modelo monocapa.....92

Figura 58. Resumen de resultados para parejas de datos con modelo monocapa.....93

Figura 59. Matriz de confusión para la clasificación de 5 fallos simultáneamente con el modelo monocapa.....94

Figura 60. Resultado del entrenamiento de los Autoencoders por separado.96

Figura 61. Matriz de confusión de la efectividad del entrenamiento de la red completa.....97

Figura 62. Validación del fallo en q7.....98



Figura 63. Resumen de resultados para la clasificación de 1 fallo por parte del modelo multicapa.	98
Figura 64. Validación del fallo en la pareja de fallos q8 y qwwtp con el modelo multicapa.	99
Figura 65. Validación del fallo en la pareja de fallos n5 y q7 con el modelo multicapa.	99
Figura 66. Resumen de resultados para parejas de datos con modelo multicapa.	100
Figura 67. Matriz de confusión para la clasificación de 5 fallos simultáneos con el modelo multicapa.	101
Figura 68. Estructura de un árbol ensamblado en el bosque.	102
Figura 69. Matriz de confusión del entrenamiento del bosque con 200 árboles.	103
Figura 70. Matriz de confusión para bosque de 200 árboles.	104
Figura 71. Matriz de confusión para bosque de 600 árboles.	104
Figura 72. Matriz de confusión para bosque de 1200 árboles.	105



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES



RESUMEN

El trabajo que se ha realizado pretende conseguir una mejora en la monitorización y la calidad de un proceso industrial. Para ello, se exponen diferentes técnicas de detección y diagnóstico de fallos basadas en datos. Esto es posible gracias a la tecnología del BigData que, a través de la implementación masiva de sensores y el control distribuido en planta, ha permitido a las empresas obtener una gran cantidad de datos que se pueden estudiar y emplear de numerosas maneras distintas. Las técnicas de control estadístico de procesos que se emplean en este trabajo son, en primer lugar, el Análisis de Componentes Principales (PCA), que permite reducir el número de variables manteniendo la información, haciendo posible el estudio de la planta con un grupo reducido de variables. En este trabajo se comparan tres modelos o métodos para la detección que utilizan esta técnica: un método lineal, uno no lineal y un modelo no lineal mejorado que busca amplificar el fallo para facilitar la detección. Por otro lado, y con el objetivo de permitir una clasificación o diagnóstico de los fallos producidos y detectados en la planta, se usarán técnicas de Redes Neuronales y Árboles de Decisión. A partir de estas herramientas se desarrollarán también tres métodos para la clasificación de los fallos: un método basado en redes neuronales con una única capa oculta (Machine Learning), un método que utiliza redes con varias capas ocultas (Deep Learning), y otro método hace uso de los árboles de decisión para crear un bosque aleatorio capaz de clasificar los fallos en clases. Tanto los métodos de detección como los de clasificación van a aplicarse a datos obtenidos de un colector de aguas simulado en el entorno de simulación visual SIMULINK, que funciona dentro del software MATLAB.

Palabras clave: BigData, Análisis de Componentes Principales (PCA), Redes Neuronales, Autoencoder, Árboles de Decisión, Machine Learning, Deep Learning.



ABSTRACT

The purpose of this project is to achieve an improvement in the monitoring and the quality of an industrial process. Therefore, different techniques based on detection and diagnosis of anomalies based on data are exposed. This is possible thanks to a Big Data technology that, through a massive implementation of sensors and distributed control in the industrial plants, has enabled to obtain a large amount of data that can be studied and used in many different ways. The statistic process control techniques that are used in this project are, firstly, the Principal Component Analysis, that allows to reduce the number of variables. Three models and methods of detection are compared. These models use a specific technique: a lineal method, a non lineal one and a non lineal improved model which seeks to amplify the error to facilitate its detection. On the other hand, and with the objective of allowing a clasification and diagnosis of the errors produced and detected in the industrial plants, Artificial Neural Networks and Decision Trees techniques are used. With these tools three methods will be developed for the classification of these faults: a method based on neural networks with a hidden layer (Machine Layer), a method that uses networks with various hidden layers (Deep Learning), and another method which uses the Decision Trees to create a random forest to classify the errors in classes. Both the detection and the classification methods are going to be used on data obtained from a water collector simulated in a environment of visual simulation SIMULINK, that works inside the software MATLAB.



CAPÍTULO I: INTRODUCCIÓN Y OBJETIVOS



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES



1.1 INTRODUCCIÓN

En un proceso productivo, el control de calidad es uno de los aspectos más importantes, pues de él depende la aceptación o rechazo del producto final referido tanto a la norma como a las especificaciones. Además, un control de calidad ineficiente puede provocar pérdidas económicas y daños humanos y/o medioambientales, entre otros. Para la optimización de un control de calidad, son imprescindibles tanto una rápida detección de los fallos en la planta o proceso, como un diagnóstico eficaz de los mismos. Debido a su importancia, estas técnicas de detección y diagnóstico de fallos resultan obligatorias en cualquier planta industrial.

La implantación de las nuevas tecnologías ha digitalizado los procedimientos que forman el ecosistema industrial, haciendo que la cadena de valor sea más eficiente. Todo esto ha sido posible gracias a tecnologías como las arquitecturas BigData o técnicas basadas en inteligencia artificial como el Deep y Machine Learning. Con estas nuevas tecnologías, es posible identificar parámetros críticos en cualquier fase de producción, permitiendo incluso desarrollar modelos predictivos para detectar posibles anomalías. Sensores repartidos por toda la planta industrial son los principales culpables de hacer esto posible, ya que su función es recopilar los datos con los que se realiza el proceso.

La monitorización de los procesos en la industria moderna ha permitido un aumento enorme de la seguridad en la producción, ya que gracias a ella se han podido desarrollar técnicas para la detección de fallos o anomalías en el proceso. Algunas de estas técnicas se basan en el Control Estadístico de Procesos (CEP) y han sido utilizadas en este trabajo, como el Análisis de Componentes Principales (PCA). Esta técnica multivariable extrae información de un conjunto de datos al que se le quiere aplicar una reducción de dimensionalidad.

Además de técnicas para la detección de fallos, también se han podido desarrollar técnicas para el diagnóstico de estas anomalías, con el objetivo de



identificarlas y poder encontrar el origen y la solución a un problema o fallo determinado. En el presente trabajo se han desarrollado diversos métodos para la clasificación de fallos, basados en redes neuronales y árboles de decisión, y así poder sacar conclusiones sobre la idoneidad o la eficacia de cada método en función de la situación.

1.2. OBJETIVOS

Este trabajo tiene como objetivo el desarrollo de técnicas de detección y diagnóstico de fallos basadas en datos, y el análisis, estudio y comparación de los resultados obtenidos para cada tipo de técnica. Para desarrollar las técnicas relacionadas con la detección se han utilizado herramientas del Control Estadístico de Procesos, como el Análisis de Componentes Principales y técnicas basadas en inteligencia computacional como los Autoencoders. Y para el desarrollo de técnicas enfocadas a la clasificación de fallos, se ha recurrido a la inteligencia artificial mediante redes neuronales, Autoencoders y árboles de decisión.

Todos los métodos han sido desarrollados y validados en el colector de aguas descrito en el Capítulo III.

1.3. ORGANIZACIÓN

La organización de la memoria del trabajo estará ordenada de la siguiente manera:

En el Capítulo I se explica de forma breve y contextualizada los métodos utilizados para la realización del trabajo, además de enumerar los objetivos que motivan el desarrollo del proyecto y un esquema de la distribución de la memoria.



En el Capítulo II se proyecta todo el contenido teórico alrededor del cual giran todos los procedimientos llevados a cabo en el proyecto. Términos como el control de calidad, el Análisis de Componentes Principales, las Redes Neuronales o los Árboles de Decisión son desarrollados en profundidad.

En el Capítulo III se introduce la planta de la que se han extraído todos los datos necesarios para los cálculos llevados a cabo. Se trata de un colector de aguas con 24 variables monitorizadas, de las cuales se extraerá la información necesaria a través de los métodos desarrollados.

En el Capítulo IV se incluyen toda la metodología, los cálculos y los resultados para cada método desarrollado. Los resultados son representados con gráficas, tablas y matrices de confusión.

El Capítulo V aparecen las conclusiones a las que se ha llegado tras la realización del trabajo, además de la explicación y la interpretación de todos los resultados obtenidos. Además, se sugieren ideas para futuros trabajos y posibles conceptos que aporten mejores resultados.

Al final del trabajo se incluyen todas aquellas referencias bibliográficas de las que se ha hecho uso para el desarrollo del presente trabajo.



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES



CAPÍTULO II: MARCO TEÓRICO



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES



2.1. CONTROL DE CALIDAD

La gestión de la producción engloba todas aquellas técnicas y métodos que tienen el fin de convertir las materias primas en productos elaborados en un proceso industrial. Esta gestión debe encontrarse en continua optimización para que el producto final cumpla los estándares de calidad, cantidad, tiempo y coste mínimo [1]. A raíz de esta idea surge el término PDCA (Plan, Do, Check, Action) hace más de 80 años, y que consiste en un método ciclo que garantiza la atención continua sobre el control de calidad. Se evalúa el proceso, se aplican las acciones correctivas pertinentes, y se vuelve a evaluar.

En la actualidad, una empresa que quiera seguir siendo competitiva deberá hacer un gran esfuerzo de adaptación, especialmente en el aspecto tecnológico y de gestión, y será la rapidez con la que se consiga esta adaptación la que determinará la competitividad de la empresa. La globalización es el principal responsable de las modificaciones que están dándose en la cultura empresarial [2]. La referencia a la hora de diseñar un modelo de gestión por parte de una empresa siempre será la normativa que haya que cumplir. Las normas ISO suelen ser las normas a cumplir en la actividad industrial, aunque existen algunos mercados como el asiático que pueden seguir otras pautas. En relación a las normas ISO, la norma que regula la gestión industrial y el control de calidad es la ISO 9001, la cual especifica los requisitos para un sistema de gestión de calidad, concretamente cuando una empresa necesita demostrar su capacidad para proporcionar los productos o servicios exigidos por el cliente, o cuando la empresa desea aumentar la satisfacción del cliente incluyendo procesos de mejora del sistema y asegurando la conformidad a los requisitos legales y del cliente [3]. La consecuencia de una buena implantación de un sistema de gestión es la ventaja competitiva de la empresa respecto de las demás dentro del mismo ámbito, que se traduce en reducción de tiempos y costes, mejora de la imagen de la empresa, satisfacción de los clientes y posibilidad de conseguir nuevos, y en general, un mayor nivel de productividad [4].



La monitorización es una etapa del proceso de gestión de calidad que proporciona la información necesaria para conocer el funcionamiento del proceso y poder controlarlo. Esta información se emplea en la detección de fallos o anomalías, lo que posibilita corregirlas de manera rápida con las herramientas y medidas correctivas correspondientes, haciendo que el proceso vuelva a su funcionamiento normal. Para que toda la información del funcionamiento del proceso pueda ser extraída y procesada, son necesarias una serie de herramientas como los sensores, los actuadores y los controladores lógicos programables (PLC). Todos estos elementos se encuentran interconectados, permitiendo que la información fluya de forma continua. Esta monitorización es compleja, y solo ha sido posible gracias a los numerosos y recientes avances en los Sistemas de Control Distribuido (Distributed Control System, DCS), y a la hiperconectividad entre máquinas de la Industria 4.0. Como se ha dicho anteriormente, uno de los objetivos principales de la monitorización del proceso es la detección y diagnóstico de fallos (FDD). Un fallo puede ser una condición anormal que provoca un mal funcionamiento del proceso no deseado, aunque también puede ser un defecto en un componente del equipo que evita que se cumplan las especificaciones o que se ponga en peligro los productos o las personas. La detección del fallo va seguida de la clasificación o diagnóstico del mismo, que consiste en identificar el tipo de fallo, la gravedad que provoca o dónde se encuentra la anomalía [4]. La última etapa sería la recuperación tras el fallo, en la que se soluciona el problema y el proceso vuelve a su funcionamiento normal.

Si la calidad se cuantifica como la probabilidad de que el producto final se encuentre dentro de los límites de calidad establecidos por los requisitos correspondientes, una variabilidad alta afecta negativamente a la calidad. Por lo que, definiendo un punto de funcionamiento en un proceso, los fallos o anomalías serán identificados como índices de variabilidad elevados respecto a dicho punto.



2.2. CONTROL ESTADÍSTICO DE PROCESOS

El Control Estadístico de Procesos (CEP) o *Statistical Process Control* (SPC) está constituido por un conjunto de técnicas que tienen la función de examinar la calidad de un proceso productivo. Los objetivos de un CEP guardan relación con lo explicado anteriormente, y entre los cuales destacan los siguientes: minimizar la producción defectuosa, mantener una actitud de mejora continua del proceso, y mantener la producción dentro de los límites marcados por las especificaciones. Para que estos objetivos se puedan alcanzar, se debe tener en cuenta que los procesos, además de generar un producto, generan información. Y es esta información la que, recogida gracias a la monitorización, permite “escuchar” el proceso. Además, las técnicas de CEP deben ser aplicadas lo más próximas posible al proceso generador de la información, de tal manera que el tiempo de reacción ante el proceso sea lo más pequeño posible [5].

2.2.1 VARIABILIDAD EN EL PROCESO DE PRODUCCION

La variabilidad es una característica inherente del proceso, pues nada puede hacerse para suprimir las variaciones en una planta, pero lo que si se puede y se debe hacer es investigar las causas que producen la variabilidad con el objetivo de minimizarla.

Para determinar la variación de un proceso, se toma una serie de muestras pequeñas y se las coloca en una escala proporcional, en el eje horizontal (Véase la Figura 1.a)). En el eje vertical se indica la frecuencia o el número de veces que ocurrieron. Eventualmente, se consiguen distribuciones como las de la Figura 1.b), tras un número determinado de muestras. Estas distribuciones irán cambiando en función de lo que revelen las muestras (Figura 1.c)). Si la distribución resultante es similar a la de la Figura 1.d), significará que solo se han encontrado causas de variaciones naturales, que se explicarán más adelante. Para encontrarse con causas de variaciones asignables, que son

aquellas que no se esperan como parte del proceso, las distribuciones creadas por las muestras serán inesperadas (mostradas en la Figura 1.e)) [6].

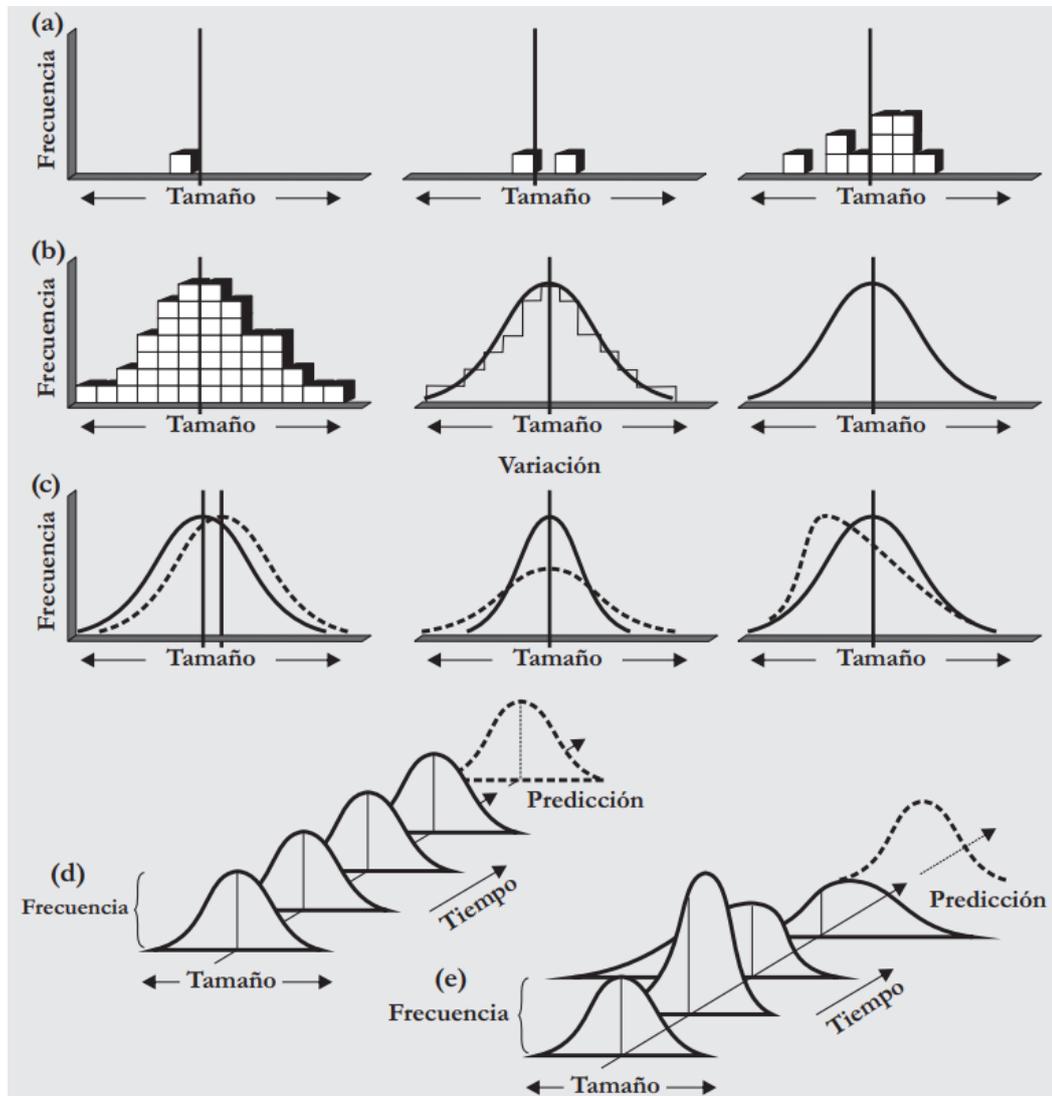


Figura 1. Variaciones naturales y asignables.

La gestión del proceso debe conseguir que las variaciones asignables sean eliminadas, de tal manera que el proceso se encuentre bajo control. Que el proceso se encuentre o no bajo control lo determina la salida. En la figura 2 se representan 3 posibles situaciones de la salida (a), b), c). En la salida a), el proceso se encuentra fuera de control. Sin embargo, en b), el proceso está controlado pero descentrado respecto de los límites de especificación de producto. El caso c) representa un proceso controlado correctamente y dentro de las especificaciones.

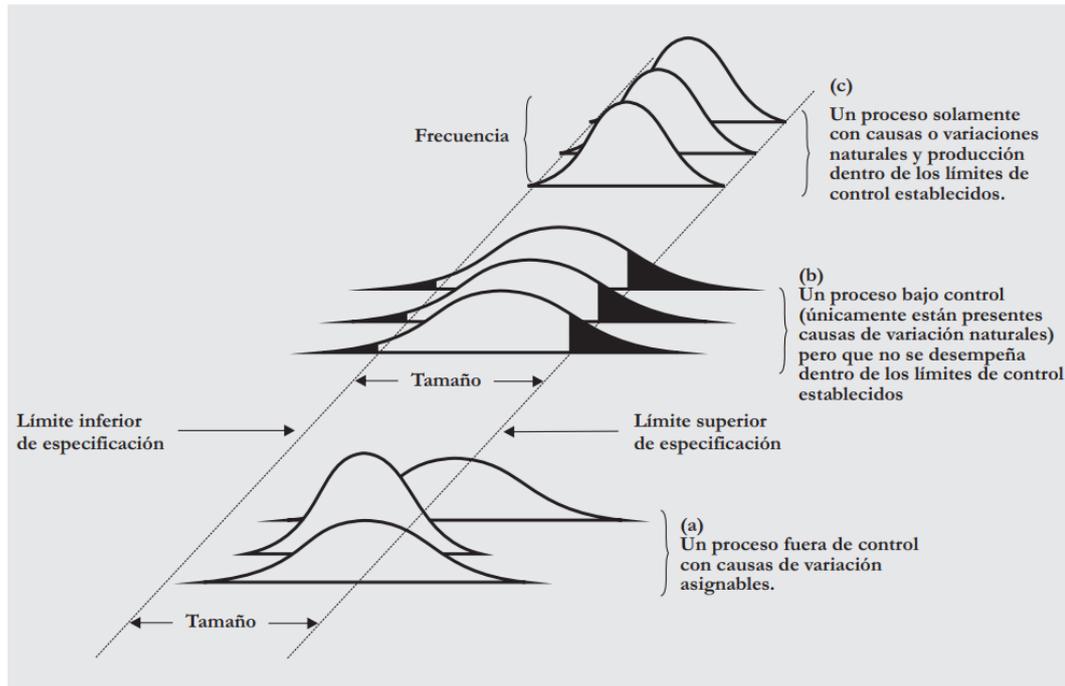


Figura 2. Tipos de salida del proceso.

2.2.2. CAUSAS DE LA VARIABILIDAD

Como se ha comentado anteriormente, las causas de la variabilidad de la salida de un proceso se pueden dividir en dos tipos:

Causas comunes: Son fuentes de variación puramente aleatorias, inidentificables e imposibles de evitar. Si la variabilidad de un proceso estuviera únicamente compuesta por causas comunes, la distribución del proceso sería simétrica, es decir, presenta el mismo número de observaciones por encima y por debajo de la media.

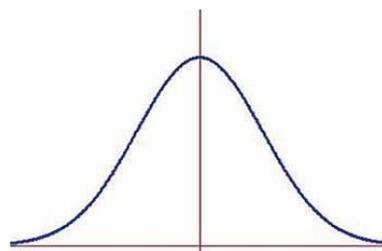


Figura 3. Distribución de proceso simétrica.

Causas asignables: Son fuentes de variación en las que cualquier factor causante de dicha variación puede ser identificado y eliminado. Estas causas pueden llegar a romper la simetría de la distribución, por lo que la gestión del proyecto es la encargada de identificar las causas asignables y eliminarlas. En la Figura 4 se muestra un ejemplo de los posibles efectos que puede tener las causas asignables sobre la simetría de una distribución [6].

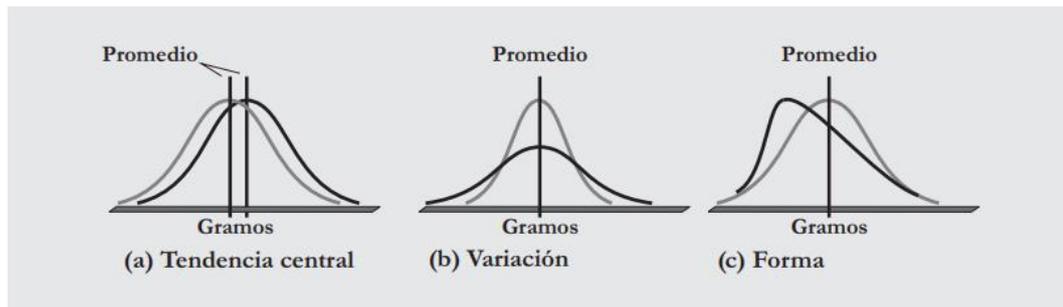


Figura 4. Ejemplo de los efectos de las causas asignables sobre la distribución de un proceso.

Una vez resuelto este problema, el proceso vuelve a estar bajo control, y serán las técnicas del Control Estadístico de Procesos, en un proceso de mejora continua, las encargadas de reducir la variabilidad obtenida. La Figura 5 muestra el cambio que sufre la distribución al ser eliminadas las causas asignables.

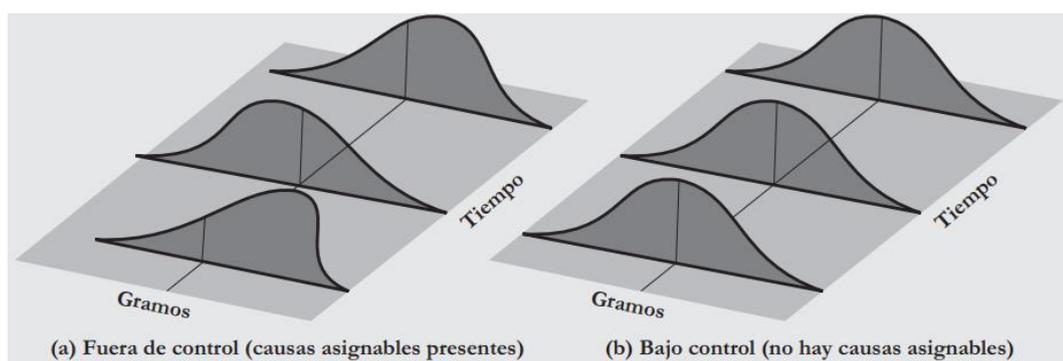


Figura 5. Ejemplo del efecto de las causas asignables sobre la distribución de un proceso.

2.2.3. GRAFICOS DE CONTROL

Los gráficos de control son una herramienta de análisis y solución de problemas. Se compone de un diagrama que muestra los valores producto de la medición de una característica de calidad. Referido al trabajo realizado, esta herramienta permite observar de forma gráfica el sistema que se analiza, y poder realizar una detección de las anomalías más sencilla. En la gráfica se incluyen datos del desempeño del proceso, y pueden ser continuos o discretos. En este caso se trata de datos continuos, y este tipo de datos se grafican junto a tres líneas horizontales que se pueden observar en el ejemplo que muestra la Figura 6 [8]: valor medio o línea central (\bar{p}), límite superior de control (LSC) y límite inferior de control (LIC). Los límites superior e inferior de control (LSC y LIC) son límites que, en caso de ser sobrepasados, el proceso probablemente haya perdido estabilidad estadística y se encuentre probablemente fuera de control. Este suceso tiene como consecuencia que sea preciso investigar y diagnosticar la causa de la perturbación, y llevar acciones correctivas una vez la anomalía hay sido clasificada [7].

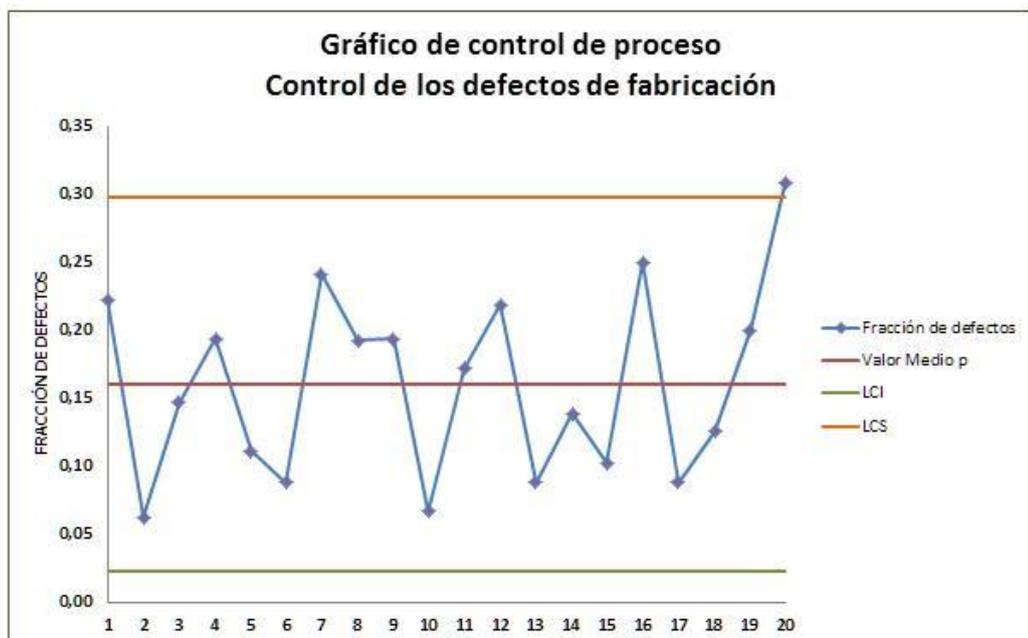


Figura 6. Ejemplo de diagrama de control para detección de anomalías.

En función del tipo de variables que se vayan a representar en el gráfico de control, existen diferentes tipos de estos diagramas. Debido a que las variables en este trabajo tienen carácter continuo, el tipo de gráfico que se va a utilizar son los gráficos R, que muestran la ganancia o pérdida de uniformidad de la dispersión dentro de una muestra. Para ello tienen en cuenta la diferencia entre el valor mayor y menor de dicha muestra y se calcula la variabilidad. Incluso cuando la media del proceso está bajo control, la variabilidad puede no estarlo. Los límites de control para este tipo de gráfica (rango) se calculan de la siguiente manera:

$$UCL_R = D_4 * R \quad (1)$$

$$LCL_R = D_3 * R \quad (2)$$

UCL_R = Límites e control superior para el rango.

LCL_R = Límites de control inferior para el rango

D_3 y D_4 = Constantes que dan 3 acotamientos de la desviación típica (3σ) para el rango dado.

Tamaño de la muestra (n)	Factor para UCL y LCL para gráficas \bar{x} (A_2)	Factor para LCL para gráficas R (D_3)	Factor para UCL para gráficas R (D_4)
2	1,880	0	3,267
3	1,023	0	2,575
4	0,729	0	2,282
5	0,577	0	2,115
6	0,483	0	2,004
7	0,419	0,076	1,924
8	0,373	0,136	1,864
9	0,337	0,184	1,816
10	0,308	0,223	1,777

Figura 7. Factores para calcular acotamientos 3σ para la gráfica R [6].

2.3. Análisis de Componentes Principales (PCA)

2.3.1. Introducción

Uno de los objetivos de este trabajo expuestos anteriormente era poder detectar los fallos o anomalías de una planta o proceso productivo, y para ello se ha recurrido a una técnica estadística de análisis de componentes principales.

El Análisis de Componentes Principales (PCA) es una de las técnicas de análisis multivariable más antiguas, introducida por Pearson en (1901), pero que realmente fue desarrollada y aplicada por Hotelling (1933). Estas técnicas de análisis multivariable necesitan gran capacidad y potencia de cálculo, por lo que su desarrollo ha ido acompasado con la mejora exponencial de los ordenadores y máquinas de cálculo [9]. Su uso se ha centrado en la predicción, detección y diagnóstico, y es utilizado como primer paso en análisis de datos de muchos tipos, como medida de comprobación de hipótesis, o para identificar “*outliners*” en el conjunto de datos.

2.3.2. Objetivos del PCA

El objetivo principal de esta técnica de análisis multivariable es reducir la dimensionalidad del conjunto de datos, permitiendo trabajar con un número de variables menor que en el origen, pero sin perder información. De esta manera, se divide el espacio en dos subespacios diferentes: uno captura la tendencia del proceso y otro obtiene el ruido. Al graficar todas las dimensiones, el conjunto de datos adopta forma de elipsoide, cuyos semiejes mayor y menor representan la tendencia del proceso y el ruido respectivamente (Figura 8). Otro objetivo de esta técnica es el de identificar nuevas variables significativas. El PCA siempre va a identificar nuevas variables, pero no garantiza que estas sean significativas, aunque pueden ser útiles en otros análisis [10].

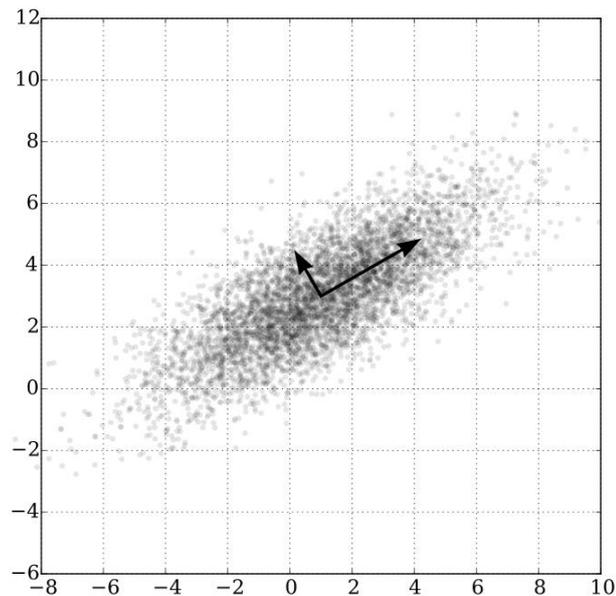


Figura 8. Representación de las m dimensiones del conjunto de datos.

2.3.3. Desarrollo matemático del método PCA

Los datos obtenidos del proceso forman una matriz de n muestras y m variables, y con la técnica PCA se consigue reducir m y poder representar el sistema en un subespacio de dimensión menor. Al conjunto de datos del proceso se le somete a un pretratamiento antes de aplicarle PCA. Este pretratamiento consta de dos etapas. Primero se eliminan aquellas variables que, debido a múltiples factores como pueden ser errores en los sensores o los instrumentos de medida, tienen valores erróneos que dificultarían el buen funcionamiento de la técnica PCA [9]. Después se normaliza las m variables a media 0 y varianza 1, con el objetivo de asegurar que todas las variables tienen el mismo peso e influencia en la monitorización del proceso.

Tras concluir el pretratamiento, se puede empezar a aplicar el método al conjunto de datos del proceso. Del proceso de pretratamiento se obtiene una matriz de datos X cuyas dimensiones son $n \times m$.

$$X = \begin{bmatrix} X_{11} & \cdots & X_{1m} \\ \vdots & \ddots & \vdots \\ X_{n1} & \cdots & X_{nm} \end{bmatrix} \quad (3)$$

Para obtener los vectores propios ortogonales, y que estén ordenados en función de la variabilidad que aportan al sistema, se calcula la matriz de correlación R a partir de la matriz X (3):

$$R = \frac{1}{n-1} X^T X \quad (4)$$

Después se calcula la descomposición de R en valores singulares:

$$R = V * \Lambda * V^T \quad (5)$$

Donde Λ es un matriz compuesta de los valores propios reales no negativos de R ordenados en orden decreciente: $(\lambda_1 \geq \lambda_2 \cdots \geq \lambda_m \geq 0)$, donde λ_i es la varianza:

$$\lambda_i = \sigma_i^2 \quad (6)$$

El criterio de elección del número de componentes principales va a consistir en elegir una variabilidad mínima aceptada para el método. De esta manera se seleccionan los a vectores propios que correspondan a los valores propios más grandes dentro de ese valor de variabilidad, y se almacenan en una matriz P de dimensión reducida. Ahora $P \in \mathfrak{R}^{m \times a}$, y forma la matriz de vectores (loadings). Con la matriz P y la matriz de datos X, se procede a proyectar los datos iniciales en un espacio de dimensión reducida, creándose la matriz de transformación T, tal que $T \in \mathfrak{R}^{n \times a}$. Los valores que componen la matriz T se los denomina scores.

$$T = X * P \quad (7)$$

A partir de P y T se puede volver al espacio de partida:

$$\hat{X} = \sum_{i=1}^a t_i p_i^T \quad (8)$$

y la matriz de ruido o residuos E se forma a partir de la diferencia entre X y \hat{X} :

$$E = X - \hat{X} \quad (9)$$

Esta matriz de residuos E permite obtener la variabilidad de los datos de observación a través de los vectores propios que se encuentran asociados a los valores singulares m -a de menos valor, y es posible representar el conjunto de datos de partida:

$$X = \hat{X} + E \quad (10)$$

2.3.4. Estadísticas para la monitorización

Para la monitorización del proceso a partir del PCA, se usan distintos estadísticos que permiten construir gráficos de control para supervisar el proceso. Así se consigue monitorizar el proceso con una única variable, a la vez que se muestra el umbral de normalidad en el mismo gráfico [4].

2.3.4.1. Estadística Hotelling (T^2)

Este estadístico utiliza el espacio de dimensión a . A partir de la matriz X normalizada con media 0 y varianza 1, y $x \in \mathfrak{R}^{mx1}$ se puede definir la estadística T^2 :

$$T^2 = x^T P \Lambda_a^{-1} P^T x \quad (11)$$

Donde Λ_a es una matriz con las primeras a columnas y filas de la matriz Λ , siendo a el número de componentes principales elegido. También se puede representar T^2 en su forma genérica, introduciendo la ecuación (7) en (11):

$$T^2 = T * \Lambda_a^{-1} * T \quad (12)$$

Se forma a partir de los autovectores que forman el subespacio de la variación imperante, por lo que este estadístico representa la variación del proceso. Sin embargo, es necesario un límite para determinar si el proceso se encuentra

fuera de control, y este umbral asociado al estadístico Hotellings se determina con el número de datos de muestreo n :

$$T_{\alpha}^2 = \frac{(n^2)*a}{n*(n-a)} F_{\alpha}(a, n - a) \quad (13)$$

En la ecuación (13) $F_{\alpha}(a, n - a)$ es el valor crítico de la función F de Fisher-Snedecor, n y $n-a$ son los grados de libertad de la función, y α es el nivel de significancia. El valor de α supondrá la tolerancia que se tendrá a las falsas alarmas, y sus valores más frecuentes están entre 0,01 y 0,05.

Debido a que T^2 se basa en los a primeros componentes principales, ofrece la medida de las desviaciones de las variables latentes, que son aquellas que mejor explican la variabilidad de la planta. El inconveniente de esto es que detectará únicamente los eventos que produzcan mayor variabilidad en las variables latentes que la explicada por las causas comunes. Pero anomalías que produzcan variaciones en el ruido no se detectarán, ya que el espacio del ruido ha sido descartado para el cálculo de este estadístico [10].

2.3.4.2. Estadístico SPE o Q

Este estadístico soluciona el problema de la detección de eventos en el espacio del ruido que tenía T^2 . Las siglas SPE significan Error de Predicción Cuadrático en español. Se calcula a partir de un vector de observación $x \in \mathfrak{R}^{m \times 1}$:

$$Q = r^T r \quad (14)$$

donde r es el vector residuo:

$$r = (I - PP^T)x \quad (15)$$

El estadístico indica la diferencia (residuo r) entre el vector de observación x y su valor proyectado en el espacio reducido de los componentes principales. Al igual que ocurría con el estadístico anterior, Q necesita de un umbral para

determinar si el proceso se encuentra fuera de los límites de control. Este umbral se calcula con los datos tomados fuera de línea para el PCA [10]:

$$Q_{\alpha} = \frac{v}{2 * \mu} * x_{\alpha}^2 (1 - \alpha, h)^{-1} \quad (16)$$

Donde $x_{\alpha}^2 (1 - \alpha, h)^{-1}$ es la inversa de a función acumulativa x^2 , α es el índice de tolerancia a falsas alarmas, μ y v son la media y la varianza de Q, y el parámetro h se calcula de la siguiente manera:

$$h = 2 * \frac{\mu^2}{v} \quad (17)$$

El estadístico Q representará fluctuaciones en el ruido siempre que el proceso esté controlado. Si la covarianza de X cambia, Q superará el umbral y el fallo será detectado.

2.4. TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

Tras la detección de un fallo en un proceso, se lleva a cabo un estudio para diagnosticar el fallo, y de esta manera se pueda identificar la causa y el origen de la anomalía detectada. Para efectuar esta clasificación se utilizarán modelos computacionales como las redes neuronales, que mediante un aprendizaje automático se encargarán de diagnosticar cada fallo, permitiendo la identificación de cada uno. Dentro del aprendizaje automático se diferencian dos ramas de la inteligencia artificial: *Machine Learning* y *Deep Learning*. La primera estudia técnicas para el desarrollo de algoritmos que se automejoren de forma automática. Por otro lado, el *Deep Learning* se centra en crear largos y potentes modelos de redes neuronales que sean capaces de tomar decisiones con gran precisión a través de datos reales [11].

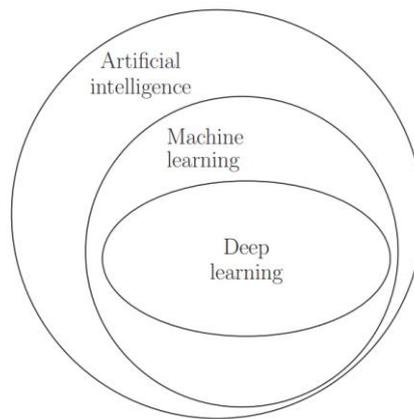


Figura 9. Relación entre la Inteligencia Artificial, Machine Learning y Deep Learning [11].

2.4.1. REDES NEURONALES ARTIFICIALES

En los últimos años las Redes Neuronales Artificiales (ANN) han recibido un gran interés para minar datos, ya que se pueden modelar grandes y complejos problemas con los medios que ofrece. Son capaces de encontrar patrones de forma inductiva gracias a algoritmos de aprendizaje basados en datos existentes [12]. Las ANN representan un método de resolución de problemas de clasificación, identificación o predicción, siempre que haya cierta tolerancia a errores y se necesite de un aprendizaje mientras se ejecuta el método.

La unidad de una red neuronal artificial es la neurona, que se define como un procesador elemental con capacidad limitada para el cálculo de, generalmente, la suma ponderada de las entradas que llegan a la neurona, y ejecuta una función de activación que produce la señal que será enviada al resto de neuronas. El modelo de una neurona se compone de un vector de pesos \vec{w} (18) que equivale a las conexiones sinápticas en una neurona real, w_0 es el umbral de activación o bias, la entrada la forma el vector \vec{x} y el escalar y es la salida (19).

$$\vec{w} = (w_1, \dots, w_n)^T \quad (18)$$

$$y = \gamma[\sum_{i=1}^m w_i x_i + w_0] \quad (19)$$

Donde γ es una función no lineal.

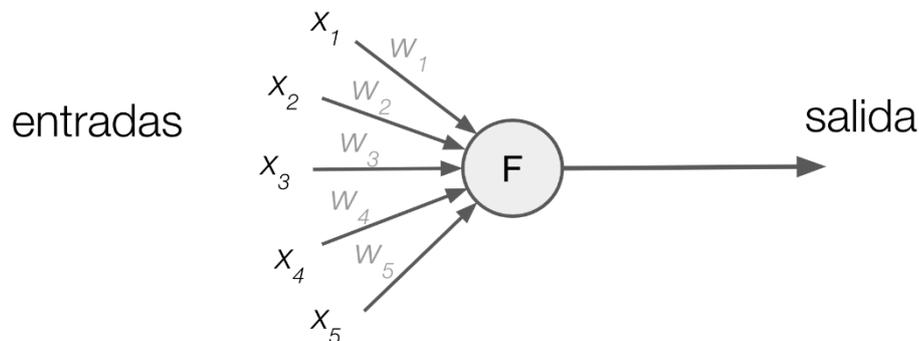


Figura 10. Ejemplo de neurona artificial con 5 entradas [12].

Las neuronas se agrupan en capas y tienen una alta conectividad entre ellas. Para su correcto funcionamiento, las neuronas son sometidas a un aprendizaje automático mediante un algoritmo, que permite que puedan minimizar el error de ajuste de datos cambiando su estructura o sus parámetros.

Las Redes Neuronales Feedforward (FANN) son el tipo de ANN más estudiado por la comunidad científica, y sus usos son numerosos dentro de amplios campos de aplicación. Su nombre indica que es alimentada hacia adelante, es decir, la información fluye desde las entradas hacia las salidas [13]. En cuanto a la arquitectura, una FANN está compuesta por neuronas divididas en diferentes capas (Figura 11), y de las que hay 3 tipos: capa de entrada, capa oculta (puede ser única o puede haber varias) y capa de salida. Una vez se define la arquitectura de la red con el número de capas ocultas y el número de neuronas en cada capa, la red se ajusta a una muestra dada a través de procesos de aprendizaje explicados más adelante.

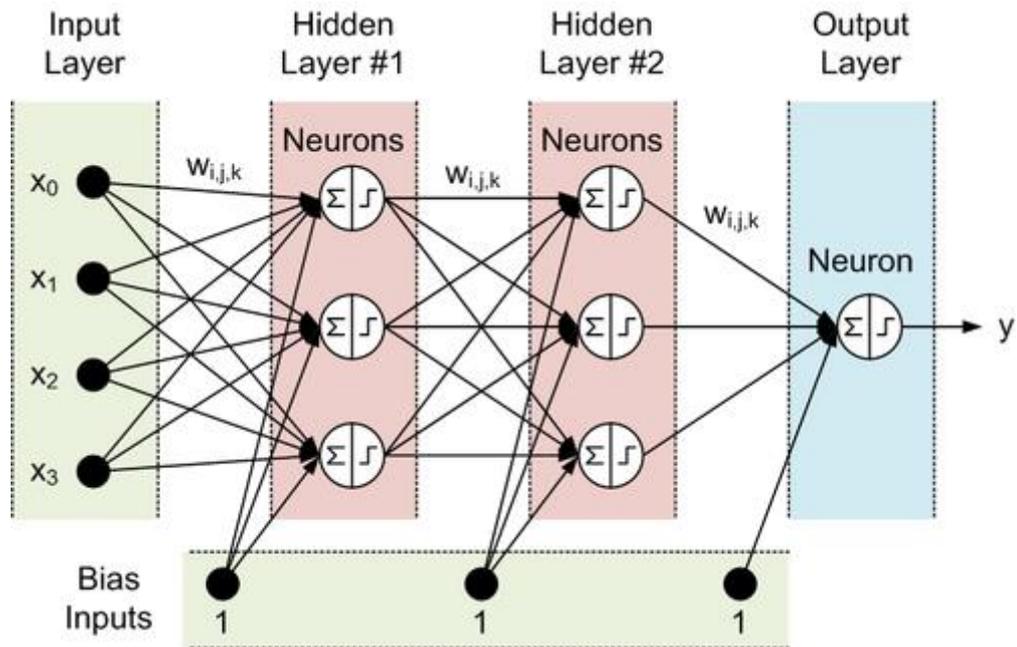


Figura 11. Arquitectura de una Red Neuronal Feedforward con dos capas ocultas [14].

En el caso de que el número de capas ocultas sea mayor de uno, la red neuronal será de aprendizaje profundo o *Deep Learning*. Este subcampo de la inteligencia artificial se centra en crear largos modelos de redes neuronales que son capaces de tomar decisiones con gran precisión a través de datos reales.

2.4.2. APRENDIZAJE AUTOMÁTICO

El aprendizaje automático, también llamado *machine learning*, es una rama de la Inteligencia Artificial que estudia técnicas para el desarrollo de algoritmos que permitan que los modelos se mejoren a sí mismos de forma automática. El objetivo es que estos modelos sean capaces de poder hacer predicciones y tomar decisiones a partir de datos reales.

Dentro del aprendizaje automático, existen dos tipos principales:

- **Aprendizaje automático supervisado:** Se caracteriza por la presencia de un agente externo que controla el proceso de entrenamiento,



estableciendo la respuesta que debería generar la red (output), a partir de una entrada determinada. De esta forma se compara la salida de la red con la salida deseada, y las diferencias que existan se ajustan iterando los pesos hasta que la salida tiende a ser la deseada [15].

- **Aprendizaje automático no supervisado**: Las redes con este tipo de aprendizaje no necesitan información externa para el ajuste de los pesos. La red utiliza el algoritmo de aprendizaje para detectar peculiaridades o correlaciones en el conjunto de entradas, y así estima la función de densidad probabilística $p(x)$ [15].

2.4.3. AUTOENCODERS

Un Autoencoder es un tipo de red de propagación hacia adelante formado por 3 capas. La peculiaridad de estas redes es que la salida tiene el mismo valor que la entrada a la red, lo que significa que la tarea del autoencoder es la de recrear las entradas. En este proceso de recreación, el Autoencoder tendrá que ser capaz de, mediante un codificador, reducir la dimensionalidad manteniendo las características que definen la entrada, y de realizar a continuación la operación inversa mediante un decodificador. El tipo de aprendizaje de un Autoencoder será por tanto no supervisado, ya que no hay que dotar a la red de la salida deseada. Todo esto obliga a que la capa de salida tenga el mismo número de neuronas que la capa de entrada.

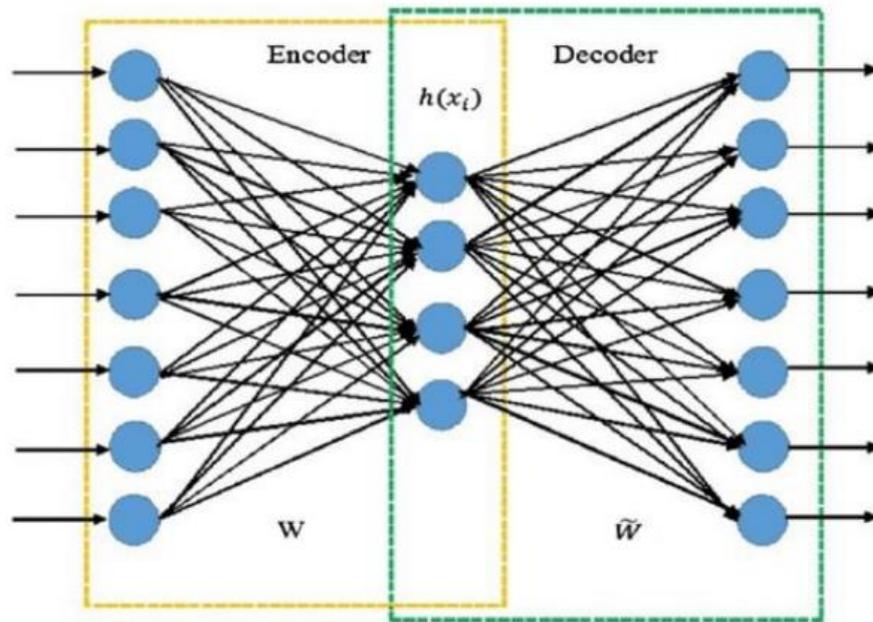


Figura 12. Arquitectura de un Autoencoder [16].

Existe un problema con la arquitectura de estas redes, y es que, si el número de neuronas en la capa oculta es igual o mayor al número de neuronas de las capas de entrada y salida, el Autoencoder corre el peligro de aprender la función de identidad en vez de aproximarla, por lo que aparece el problema de sobreajuste. En la figura 13 se ilustra este problema, donde la curva negra representa una red sobre ajustada, y la recta azul se corresponde con una red mejor ajustada, que tendrá menor precisión, pero mucha mayor flexibilidad a datos nuevos [17].

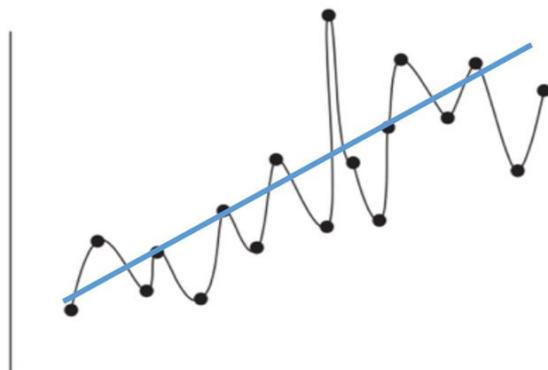


Figura 13. Sobreajuste de una red [17].

La solución a este problema es incluir la condición de que tiene que haber menos neuronas en la capa oculta que en las capas de entrada y salida del Autoencoder. Los Autoencoders que cumplen esta condición se denominan Autoencoders Simples.

2.4.3.1. AUTOENCODERS APILADOS (STACKED AUTOENCODERS)

Se pueden crear redes neuronales con varias capas ocultas a partir del apilamiento de varios Autoencoders. Sabiendo que el resultado real del Autoencoder no se encuentra en la capa de salida sino en la capa oculta, se puede usar esa capa oculta como entrada de otro Autoencoder, de manera que la unión de los Autoencoders no es uno detrás de otro, sino que sus capas se encuentran combinadas, como refleja la Figura 14.

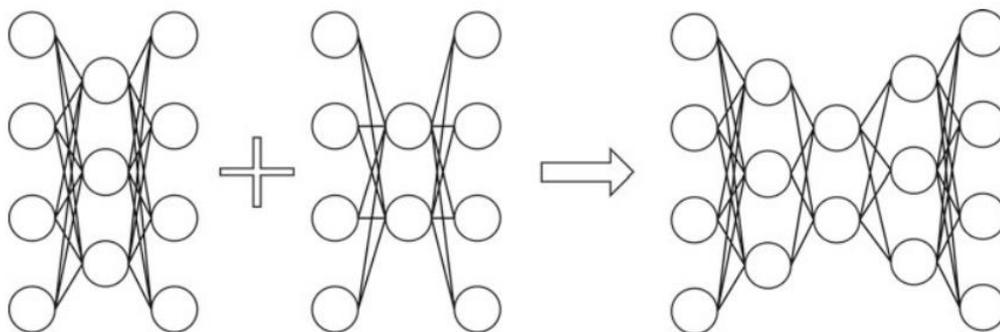


Figura 14. Combinación de 2 Autoencoders [18].

Cabe destacar que, aunque se combinen los Autoencoders, el número de neuronas a la entrada y a la salida de la red seguirá siendo iguales. Solo la capa intermedia y la arquitectura del Autoencoder se verá modificada. La arquitectura formada de cuello de botella es ideal para la función que desempeña el Autoencoder, pues la codificación y decodificación se realiza de forma gradual. El resultado real del Stacked Autoencoder es la construcción de la representación distribuida por la capa intermedia [18].

2.4.4. ÁRBOLES DE DECISIÓN (RANDOM FOREST)

Un árbol de decisión es un modelo de predicción cuyo objetivo es el aprendizaje inductivo a partir de construcciones lógicas y observaciones. Probablemente sea el modelo para clasificación más utilizado. También representa una técnica de minería de datos (Data Mining) proveniente de la inteligencia artificial y la estadística, donde se encuentran técnicas como las redes neuronales o la regresión lineal

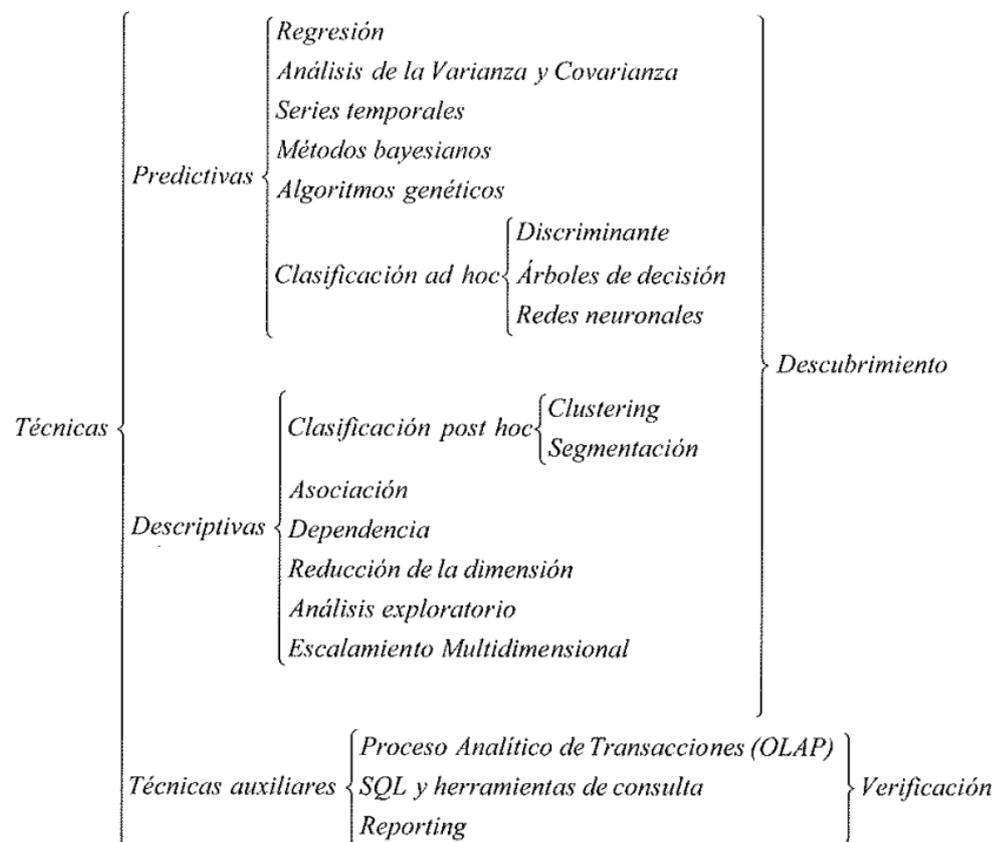


Figura 15. Esquema de la clasificación de las técnicas Data Mining [19].

Los árboles de decisión crean un modelo de clasificación basado en los diagramas de flujo, y clasifican casos en grupos o pronostican valores de una variable dependiente basada en variables independientes. Algunas ventajas de los árboles de decisión son la sencillez de interpretar la decisión, la reducción



del número de variables independientes o la forma en la que facilita la comprensión del conocimiento utilizado en la decisión [19].

El conocimiento obtenido durante el proceso de aprendizaje inductivo se representa mediante un árbol, que su representación gráfica se basa en un conjunto de nodos, hojas y ramas. Existe un nodo principal, también denominado raíz, que es el atributo del que se inicia el proceso de clasificación. Luego se encuentran los nodos internos, que corresponden a cada una de las preguntas sobre el atributo del problema. Después, cada posible respuesta a estas preguntas se representa con un nodo hijo. Cada rama que sale de estos nodos está etiquetada con los posibles valores del atributo. Los nodos hoja son los nodos finales, y se corresponden con una decisión que se corresponde con una de las variables clase del problema [20].

Para generar el algoritmo del árbol de decisión, el proceso se divide en dos etapas. Primero se construye el árbol a partir del conjunto del entrenamiento, y la parte del conjunto de entrenamiento que hay en cada nodo interno se divide en los valores que puede tener ese atributo. Para la creación del árbol se comienza por el nodo raíz, se elige un atributo de prueba y se divide el conjunto de entrenamiento en varios subconjuntos, y para cada uno se genera un nodo nuevo. Si un nodo tiene objetos de varias clases se genera un nodo interno. Si no, se forma una hoja con una etiqueta de clase. La segunda etapa está relacionada con el proceso de clasificación, donde los nuevos objetos son clasificados por el árbol previamente creado. Para la clasificación, se recorre el árbol desde la raíz hasta la hoja, donde se determina la clase del objeto. La elección del camino hasta la hoja depende de las decisiones tomadas en los nodos internos [20].

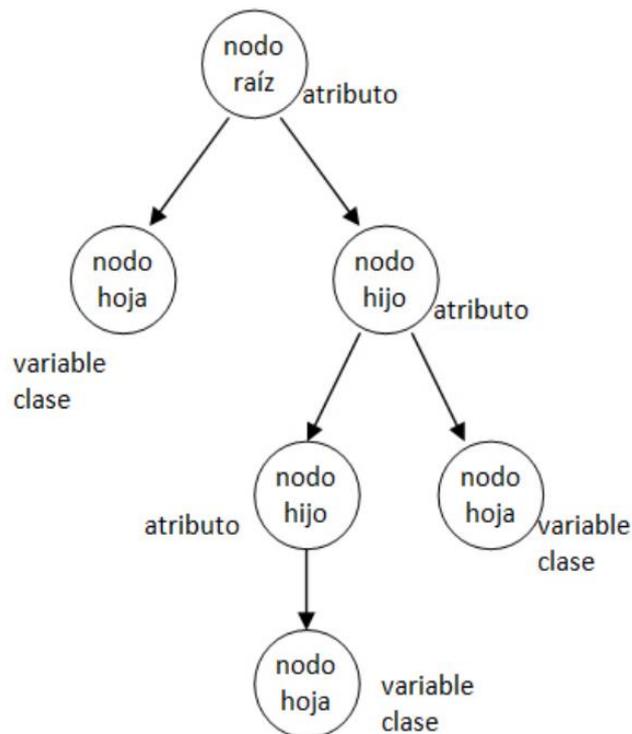


Figura 16. Esquema del árbol de decisión [20].

Los árboles de decisión se ensamblan entre sí formando lo que se denomina un bosque. Esto ayuda al procesamiento de todos los cálculos y previene el sobreajuste de los datos, lo que provocaría un mal funcionamiento del método de clasificación. Existen diferentes modos de ensamblaje, pero los más utilizados por la comunidad científica son: Boosting, Bagging y Random Forest. El algoritmo Boosting construye árboles de forma secuencial e intentando reducir el error del árbol anterior. Los pesos se distribuyen tras el entrenamiento de tal manera que los árboles con mayor error tienen pesos mayores, lo que provoca que en la siguiente etapa el algoritmo centra el entrenamiento en los árboles con mayor error.

El algoritmo Bagging se centra en reducir la varianza de un conjunto de árboles o bosque. Primero se crean subgrupos de los datos de entrenamiento, con las observaciones divididas aleatoriamente en estos subgrupos. Cada subgrupo se ocupará del entrenamiento de un grupo de árboles, y una vez finalizados los entrenamientos, se combinan los resultados con media o moda, si se trata de un problema de regresión o de clasificación.



Los bosques aleatorios surgen del algoritmo Bagging. La ventaja de este algoritmo respecto a los demás es que crea árboles con poca correlación entre ellos. De esta manera, los errores de los árboles son individuales y no condicionan al resto que forman el bosque. Además, tienen un buen funcionamiento con grandes cantidades de datos y variables, debido al poco efecto que tiene el ruido sobre el algoritmo. Estas características colocan a este método como uno de los más utilizados para el diagnóstico de fallos. [17]

2.4.5. MATRIZ DE CONFUSIÓN

La matriz de confusión es una herramienta que facilita la visualización y la evaluación de la eficacia en la clasificación de las anomalías de un proceso. La matriz se encuentra en el software MATLAB, y se usa principalmente tras un aprendizaje automático supervisado.

Las filas que componen la matriz se corresponden con cada una de las clases de salida, mientras que las columnas representan las clases deseadas. La diagonal de la matriz estará formada por tantos elementos como clases haya (matriz cuadrada), y las celdas que pertenecen a la matriz se denominan celdas de acierto. Si los datos se encuentran dentro de esta celda, estarán bien clasificados. Por el contrario, si están fuera de la diagonal de la matriz, la celda se corresponderá con una celda de fallo, y esos datos estarán clasificados erróneamente.

La matriz tiene una fila y una columna que recogen determinada información del total. La última fila muestra todas las observaciones para las clases objetivo, con el porcentaje de verdaderos positivos (sensibilidad) y falsos negativos [4]. La última columna muestra todas las observaciones para las clases de salida, con el porcentaje de aciertos detectados (precisión) y de falsos descubrimientos. La última celda de la diagonal de matriz contiene la información del porcentaje de predicciones correctas y de las incorrectas. Es el valor más importante que aporta la matriz, y la referencia sobre la que se tomarán las decisiones correspondientes en el trabajo.

Confusion Matrix

Output Class	1	484 9.7%	1 0.0%	0 0.0%	1 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.4% 0.6%	
	2	6 0.1%	493 9.9%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	4 0.1%	1 0.0%	0 0.0%	3 0.1%	97.0% 3.0%	
	3	1 0.0%	6 0.1%	491 9.8%	0 0.0%	5 0.1%	2 0.0%	0 0.0%	0 0.0%	2 0.0%	2 0.0%	96.5% 3.5%	
	4	0 0.0%	0 0.0%	1 0.0%	496 9.9%	0 0.0%	0 0.0%	0 0.0%	4 0.1%	1 0.0%	0 0.0%	98.8% 1.2%	
	5	3 0.1%	0 0.0%	0 0.0%	0 0.0%	491 9.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.4% 0.6%	
	6	1 0.0%	0 0.0%	0 0.0%	3 0.1%	1 0.0%	495 9.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.0%	98.6% 1.4%
	7	3 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	492 9.8%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	99.2% 0.8%
	8	0 0.0%	0 0.0%	5 0.1%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	492 9.8%	0 0.0%	0 0.0%	1 0.0%	98.4% 1.6%
	9	1 0.0%	0 0.0%	2 0.0%	0 0.0%	1 0.0%	0 0.0%	4 0.1%	3 0.1%	497 9.9%	1 0.0%	97.6% 2.4%	
	10	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	490 9.8%	99.4% 0.6%	
			96.8% 3.2%	98.6% 1.4%	98.2% 1.8%	99.2% 0.8%	98.2% 1.8%	99.0% 1.0%	98.4% 1.6%	98.4% 1.6%	99.4% 0.6%	98.0% 2.0%	98.4% 1.6%
		1	2	3	4	5	6	7	8	9	10		
		Target Class											

Figura 17. Ejemplo de matriz de confusión con tasa de clasificación=98.4 %.[4]



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**



CAPÍTULO III: COLECTOR DE AGUAS



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES

3.1 CASO DE ESTUDIO

La planta de la que han extraído los datos necesarios para la realización de este trabajo consiste en una red de recogida de agua y alcantarillado simulada [21]. Este colector de aguas, desarrollado en [22], tiene objetivo de servir como banco de pruebas dentro de la comunidad científica, especialmente en ingeniería de control. La particularidad de este modelo integrado se encuentra en que representa la interconexión que existe en la actualidad entre las plantas de tratamiento de aguas residuales (EDAR) con otros elementos como son las redes de alcantarillado o los equipos de recogida de agua fluvial. Además, la simulación cuenta con algoritmos que generan las características propias de distintas situaciones meteorológicas, generando así un volumen de precipitaciones cuya intensidad y duración dependen del día de la semana o la estación del año. Como consecuencia de esto el modelo está dotado de una gran complejidad, y su desarrollo ha sido posible gracias al exponencial aumento de la potencia computacional en las últimas décadas.

3.2 DESCRIPCIÓN DE LA PLANTA

La planta (Figura 18) está formada por seis colectores de agua pluvial y residual (1, 2, 3, 4, 5, 6) y seis tanques de almacenamiento de aguas residuales (ST1, ST2, ST3, ST4, ST5, ST6), de los cuales uno de ellos se encuentra desconectado (ST5). El resto de la planta lo completan las tuberías que interconectan todos los elementos, cinco válvulas que regulan el caudal de salida de cada depósito conectado, una bomba hidráulica y la planta de tratamiento de aguas residuales (WWTP en la Figura 18, por sus siglas en inglés). El objetivo del alcantarillado es recoger todas las aguas residuales y conducir las a la EDAR, a la vez que se mantiene un caudal con un valor cercano al nominal, y una variabilidad baja. Los tanques de los que dispone la planta tienen la función de almacenar aguas residuales, de tal manera que sea posible aumentar el caudal cuando este sea bajo, y cortar el suministro cuando el valor del caudal aumente.

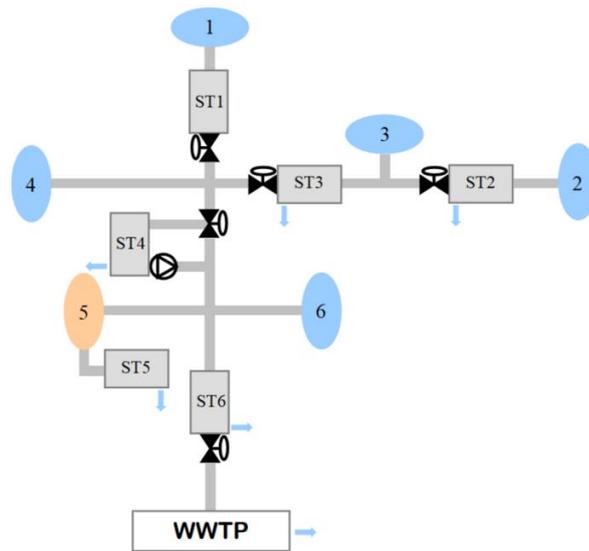


Figura 18. Esquema de la planta de alcantarillado.

En la Figura 19 se muestra el diagrama de bloques del sistema de alcantarillado, en el que el depósito de almacenamiento 6 ha sido renombrado a 5. Los triángulos azul marino se corresponden a los depósitos de almacenamiento, las elipses celestes a los colectores de agua, y los rectángulos amarillos representan las tuberías.

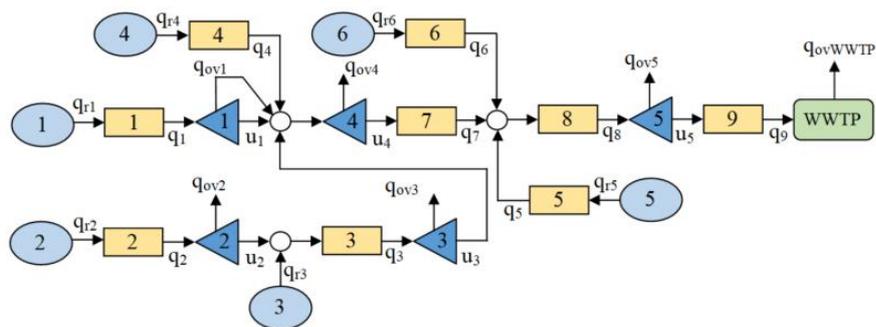


Figura 19 Diagrama de bloques simplificado de la planta de alcantarillado

Para la detección y diagnóstico de los fallos de la planta es necesario obtener información del proceso a lo largo de un periodo de tiempo. Esta información es extraída gracias a una serie de sensores repartidos en diferentes zonas. La planta posee sensores de nivel de depósito en cada uno de los tanques,



sensores de caudal en todas las tuberías que componen la instalación, y sensores de apertura en las válvulas alojadas en cada uno de los depósitos de almacenamiento.

Para el desarrollo del trabajo, se hace uso de un modelo simulado de la planta en SIMULINK [22], de donde se obtienen los conjuntos de datos necesarios para el estudio. Los datos se obtienen a partir de sensores distribuidos por la planta.

3.3 VARIABLES Y DATOS DE LA PLANTA

En la Figura 20 están representadas todas las variables que se pueden recoger en la planta. De estas variables q_1 , q_2 , q_4 , q_5 y q_6 son perturbaciones que entran al sistema, por lo que no se pueden modificar. Por otro lado, q_3 , q_7 , q_8 , q_9 , y los niveles de los cinco depósitos son las variables de estado del sistema, que se consideran como salidas del proceso.

El objetivo de control del sistema es que el flujo a la entrada de la depuradora (q_9) esté lo más cercano posible a su valor nominal independientemente de las condiciones del sistema al mismo tiempo que se evitan desbordamientos en los tanques y en la propia depuradora. Para lograr este objetivo, se considera como variables manipuladas el caudal de flujo a la salida de los tanques (lo que se ha llamado u_1 , u_2 , u_3 , u_4 y u_5), y el controlador calculará estas variables para minimizar la diferencia entre el caudal que entra a la depuradora y su valor nominal, para lo cual además distribuirá el caudal de agua entre los diferentes tanques de la red maximizando la diferencia entre el nivel de agua en cada tanque y el valor de referencia de dicho nivel calculado de forma dinámica.

VARIABLE	DESCRIPCIÓN	UNIDADES
q1	CAUDAL EN LA TUBERÍA 1	m ³ /d
q2	CAUDAL EN LA TUBERÍA 2	m ³ /d
q3	CAUDAL EN LA TUBERÍA 3	m ³ /d
q4	CAUDAL EN LA TUBERÍA 4	m ³ /d
q5	CAUDAL EN LA TUBERÍA 5	m ³ /d
q6	CAUDAL EN LA TUBERÍA 6	m ³ /d
q7	CAUDAL EN LA TUBERÍA 7	m ³ /d
q8	CAUDAL EN LA TUBERÍA 8	m ³ /d
qwvtp	CAUDAL EN LA TUBERÍA 9 ó CAUDAL EN LA DEPURADORA	m ³ /d
n1	NIVEL DEL AGUA EN EL DEPÓSITO 1	m
n2	NIVEL DEL AGUA EN EL DEPÓSITO 2	m
n3	NIVEL DEL AGUA EN EL DEPÓSITO 3	m
n4	NIVEL DEL AGUA EN EL DEPÓSITO 4	m
n5	NIVEL DEL AGUA EN EL DEPÓSITO 5	m
u1	APERTURA DE LA VÁLVULA 1	%
u2	APERTURA DE LA VÁLVULA 2	%
u3	APERTURA DE LA VÁLVULA 3	%
u4	APERTURA DE LA VÁLVULA 4	%
u5	APERTURA DE LA VÁLVULA 5	%
ref1	REFERENCIA DEL CONTROLADOR PARA EL NIVEL 1	m
ref2	REFERENCIA DEL CONTROLADOR PARA EL NIVEL 2	m
ref3	REFERENCIA DEL CONTROLADOR PARA EL NIVEL 3	m
ref4	REFERENCIA DEL CONTROLADOR PARA EL NIVEL 4	m
ref5	REFERENCIA DEL CONTROLADOR PARA EL NIVEL 5	m

Figura 20. Variables del proceso.

Los sensores recogen información de la planta cada 3 minutos y durante 10 días de funcionamiento de la planta. Sabiendo esto:

$$n = \frac{60 \text{ min}}{3} * \frac{24 \text{ h}}{1 \text{ día}} * 10 \text{ días} = 4800 + 1 = 4801$$

donde n es el número de observaciones en la planta. Se suma una unidad debido a que el sensor envía información en el instante inicial. De esta manera, la matriz de datos del proceso $X \in \mathfrak{R}^{n \times m}$, donde $m=24$ y $n=4801$.

Para estudiar la detección y clasificación de fallos es necesario provocarlos. Para ello, a la planta simulada hay que añadirle un fichero donde se especificará qué fallos se van a provocar y en qué instante de tiempo. En este fichero (Figura 21) se pueden provocar 19 fallos distintos de tipo *offset*, es decir, el fallo que se elija consistirá en añadir un aumento en el valor de una variable. También se debe elegir el instante de tiempo en el que se produce la

anomalía. Tanto el valor del offset del fallo que se elige provocar como el instante de tiempo en el que se provoca, se introducen en el fichero de la Figura 21. Después se simula la planta desarrollada en SIMULINK [21], y tras el periodo de simulación se almacenan los datos de las 24 variables en la matriz X, que es la matriz a partir de la cual se realiza el resto de los procedimientos para el estudio.

```
time_step=(24*60/3)*Ts; %%tiempo en que aparece el fallo => en el primer día

off1=0;      %%tamaño de fallo para el nivel del dep. 1 como offset
off2=2.4;    %%tamaño de fallo para el nivel del dep. 2 como offset
off3=0;      %%tamaño de fallo para el nivel del dep. 3 como offset
off4=0;      %%tamaño de fallo para el nivel del dep. 4 como offset
off5=0;      %%tamaño de fallo para el nivel del dep. 5 como offset
off6=0;      %%tamaño de fallo válvula dep.1 como offset
off7=0;      %%tamaño de fallo válvula dep.2 como offset
off8=0;      %%tamaño de fallo válvula dep.3 como offset
off9=0;      %%tamaño de fallo válvula dep.4 como offset
off10=0;     %%tamaño de fallo válvula dep.5 como offset
off11=0;     %%tamaño del fallo para ref. caudal dep. 1 como offset
off12=0;     %%tamaño del fallo para ref. caudal dep. 2 como offset
off13=0;     %%tamaño del fallo para ref. caudal dep. 3 como offset
off14=0;     %%tamaño del fallo para ref. caudal dep. 4 como offset
off15=0;     %%tamaño del fallo para ref. caudal dep. 5 como offset
off16=0;     %%tamaño del fallo para el caudal 3 como offset
off17=0;     %%tamaño del fallo para el caudal 7 como offset
off18=0;     %%tamaño del fallo para el caudal 8 entre como offset
off19=0;     %%tamaño del fallo para el caudal 9 (WWTP) como offset
```

Figura 21. Código del archivo para provocar fallos.



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES



CAPÍTULO IV: APLICACIÓN DESARROLLADA



4.1. PROCEDIMIENTO. DATOS DE FALLO.

En este capítulo se van a aplicar las técnicas que se han detallado en el Capítulo II a datos extraídos del colector de aguas, de forma que se estudiará la detección de fallos producidos en la planta y su posterior diagnóstico.

Para este trabajo se han provocado un total de 10 fallos independientes, lo que dará lugar a 10 matrices de datos X. Todos los fallos se han efectuado para la observación $n=480$, lo que significa que el fallo se provoca al comienzo del segundo día de simulación, con la intención de que, si se produjesen oscilaciones al comienzo de la simulación, le diera tiempo a la planta a estabilizarse. La Figura 22 muestra una tabla con un resumen de todos los fallos provocados en la planta. También se ha obtenido una matriz X de comportamiento normal, es decir, sin ningún fallo provocado. Esta matriz será necesaria para aplicar la técnica de Análisis de Componentes Principales.

VARIABLE	NOMBRE FALLO	TIPO	VALOR
qwwtp	SENSOR CAUDAL DEPURADORA	OFFSET	20000
			23000
q7	SENSOR CAUDAL Q7	OFFSET	2000
			3500
q8	SENSOR CAUDAL Q8	OFFSET	6000
			8500
n4	SENSOR NIVEL 4° DEPÓSITO	OFFSET	2
			2,4
n5	SENSOR NIVEL 5° DEPÓSITO	OFFSET	2
			2,4

Figura 22. Tabla de fallos provocados en la planta.

Se han provocado 10 fallos, pero solo sobre 5 variables. Cada una de ellas tiene asociados 2 fallos de distinta magnitud o valor. Se ha hecho de esta manera para que estos mismos fallos se puedan utilizar tanto en el estudio de detección como en el de clasificación, y se tenga un conjunto de datos de entrenamiento de los métodos y un conjunto diferente para validar dichos métodos.

4.2. MODELOS PARA LA DETECCIÓN DE FALLOS

4.2.1. ANÁLISIS DE COMPONENTES PRINCIPALES (PCA)

Para aplicar esta técnica multivariante, primero se trabaja fuera de línea, donde se realiza la reducción de dimensión del espacio y se adquieren los componentes principales a partir de la matriz X de comportamiento normal. También se calcularán los dos estadísticos (T^2 y Q) mencionados en el capítulo II y sus umbrales. Después de esto se efectuará un trabajo en línea donde se validará el método con los datos de fallo de la Figura 22.

4.2.1.1. FUERA DE LÍNEA

Los datos que se utilizan para la obtención de los componentes principales son los datos de comportamiento normal de la planta (matriz X de dimensiones 4801×24). El objetivo es reducir el espacio $\mathfrak{R}^{24 \times 4801}$ a otro con dimensión $\mathfrak{R}^{a \times 4801}$, siendo a el número de componentes principales. Se normalizan los datos a media 0 y varianza 1: con X normalizado, se calcula la matriz de correlaciones R (Ecuación 4), y se guardan en Λ los valores singulares en orden decreciente, extraídos de la matriz R . También se extrae de R la matriz de vectores propios V , la matriz de loadings (Ecuación 5). Se elige una variabilidad del 95%, lo que dará lugar a la selección de aquellos vectores de carga que se correspondan con los valores singulares cuya variabilidad sea mayor o igual a la elegida, y se almacenarán en la matriz $P \in \mathfrak{R}^{a \times 24}$. Con el porcentaje de varianza del 95%, el número de componentes principales es 8 ($a=8$).

También fuera de línea se calculan los dos umbrales estadísticos que vamos a utilizar para el estudio (T_α y Q_α) con las ecuaciones (13) y (16):

$$T_\alpha = 20.1534$$

$$Q_\alpha = 6.3293$$

4.2.1.2. PCA ONLINE

Una vez obtenidos los componentes principales y los umbrales de los estadísticos, se procede a la parte de la detección. Como se ha explicado anteriormente, los datos recogidos en la planta durante 10 días forman una matriz $X \in \mathfrak{R}^{4801 \times 24}$, que serán datos normales si no se produce ninguna anomalía en el proceso, o podrán ser datos de fallos si se provoca alguno en la simulación. Estos datos de funcionamiento serán normalizados con la media y la varianza de la normalización realizada fuera de línea con los datos de funcionamiento normal. Una vez se normaliza X , se calculan los estadísticos T^2 y Q . Para dicho cálculo se utilizan las ecuaciones (11) y (14), en las que la diferencia con respecto a fuera de línea es que se introduce la matriz X que se acaba de normalizar.

Una vez terminados los últimos cálculos, se procede a graficar las soluciones e interpretar los resultados.

4.2.1.3. RESULTADOS DEL PCA CON UMBRAL CONSTANTE

Para el estudio y análisis de la detección de fallos, se ha decidido calcular 3 parámetros para cada simulación, con el objetivo de reflejar con mayor claridad la eficacia de la detección de anomalías:

1. Tiempo de detección de fallo (t_fallo): Indica el instante de tiempo, (desde $n=0$ hasta $n=4801$) en el que se detecta el fallo. La referencia que se deberá tener en cuenta es $t=480$, pues todos los fallos se van a provocar en ese instante, el cual equivale al primer día de simulación de la planta.
2. Porcentaje de alarmas: Indica el porcentaje de los datos que se encuentran por encima del umbral del estadístico. Solo tiene en cuenta en el cálculo el intervalo de valores comprendido entre el instante de fallo detectado y el final de la simulación ($t_fallo;4801$).

3. Porcentaje de falsas alarmas: Indica el porcentaje de los datos que se encuentran por encima del umbral antes de que se detecte el fallo. Solo tiene en cuenta en el cálculo el intervalo de valores comprendido entre el instante inicial y el instante de fallo detectado ($n=1; t_{\text{fallo}}$).

Cabe destacar que, debido a las posibles irregularidades del proceso en cualquier estado de funcionamiento, se considerará que se ha detectado un fallo en la planta cuando el umbral sea sobrepasado por el estadístico durante 15 instantes de tiempo seguidos, que equivale a 45 minutos de funcionamiento descontrolado por parte de la planta. Por otro lado, puesto que se ha comprobado que la simulación del proceso tarda un tiempo en estabilizarse, no se han tenido en cuenta los 10 primeros instantes de tiempo para la detección, lo que equivale a 30 minutos de funcionamiento.

A continuación se muestran varios resultados gráficos de la detección de diferentes fallos que se han provocado, junto con el valor de los parámetros explicados anteriormente. La Figura 23 muestra la detección del fallo en el nivel del 5º depósito (n_5), con un $\text{offset}=2$.

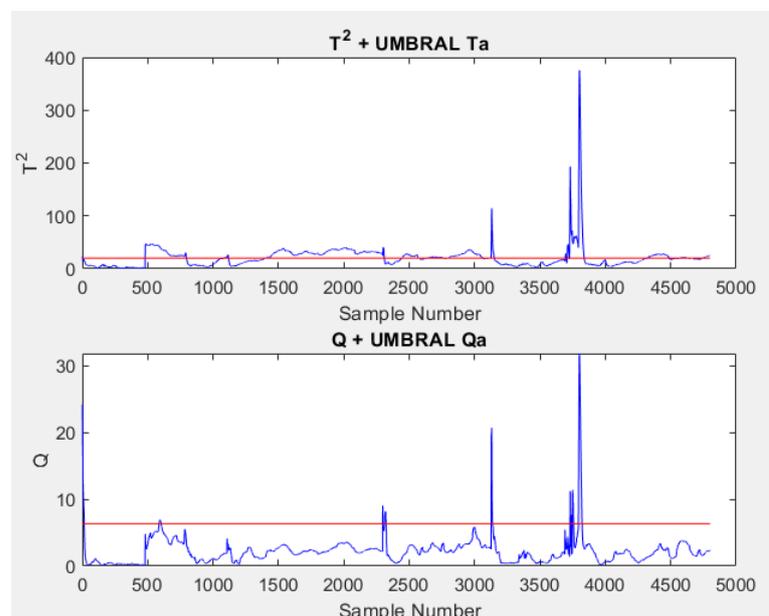


Figura 23. (a) **Arriba**. Gráfico de control T^2 para el fallo en n_5 ($\text{offset}=2$). (b) **Abajo**. Gráfico de control Q para el fallo en n_5 ($\text{offset}=2$).

Se aprecia que en antes de la observación 500 ya se acercan al umbral ambos estadísticos. Los parámetros que se enunciaron anteriormente se muestran en la Figura 24.

TIEMPO DETECCIÓN FALLO (n)		PORCENTAJE ALARMAS (%)		FALSAS ALARMAS (%)	
T2	Q	T2	Q	T2	Q
481	587	46,4126	1,6907	0,2219	1,0381

Figura 24. Resultado paramétrico del fallo en n5 (offset=2).

El tiempo de detección de fallo es bueno en ambos estadísticos, así como los porcentajes de falsas alarmas. El porcentaje de alarmas no supera el 50% en el caso de T^2 , y es extremadamente bajo en Q (1.69 %). Un valor tan bajo puede indicar que el fallo no provoca desorden en el proceso productivo a lo largo del tiempo, sino que únicamente lo altera durante un pequeño intervalo de tiempo. Sin embargo, la gráfica de la Figura 23 (b) cambia claramente la tendencia durante todas las observaciones desde el momento en el que se provoca el fallo. Esto podría indicar que el umbral no se adecúa correctamente al funcionamiento de la planta.

Si se observa el resultado del fallo en el nivel del 4º depósito (n4 con offset=2) en la Figura 24, se puede identificar un comportamiento similar de los estadísticos con respecto al fallo anterior.

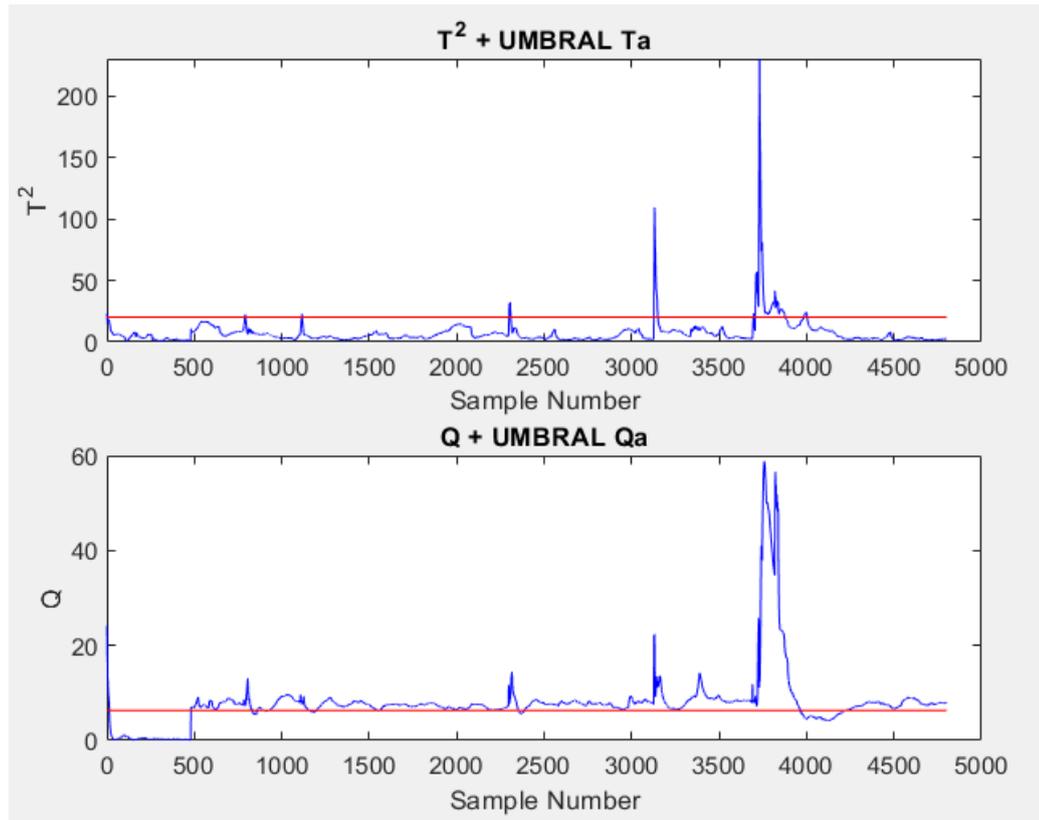


Figura 25. (a) Arriba. Gráfico de control T^2 para el fallo en n4 (offset=2,4). (b) Abajo. Gráfico de control Q para el fallo en n4 (offset=2,4).

TIEMPO DETECCIÓN FALLO (n)		PORCENTAJE ALARMAS (%)		FALSAS ALARMAS (%)	
T2	Q	T2	Q	T2	Q
3128	481	4,4879	81,4444	0,5771	1,2771

Figura 26. Resultado paramétrico del fallo en n4 (offset=2).

Como reflejan los datos aportados por la Figura 26, T^2 no logra detectar el fallo hasta la observación 3128, donde se aprecia un pico de gran valor. Por otra parte, Q sí que detecta el fallo con suma rapidez, pues lo hace en la observación 481. Además, el porcentaje de alarmas en Q es alto, lo cual es otro indicador positivo con respecto a la efectividad de Q en la detección.

En la tabla de la Figura 27 se muestra un resumen de los 10 fallos provocados en la planta. Tan solo el fallo en el nivel del 5º depósito consigue ser detectado con rapidez, lo cual indica que la efectividad del PCA para la detección no está

siendo bueno. Esta falta de efectividad puede deberse al diseño de la planta, ya que como se mencionó en el Capítulo III, la simulación del proceso lleva implementados algoritmos que tienen en cuenta la variabilidad meteorológica. La idea de esta hipótesis surge de que, en comportamiento normal, las gráficas de los estadísticos fuera de línea (Figura 28) poseen picos de funcionamiento en los mismos intervalos de observaciones que donde se están detectando los fallos. Los picos surgen sobre la observación 3200-3300, y sobre la observación 3500-3400. Estos picos pueden representar periodos de tiempo en los que las lluvias han sido más intensas, mientras que en los primeros días de simulación es probable que las lluvias hayan sido más escasas, lo que hace más complicado la detección de fallos. Un aumento del nivel de un depósito, o el aumento volumétrico de un determinado caudal será más fácil de controlar por la planta cuando las lluvias sean menores que cuando haya inundaciones. Es por esto por lo que es posible que los fallos no se detecten hasta llegada a la etapa de máximas lluvias, la cual, viendo los datos de la Figura 28, hace suponer que se encuentran sobre el séptimo y el octavo día.

ESTUDIO DETECCIÓN FALLOS								
NOMBRE FALLO	TIPO	VALOR	TIEMPO DETECCIÓN FALLO		PORCENTAJE ALARMAS		FALSAS ALARMAS	
			T2	Q	T2	Q	T2	Q
SENSOR NIVEL 5º DEPÓSITO	OFFSET	2	481	587	46,7	1,6907	0,2119	1,0381
	OFFSET	2,4	481	583	54	7,8272	0,2119	1,0453
SENSOR NIVEL 4º DEPÓSITO	OFFSET	2	827	585	7,33	10,645	0,1222	1,7361
	OFFSET	2,4	3128	481	4,49	81,4444	0,5771	1,2712
SENSOR CAUDAL Q7	OFFSET	2000	3128	3843	3,03	0,7514	0,3206	1,0433
	OFFSET	3500	3128	3791	3,09	1,1271	0,3206	0,661
SENSOR CAUDAL Q8	OFFSET	6000	3128	3786	3,82	1,315	0,7695	0,503
	OFFSET	8500	1595	3782	8,6	1,5654	0,3153	0,4241
SENSOR CAUDAL DEPURADORA	OFFSET	20000	3128	3179	2,98	5,9695	0,3206	0,7571
	OFFSET	23000	3128	1084	11,5	4,8633	0,4168	0,5581

Figura 27. Resumen de los 10 fallos provocados en la planta con umbral constante.

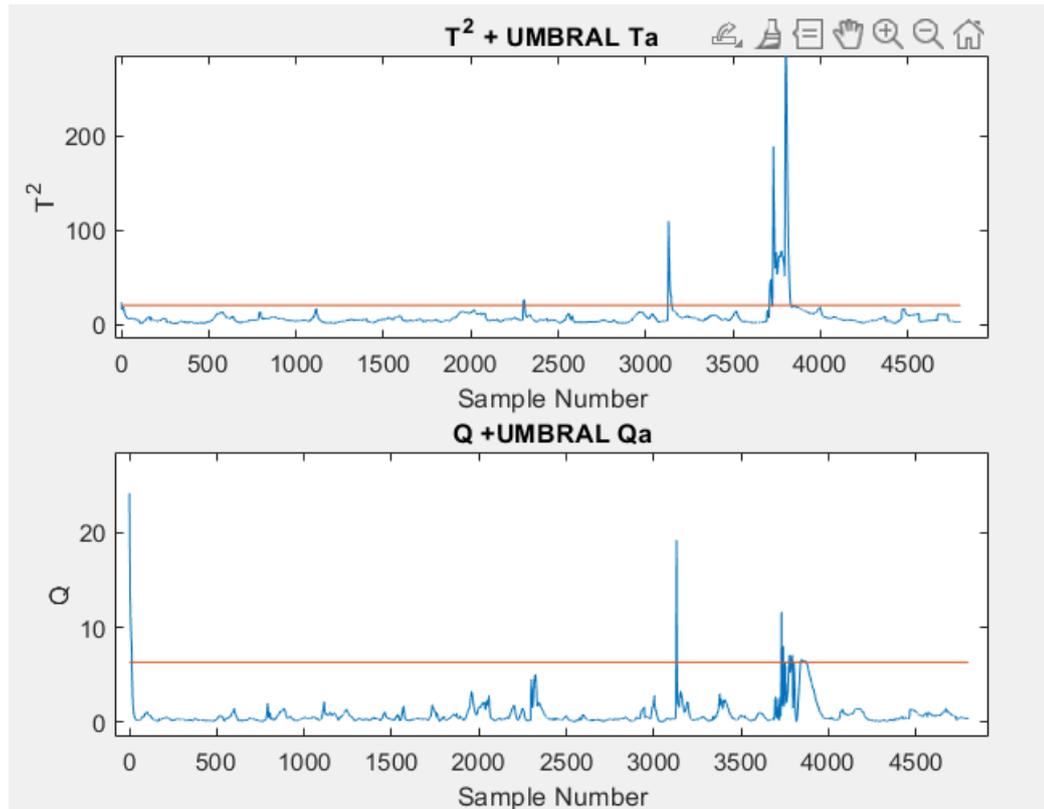


Figura 28. (a) Arriba. Gráfico de control T^2 para comportamiento normal. (b) Abajo. Gráfico de control Q para comportamiento normal.

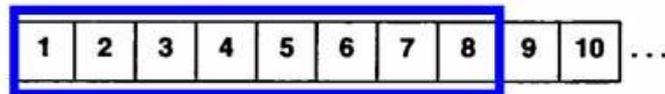
Para intentar solucionar este problema, en el apartado siguiente se van a modificar los umbrales de ambos estadísticos, introduciendo el concepto de umbral adaptativo.

4.2.1.4. RESULTADOS DEL PCA CON UMBRAL ADAPTATIVO

En este apartado se va a introducir una modificación el proceso de detección de los fallos, con el objetivo de mejorar los resultados obtenidos en el apartado anterior, ya que distan bastante de ser precisos y concluyentes. Se introduce el concepto de Umbral Adaptativo, que consiste en ajustar el umbral a los valores del estadístico en cada momento, obteniendo una curva variable en el tiempo en vez de una recta paralela al eje de abscisas.

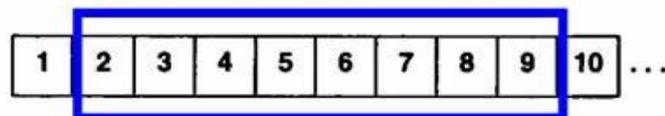
Para formar el umbral adaptativo de T^2 se ha utilizado el mismo procedimiento que para calcular el umbral adaptativo de Q (Ecuación 16), pero aplicando el método de ventana deslizante al vector T^2 (1×4801). Este método consiste en ir calculando valores del umbral, pero para intervalos de observaciones menores. Se ha decidido que serán 10 las observaciones que formen cada intervalo, y se desplazará una observación al cambiar al siguiente intervalo. La Figura 29 muestra el procedimiento. De esta manera se obtiene un vector del umbral adaptativo de T^2 de dimensiones 1×4792 . Los 9 valores que faltan para completar el vector son los 9 primeros porque ambas ventanas deslizantes se han realizado a pasado. Estos 9 valores serán igual al valor del umbral del apartado anterior. El cálculo del umbral adaptativo de Q es igual que el de T^2 , pero aplicando la ventana deslizante al vector Q (1×4801).

ESTADO INICIAL DE LA VENTANA DESLIZANTE



(a)

ESTADO DE LA VENTANA UNA VEZ DESLIZADA



(b)

Figura 29. Esquema de funcionamiento de la ventana deslizante aplicada a un vector.

Los resultados obtenidos de provocar el fallo en *nivel del 5º depósito con offset=2,4* indican un buen funcionamiento del PCA tras la mejora.

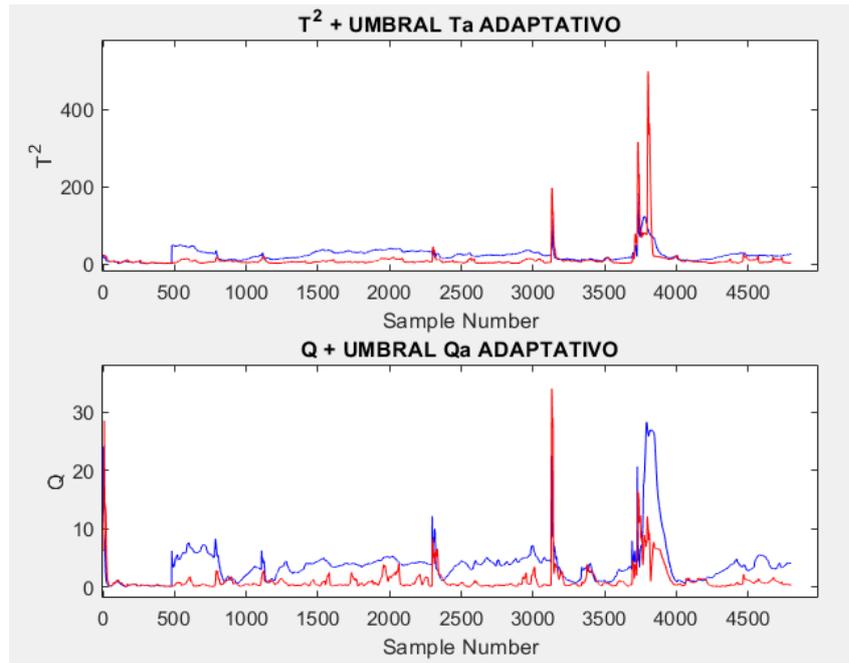


Figura 30. (a) Arriba. Gráfico de control T^2 para el fallo en n5 (offset=2,4). (b) Abajo. Gráfico de control Q para el fallo en n5 (offset=2,4).

La curva roja de las figuras Xa y Xb, son los umbrales adaptativos de ambos estadísticos. Al igual que en el apartado anterior, en aquellos lugares en donde el umbral esté por encima del estadístico durante n=15 observaciones consecutivas, el fallo será detectado por el PCA.

TIEMPO DETECCIÓN FALLO (n)		PORCENTAJE ALARMAS (%)		FALSAS ALARMAS (%)	
T2	Q	T2	Q	T2	Q
481	481	85,8067	84,3039	1,0593	0,6356

Figura 31. Resultados de PCA adaptativo con fallo n5 (offset=2,4).

La detección de los fallos con PCA y umbral adaptativo aporta mejores resultados que en el apartado anterior, aumentando la efectividad del método de forma considerable. Así lo refleja la tabla de la Figura 32, en la que se encuentra el resumen de resultados tras provocar los 10 fallos distintos.

			ESTUDIO DETECCIÓN FALLOS					
			TIEMPO DETECCIÓN FALLO		PORCENTAJE ALARMAS		FALSAS ALARMAS	
NOMBRE FALLO	TIPO	VALOR	T2	Q	T2	Q	T2	Q
NIVEL 5º DEPÓSITO	OFFSET	2	481	481	70,048	74,5773	1,0593	0,6356
	OFFSET	2,4	481	481	85,8067	84,3039	1,0593	0,6356
NIVEL 4º DEPÓSITO	OFFSET	2	481	481	44,5627	83,5107	1,0593	0,6356
	OFFSET	2,4	481	481	38,781	89,7934	1,0593	0,6356
CAUDAL Q7	OFFSET	2000	846	481	25,3392	49,6765	2,3895	0,6356
	OFFSET	3500	682	481	37,2991	68,1277	1,3373	0,6356
CAUDAL Q8	OFFSET	6000	651	481	38,0714	63,1183	0,7788	0,6356
	OFFSET	8500	641	481	46,0447	69,6097	0,7911	0,6356
CAUDAL DEPURADORA	OFFSET	20000	678	653	32,9785	55,0824	1,0463	2,3292
	OFFSET	23000	481	481	57,942	84,45	1,0593	0,6356

Figura 32. Resumen de los 10 fallos provocados en la planta con umbral adaptativo.

De la figura anterior se puede destacar que la mayoría de los datos se detectan de forma prácticamente instantánea desde su aparición, y no hay ninguno que tenga un tiempo de detección superior a las 653 observaciones, lo que equivaldría a un tiempo de detección máximo de 8,63 horas. Si se comparan los dos estadísticos, se llega a la conclusión de que el funcionamiento de Q es mejor que el que desarrolla T^2 , tanto en tiempo de detección como en porcentaje de alarmas, ya que los fallos detectados por Q se mantienen en el tiempo más que los de T^2 .

4.2.2. PCA NO LINEAL

Con el propósito y el objetivo de continuar mejorando el modelo obtenido anteriormente, se ha desarrollado un modelo de PCA no lineal. En este modelo se utilizan las siguientes herramientas: Redes Neuronales y Autoencoders. También se compararán los resultados obtenidos con el umbral constante y con el umbral adaptativo.

4.2.2.1. PCA NO LINEAL OFFLINE

La matriz X de datos de funcionamiento normal se normaliza con media 0 y varianza 1, al igual que en PCA lineal. La diferencia es que ahora se procede a crear y entrenar una red neuronal para la detección de fallos. La creación de la red se hará por capas, donde cada capa es un Autoencoder con un número de neuronas determinado en la capa oculta. Estos Autoencoders se irán entrenando por separado, y una vez estén entrenados, se apilarán todos ellos (Stacked Autoencoder) formando una única red neuronal, donde el número de entradas sea igual al número de salidas.

La arquitectura de la red es simétrica, formada por las capas de entrada y salida, de 24 neuronas cada una, y 3 capas ocultas: 2 capas ocultas de 30 neuronas, y la capa oculta central compuesta por 10 neuronas. De esta manera se consigue una arquitectura *bottleneck*, que favorece el funcionamiento de las redes neuronales [21].

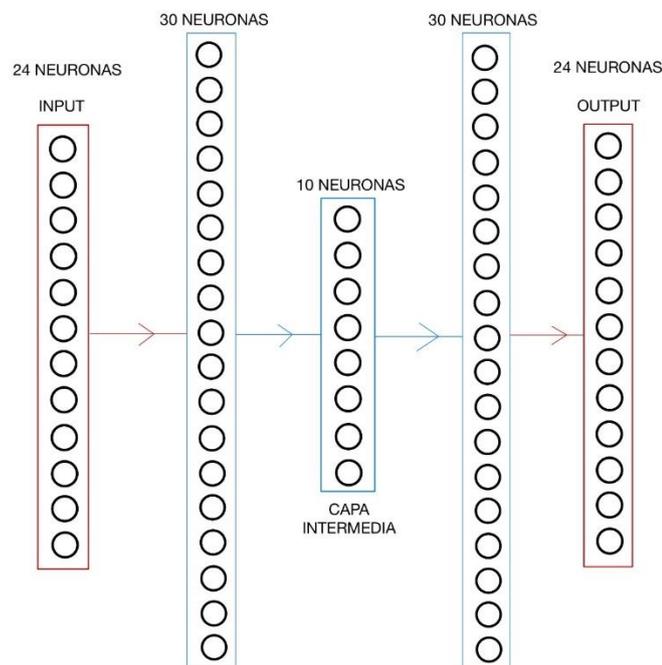


Figura 33. Arquitectura de la red final obtenida.

El primer Autoencoder, de 30 neuronas en la capa oculta, se entrena con la matriz X normalizada, por lo que dispondrá de 24 neuronas en las capas de



entrada y salida. Para conseguir la arquitectura deseada, se codifica el primer Autoencoder para obtener la capa oculta (30 neuronas), que será la matriz de datos de entrenamiento para el segundo Autoencoder, de 10 neuronas en la capa oculta. Para el tercer Autoencoder el proceso es el mismo: se codifica el segundo Autoencoder y se extrae la capa oculta (10 neuronas), que será de nuevo de la matriz de datos de entrenamiento del tercer Autoencoder, creado con 30 neuronas en la capa oculta para completar la simetría. Llegado a este punto, se tiene el tercer Autoencoder con 30 neuronas en la capa oculta, pero no se puede crear un cuarto Autoencoder, ya que este tipo de red neuronal, como se ha explicado en el capítulo II, aproxima la salida a la entrada tras un proceso de codificación y decodificación. Por lo que, para la última capa, no se podrá proceder con un cuarto Autoencoder, sino con una red neuronal con una capa oculta, cuyas únicas diferencias son que para el entrenamiento se introducen tanto los datos de entrada (capa oculta de 30 neuronas extraída del tercer Autoencoder), como los *datos objetivo*, que es la salida que se desea que tenga la red (matriz X normalizada); y que el número de salidas es distinto que el de entradas. La capa de salida de esta red tiene 24 neuronas, por lo que se completa la simetría y se procede a apilar la red. Esta red final también deberá ser entrenada, y para dicho entrenamiento tanto los datos de entrada como los datos objetivo serán la matriz X de datos de funcionamiento bajo control normalizada. La Figura 34 muestra los esquemas de los 3 Autoencoders y la red que forman la red final.

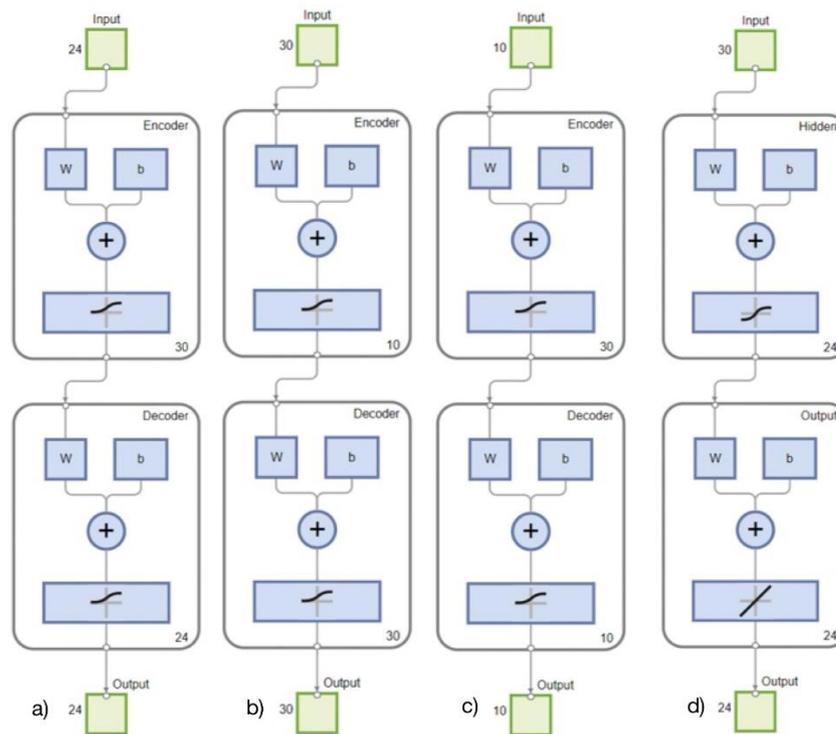


Figura 34. a) Arquitectura Autoencoder 1. b) Arquitectura Autoencoder 2. c) Arquitectura Autoencoder 3. d)Arquitectura Red

En la Figura 35 se muestran los entrenamientos de los 3 Autoencoders y de la red por separado, antes de apilarlos y formar la red final. En estas gráficas se muestra la comparación entre la entrada (curva azul) y la salida real (curva roja) de cada una de las redes por separado.

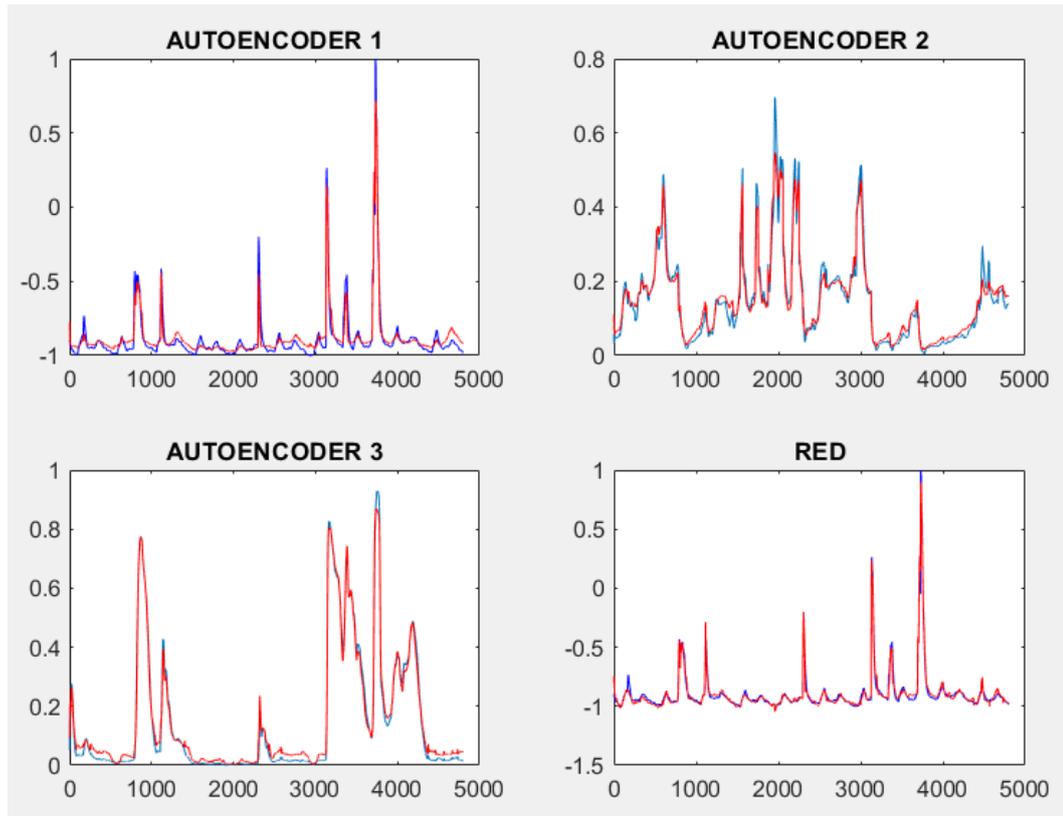


Figura 35. Gráficas de entrenamiento de Autoencoders y red.

Una vez creada la red, el paso siguiente es el cálculo de los estadísticos y los umbrales fuera de línea para comportamiento normal. Para calcular T^2 se necesitará conocer el valor de las salidas de la capa intermedia de 10 neuronas. Habrá que calcular por lo tanto las salidas de la capa de 30 neuronas, y con estas salidas que son entradas a la siguiente capa de la red neuronal se calcularán las que salen de la capa intermedia. La función de transferencia de la red es logarítmica, por lo que las salidas tendrán la siguiente forma:

$$a = \text{logsig}(w * p + b) \quad (20)$$

donde w son los pesos de cada capa, p son las entradas relativas a la capa, y b son los umbrales de activación o bias. logsig es la función no lineal γ de la ecuación (19) en el Capítulo II. Por tanto, para la capa de 30 neuronas la salida toma el valor siguiente:

$$a_{30} = \text{logsig}(w_{30} * X_{tn} + b_{30}) \quad (21)$$

donde a_{30} es una matriz de 30×4801 y X_{tn} es la matriz X de datos de funcionamiento normal normalizada. Y para la capa intermedia, la ponderación es:

$$a_{10} = \text{logsig}(w_{10} * a_{30} + b_{30}) \quad (22)$$

donde a_{10} es una matriz 10×4801 . Para el cálculo de T^2 se utiliza la ecuación genérica (12), donde $T = a_{10}$, que se corresponde con la salida de la capa del cuello de botella de la red. El cálculo de Λ^{-1} es distinto que en PCA lineal puesto que no se ha realizado el cálculo de los valores propios. Por lo tanto:

$$\Lambda^{-1} = \left(\frac{T^t * T}{n-1} \right)^{-1} \quad (23)$$

El cálculo de Q se realiza empleando la ecuación (14), pero el residuo r en este caso se calcula como la diferencia entre la salida de la red entrenada con los datos de entrenamiento, y los propios datos de entrenamiento que es la salida deseada:

$$r = X_{red} - X_n \quad (24)$$

Tras el cálculo de los dos estadísticos fuera de línea, se procede a calcular los estadísticos T_α y Q_α con sus respectivas ecuaciones (13) y (16). Estos umbrales, al igual que en PCA lineal, serán los utilizados en la etapa en línea.

4.2.2.2. PCA NO LINEAL ONLINE

Con la red creada a partir de los datos de funcionamiento normal, calculados los estadísticos T^2 y Q , y obtenidos los umbrales de ambos estadísticos, se procede a la evaluación y el estudio de la detección de fallos por parte del PCA no lineal. Para ello, se seguirá un procedimiento similar al empleado en el PCA lineal. Se normaliza la matriz con fallo X con la media y la varianza de la normalización realizada fuera de línea con los datos de funcionamiento normal.

Tras obtener la matriz de datos de fallo X normalizada (X_n), se calculan los estadísticos T^2 y Q de la misma manera que fuera de línea, pero sustituyendo la matriz X de datos normales normalizada por la de datos de fallo normalizada.

Una vez terminados los cálculos de los estadísticos y sus umbrales, se procede a graficar las soluciones e interpretar los resultados.

4.2.2.3. RESULTADOS PCA NO LINEAL CON UMBRAL CONSTANTE

Para el estudio de la detección de fallos con el modelo no lineal, se tendrán en cuenta el valor de los 3 parámetros que se usaron con anterioridad: tiempo de detección de fallo, porcentaje de alarmas y porcentaje de falsas alarmas.

			ESTUDIO DETECCIÓN FALLOS					
			TIEMPO DETECCIÓN FALLO (n)		PORCENTAJE ALARMAS (%)		FALSAS ALARMAS (%)	
NOMBRE FALLO	TIPO	VALOR	T2	Q	T2	Q	T2	Q
NIVEL 5º DEPÓSITO	OFFSET	2	3727	3776	8,5581	5,0682	1,1834	1,911
	OFFSET	2,4	3127	481	6,209	2,6383	0,5452	1,6949
NIVEL 4º DEPÓSITO	OFFSET	2	3728	3728	7,7281	8,1006	1,3982	0,8873
	OFFSET	2,4	3127	3127	7,4627	5,791	0,449	0,5752
CAUDAL Q7	OFFSET	2000	3728	3785	8,5661	2,5565	0,6184	1,1123
	OFFSET	3500	3728	3783	8,6592	2,0608	0,5916	1,0069
CAUDAL Q8	OFFSET	6000	3728	3784	8,7523	1,7682	0,5916	0,6887
	OFFSET	8500	3727	3781	8,8372	1,8609	0,6455	0,7423
CAUDAL DEPURADORA	OFFSET	20000	10	3727	2,9215	6,8837	100	0,5648
	OFFSET	23000	3727	3727	9,3953	5,2093	0,6455	0,3765

Figura 36. Tabla de resultados de PCA no lineal con umbral constante.

Tras realizar los cálculos de los 10 fallos distintos sobre el PCA no lineal (resumen en la Figura 36), a la vista está que este modelo de detección de fallos no sólo no consigue su objetivo de mejorar la eficacia en la detección de fallos, sino que empeora el modelo anterior. Tan sólo el fallo en n5 es detectado por el estadístico Q , resaltado en verde en la Figura 36. El resto de los tiempos de fallo distan bastante de ser aceptables en la detección, pues la mayoría se encuentran por encima de la observación $n=3700$. Los porcentajes de alarmas

detectadas son extremadamente bajas, pues ninguno supera el 10%. El porcentaje de falsas alarmas no supera el 2% en ningún fallo, lo que supone un aspecto positivo, pero esto se debe, como se verá a continuación las Figuras 37 y 38, a que el umbral tiene un valor demasiado elevado.

En la Figura 37 a) y b) se han representado ambos estadísticos y sus umbrales para el caso de fallo en el caudal 7 con $\text{offset}=20000$. En dicha gráfica los umbrales tanto de T^2 como de Q apenas son superados por los estadísticos, y ese sobrepaso de los umbrales siempre se produce en las mismas observaciones: aproximadamente en la observación $n=3700$, lo que equivale al final de séptimo día de simulación. El máximo producido entre las observaciones 2000 y 2500 sí que podría ser considerado motivo suficiente para considerar que se produce fallo, ya que es un máximo que se encuentra fuera de las zonas donde siempre se amplifica la señal (posibles zonas de lluvia intensa/inundaciones en comportamiento normal). Sin embargo, debido a la condición de $n=15$ observaciones consecutivas por encima del umbral, no se detecta. Aun así seguiría siendo un tiempo de detección excesivamente largo.

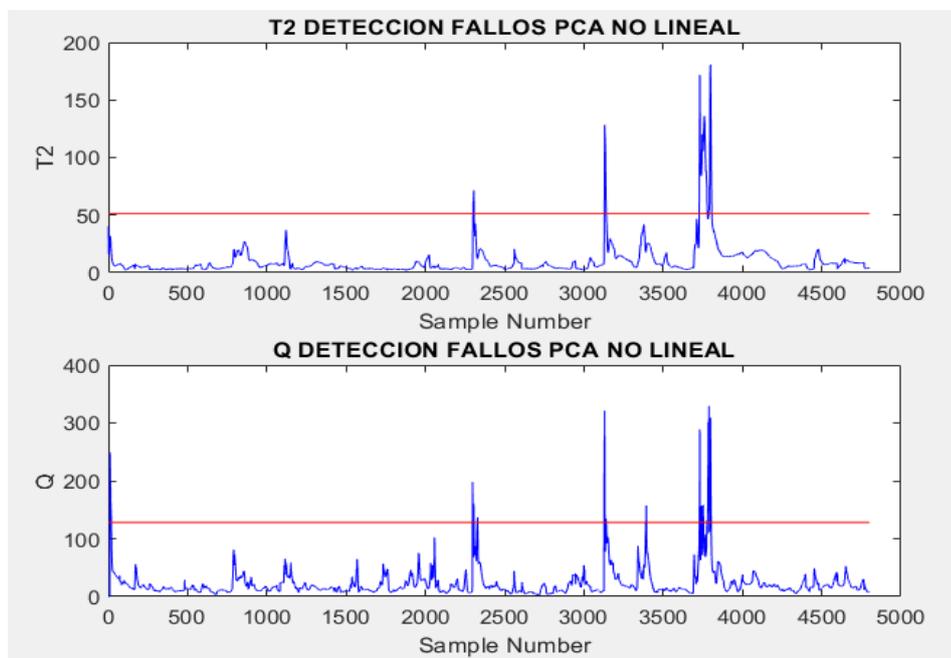


Figura 37. (a) Arriba. Gráfico de control T^2 para el fallo en q7 (offset=2000). (b) Abajo. Gráfico de control Q para el fallo en q7 (offset=2000).

Otro gráfico de control se muestra en la Figura 38 (a) y (b). En este caso se trata del fallo en el nivel del 4º depósito con $\text{offset}=2,4$. En el caso del estadístico T^2 (Figura 38 (a)) se produce un aumento del estadístico antes de que se provoque siquiera el fallo, lo que indica un grave error en el funcionamiento del modelo. Aun así debido al excesivo valor del umbral, en la Figura 38 (a) el umbral solo es sobrepasado en los mismos puntos que en el ejemplo de fallo anterior. Por otro lado, en la Figura 38 (b) el estadístico Q consigue detectar el fallo con facilidad, siendo el único fallo que consigue ser detectado por algún estadístico de los 10 fallos simulados.

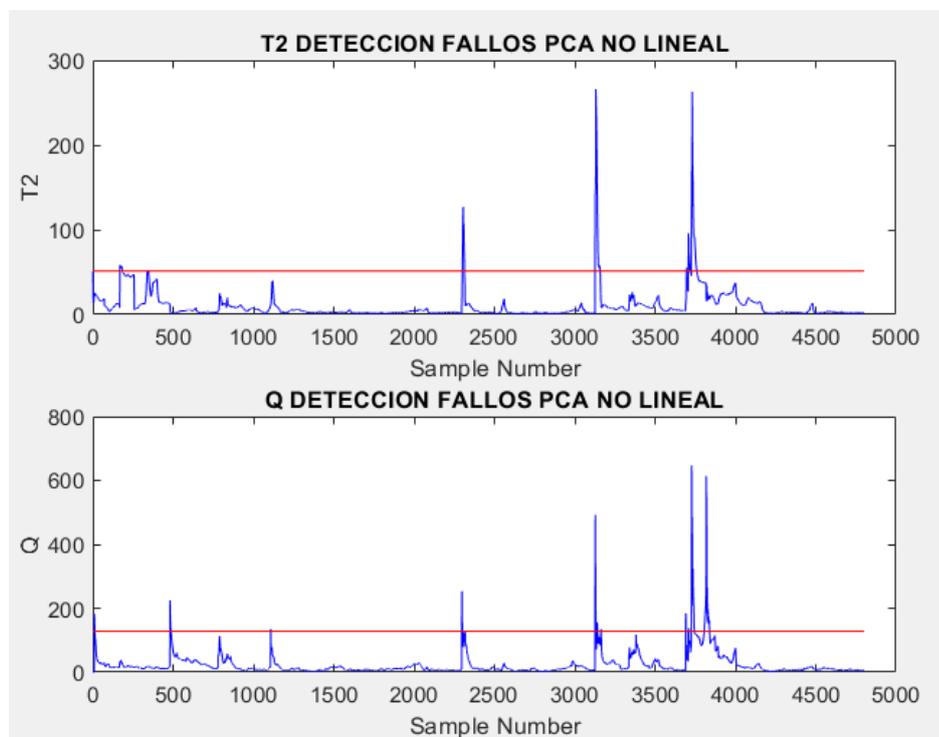


Figura 38. (a) Arriba. Gráfico de control T^2 para el fallo en n_4 ($\text{offset}=2,4$). (b) Abajo. Gráfico de control Q para el fallo en n_4 ($\text{offset}=2,4$).

4.2.2.4. RESULTADOS PCA NO LINEAL CON UMBRAL ADAPTATIVO

La gráfica de la Figura 39 muestra el valor de los parámetros para la detección, pero referentes a un umbral adaptativo creado mediante ventana deslizante en pasado.

			ESTUDIO DETECCIÓN FALLOS					
			TIEMPO DETECCIÓN FALLO (n)		PORCENTAJE ALARMAS (%)		FALSAS ALARMAS (%)	
NOMBRE FALLO	TIPO	VALOR	T2	Q	T2	Q	T2	Q
NIVEL 5º DEPÓSITO	OFFSET	2	94	325	6,6907	23,4309	1,1765	14,2405
	OFFSET	2,4	110	306	8,0759	19,2838	9,0000	6,0606
NIVEL 4º DEPÓSITO	OFFSET	2	180	345	5,8200	23,2443	1,1696	0,2976
	OFFSET	2,4	118	206	6,7891	22,8677	0,9174	5,5838
CAUDAL Q7	OFFSET	2000	NONE	205	NONE	29,0407	NONE	3,0612
	OFFSET	3500	NONE	206	NONE	33,1158	NONE	3,0457
CAUDAL Q8	OFFSET	6000	1615	210	2,4474	35,6490	0,1868	1,4925
	OFFSET	8500	327	210	3,3296	36,6289	0,3145	0,4975
CAUDAL DEPURADORA	OFFSET	20000	25	555	0,4187	22,8632	6,2500	4,5788
	OFFSET	23000	125	295	3,8700	33,4369	0,8621	5,5944

Figura 39. Tabla de resultados de PCA no lineal con umbral adaptativo.

Así como en el modelo de PCA lineal la introducción del umbral adaptativo conseguía una gran mejora del modelo en la detección, en el caso del PCA no lineal los resultados con umbral adaptativo son peores incluso que con el umbral constante. De la Figura 39 se puede observar que T^2 no consigue detectar los fallos provocados en el caudal 7 (valores marcados en rojo), y en la mitad de las muestras de fallos el tiempo de detección de Q es para $n < 480$, lo que significa que se detecta antes de que se produzca, lo cual es erróneo. El porcentaje de alarmas en el estadístico T^2 es muy bajo (la mayoría por debajo del 8%), y aunque este parámetro mejora ligeramente en Q con una media de 27,95%, siguen siendo valores que se encuentran muy lejos de ser válidos. Únicamente Q consigue detectar un único fallo en el caudal de la depuradora con relativa rapidez ($n=555$)

Si se centra el estudio en la observación de las gráficas, la evaluación del método sigue arrojando malos resultados.

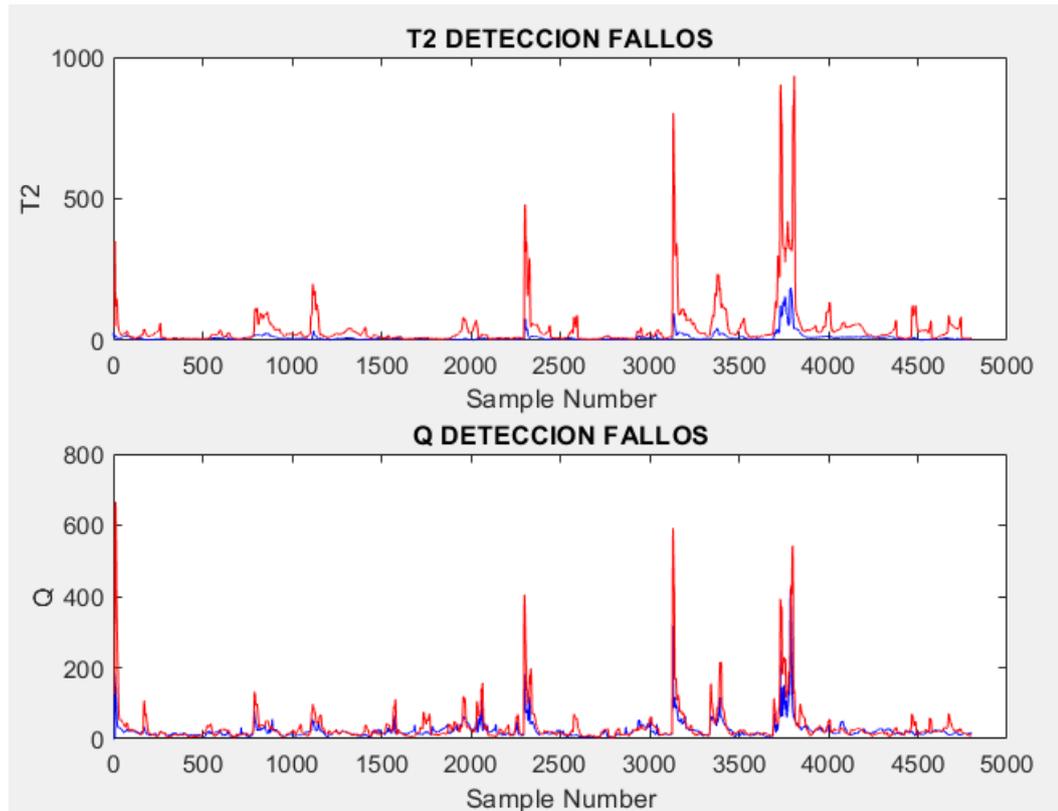


Figura 40. (a) Arriba. Gráfico de control T^2 para el fallo en q8 (offset=8500). (b) Abajo. Gráfico de control Q para el fallo en q8 (offset=8500).

En el fallo de q8 (Figura 40 (a) y (b)) ninguno de los estadísticos experimenta algún tipo de perturbación alrededor de la observación $n=500$ (momento aproximado donde se provoca el fallo), dejando ver el mal funcionamiento del modelo para algunos fallos como este.

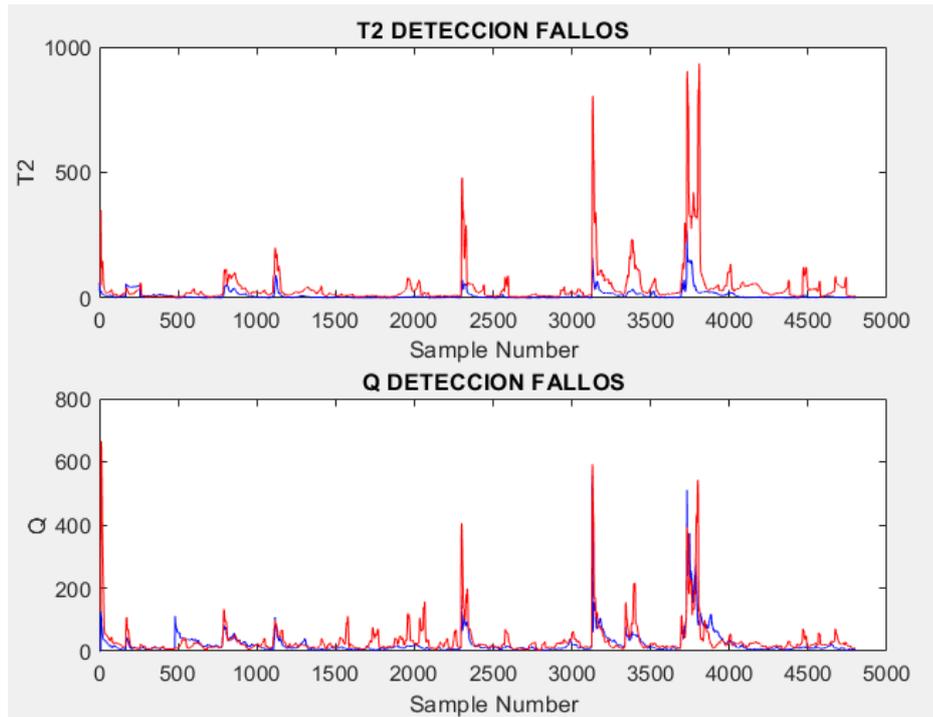


Figura 41. (a) Arriba. Gráfico de control T^2 para el fallo en n_4 (offset=2). (b) Abajo. Gráfico de control Q para el fallo en n_4 (offset=2).

En otros fallos como el de la Figura 41, el estadístico Q si que se ve perturbado por el fallo en $n=500$. Sin embargo, la detección se produce antes debido a que el umbral adaptativo se encontraba por debajo del estadístico incluso antes de provocarse el fallo.

El modelo de PCA no lineal ha estado lejos del comportamiento esperado, por lo que en el apartado siguiente se toma otro camino para la búsqueda de la mejora del PCA lineal.

4.2.3. PCA DEL RESIDUO

Este nuevo modelo de PCA incluye una parte no lineal que involucra redes neuronales, y una parte lineal igual que el primer modelo. El objetivo con este nuevo modelo es el de amplificar el fallo para que, tanto con un umbral

constante como con uno adaptativo, se puedan detectar los fallos de forma rápida y sencilla.

4.2.3.1. PCA DEL RESIDUO FUERA DE LÍNEA

Primero se normaliza la matriz X de datos normales a $+1$, -1 , y se crea una red neuronal con la misma arquitectura que en Figura 33: red simétrica con 3 capas ocultas de 30, 10 y 30 neuronas. Se toma la misma arquitectura que en el apartado anterior para que la mejora sea acorde a lo que se ha trabajado anteriormente. El entrenamiento de esta red tiene como datos de entrada y datos objetivo (valores que la red tiene que intentar replicar en la salida) la matriz X de datos normales normalizada. Tras finalizar el entrenamiento, se obtiene la salida de la red y se calcula el residuo como la diferencia entre los datos objetivo y la salida real (Ecuación 24). Donde r es el residuo, X_n es la matriz X de datos normales normalizada, y X_{red} es la salida de la red tras el entrenamiento.

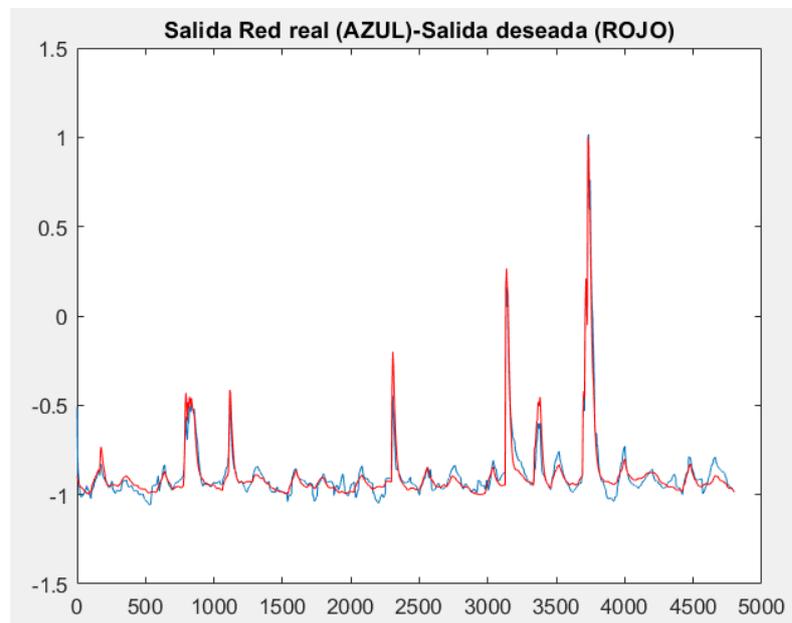


Figura 42. Representación del entrenamiento de la red.

En la Figura 42 se ilustra cómo ha sido el entrenamiento. La curva azul es la salida de la red (X_{red}), mientras que la curva roja es la salida objetivo o deseada (X_n)

Ahora se somete a la matriz r a un PCA lineal desarrollado anteriormente, con una variabilidad de los componentes principales mayor o igual al 95%. Ahora si se calculan los dos estadísticos T^2 y Q y sus respectivos umbrales T_α y Q_α . También se procede a calcular los umbrales adaptativos de ambos estadísticos de la misma manera que en el resto de los modelos.

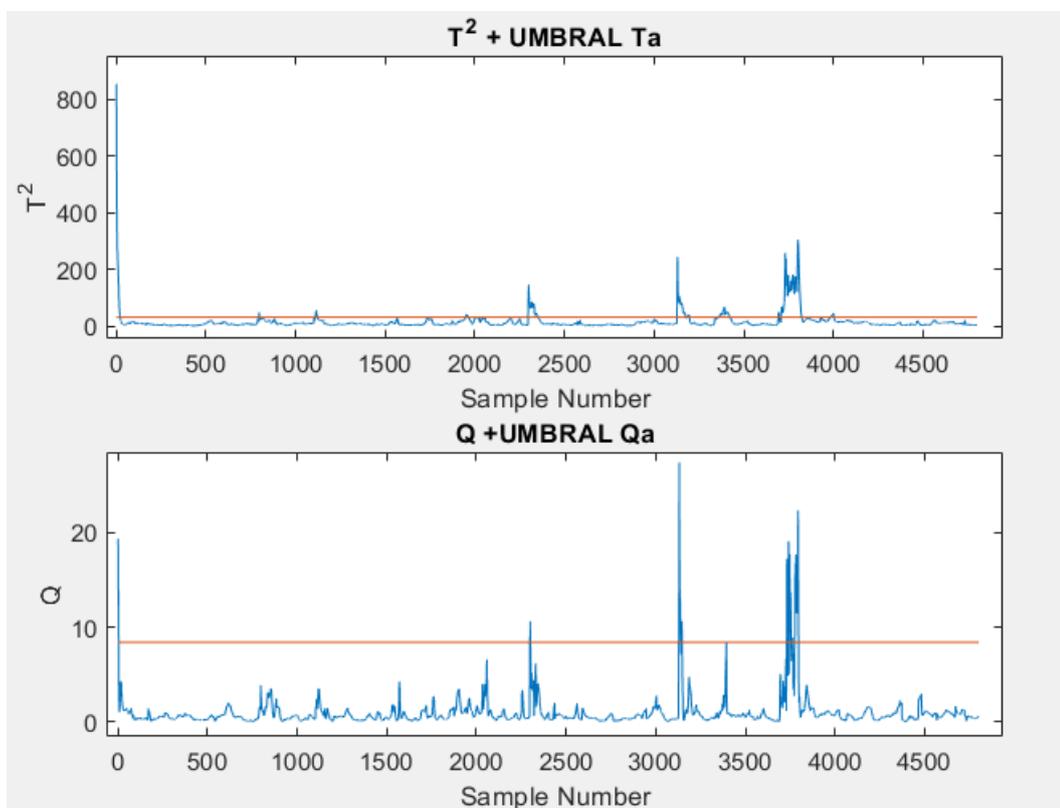


Figura 43. (a) Arriba. Gráfico de control T^2 para el funcionamiento normal con umbral constante. (b) Abajo. Gráfico de control Q funcionamiento normal con umbral constante.

En las Figuras 43 y 44 se muestra el resultado del PCA no lineal de los residuos aplicados a los datos de comportamiento normal (sin fallo) para umbral constante y adaptativo respectivamente. Llegados a este punto, se procede al

tratamiento de los datos de fallo para poder incluirlos en el modelo y verificar si se produce la detección y cuál es su efectividad.

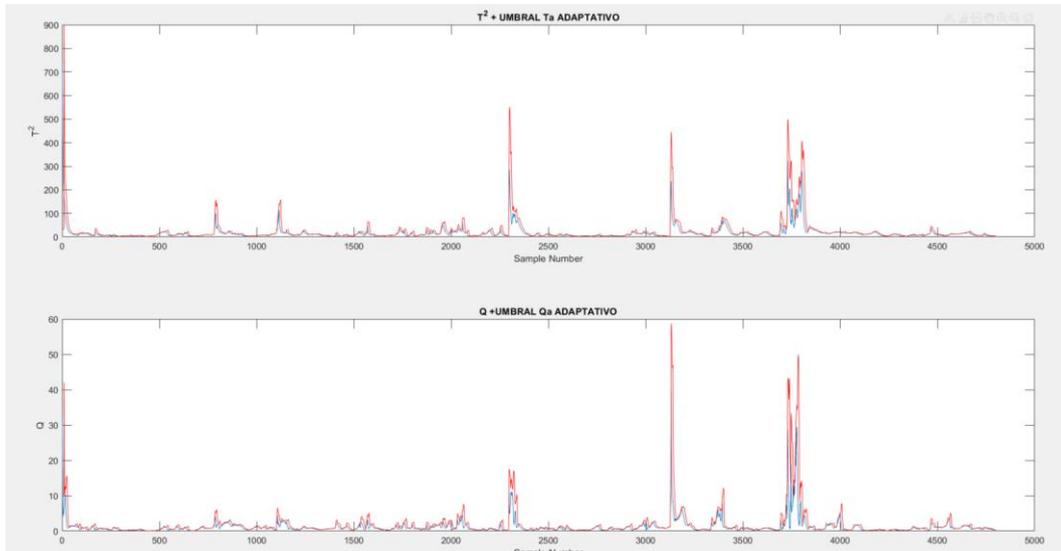


Figura 44. (a) Arriba. Gráfico de control T^2 para funcionamiento normal con umbral adaptativo. (b) Abajo. Gráfico de control Q para funcionamiento normal con umbral adaptativo.

4.2.3.2. PCA DEL RESIDUO ONLINE

El tratamiento de los datos de fallo es similar al tratamiento de los datos de comportamiento bajo control de la planta. Los datos son normalizados a -1 y +1, pero con respecto a los valores de la normalización de los datos de funcionamiento normal. Después se introducen en la red ya creada, y se calcula r :

$$rf = Xf - Xredf \quad (25)$$

Donde Xf es la matriz de datos de fallo normalizada y $Xredf$ es la salida de la red con Xf como entrada. Se aplica el modelo de PCA lineal a rf , pero la normalización con media 0 y varianza 1 se realiza con respecto a los valores de la normalización de r fuera de línea. De aplicar PCA se obtienen T^2 y Q

referentes a rf , y se obtienen los resultados correspondientes al comparar estos estadísticos con los umbrales calculados fuera de línea.

4.2.3.3. RESULTADOS PCA CON RESIDUO Y UMBRAL CONSTANTE

Los resultados obtenidos con el umbral constante no son excelentes, pero se mejora con claridad el modelo con PCA no lineal.

			ESTUDIO DETECCIÓN FALLOS					
			TIEMPO DETECCION FALLO (n)		PORCENTAJE ALARMAS (%)		FALSAS ALARMAS (%)	
NOMBRE FALLO	TIPO	VALOR	T2	Q	T2	Q	T2	Q
NIVEL 5º DEPÓSITO	OFFSET	2	481	481	31,2037	75,1389	2,7542	1,2712
	OFFSET	2,4	481	481	94,0509	80,2083	2,7542	1,2712
NIVEL 4º DEPÓSITO	OFFSET	2	481	481	90,463	76,875	2,7542	1,2712
	OFFSET	2,4	481	481	100	100	2,7542	1,2712
CAUDAL Q7	OFFSET	2000	1729	NONE	17,7734	NONE	2,4419	NONE
	OFFSET	3500	481	885	38,4722	22,7783	2,7542	1,5982
CAUDAL Q8	OFFSET	6000	1609	1609	27,005	3,4461	4,4375	0,375
	OFFSET	8500	522	1461	44,8469	15,6886	2,5341	0,4132
CAUDAL DEPURADORA	OFFSET	20000	515	818	50,2333	16,8968	2,5692	0,7417
	OFFSET	23000	418	545	96,7593	64,1682	2,7542	1,1194

Figura 45. Resultados PCA con residuo y umbral constante.

Los fallos en los niveles de los depósitos son detectados de forma instantánea, y además tienen un porcentaje de alarmas muy alto, alguno incluso del 100%. Sin embargo, los fallos de los caudales no tienen un correcto funcionamiento en este modelo, ya que se detectan tarde tanto en q7 como en q8, e incluso un fallo en q7 no es siquiera detectado por el estadístico Q. En cuanto a las falsas alarmas, los datos siguen siendo positivos, ya que en ningún fallo se supera el 5%. En la Figura 46 se puede observar el buen funcionamiento de este modelo en algunos fallos, como es el ejemplo del fallo en el nivel del 4º depósito. El fallo es muy visual y se aprecia que efectivamente, puesto que se está trabajando con el residuo del modelo neuronal y no con las variables directamente, la detección funciona mejor debido a que se está amplificando el fallo.

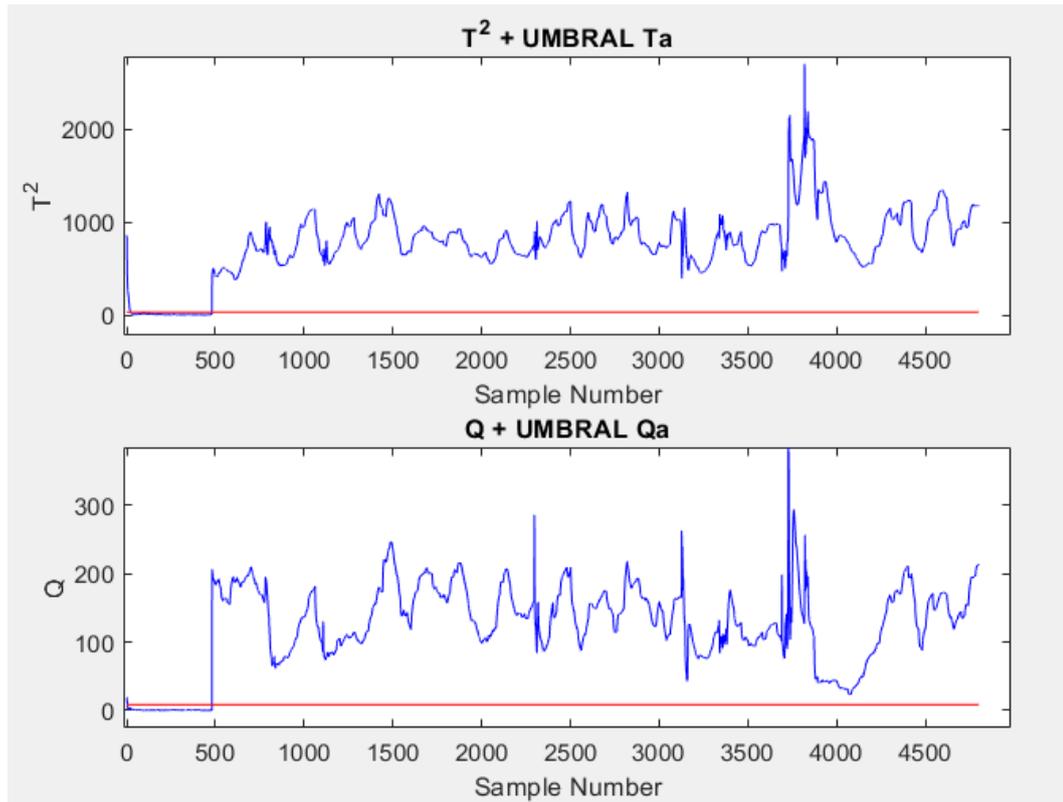


Figura 46. (a) Arriba. Gráfico de control T^2 para el fallo n4 (offset=2,4) con umbral constante. (b) Abajo. Gráfico de control Q para el fallo n4 (offset=2,4) con umbral constante.

4.2.3.4. RESULTADOS PCA CON RESIDUO Y UMBRAL ADAPTATIVO

Si se introduce el umbral adaptativo en el modelo del PCA con residuo, los resultados que se obtienen son prácticamente inmejorables en cuanto a detección de fallos se refiere:

			ESTUDIO DETECCIÓN FALLOS					
			TIEMPO DETECCIÓN FALLO (n)		PORCENTAJE ALARMAS (%)		FALSAS ALARMAS (%)	
NOMBRE FALLO	TIPO	VALOR	T2	Q	T2	Q	T2	Q
NIVEL 5º DEPÓSITO	OFFSET	2	481	481	97,1528	98,8194	0,8475	0,6356
	OFFSET	2,4	481	481	95,7639	98,8889	0,8475	0,6356
NIVEL 4º DEPÓSITO	OFFSET	2	481	481	99,0741	99,3287	0,8475	0,6356
	OFFSET	2,4	481	481	100	100	0,6356	0,6356
CAUDAL Q7	OFFSET	2000	481	481	74,9074	88,9583	0,6356	0,6356
	OFFSET	3500	481	481	90,3472	97,4074	0,6356	0,6356
CAUDAL Q8	OFFSET	6000	518	418	72,029	76,6435	0,5894	0,6356
	OFFSET	8500	481	481	85,625	88,1019	0,6356	0,6356
CAUDAL DEPURADORA	OFFSET	20000	481	543	85,9859	69,9859	0,6356	3,1835
	OFFSET	23000	481	481	97,7083	97,0139	0,6356	0,6356

Figura 47. Resultados de PCA con residuo y umbral adaptativo

Ahora todos los fallos se detectan de forma inmediata por al menos uno de los dos estadísticos. Además, el porcentaje de alarmas es el más alto de los obtenidos hasta ahora, con una media de 89,85% en T^2 y 91,51% en Q. Como comentario final, decir que el porcentaje de falsas alarmas no supera el 1% en ninguna ocasión, lo que es otro indicador más del buen funcionamiento de este modelo. En cuando a la parte gráfica de los fallos, se consigue amplificar ampliamente cada fallo, por lo que el umbral adaptativo es incapaz de seguir al estadístico una vez se provoca el fallo.

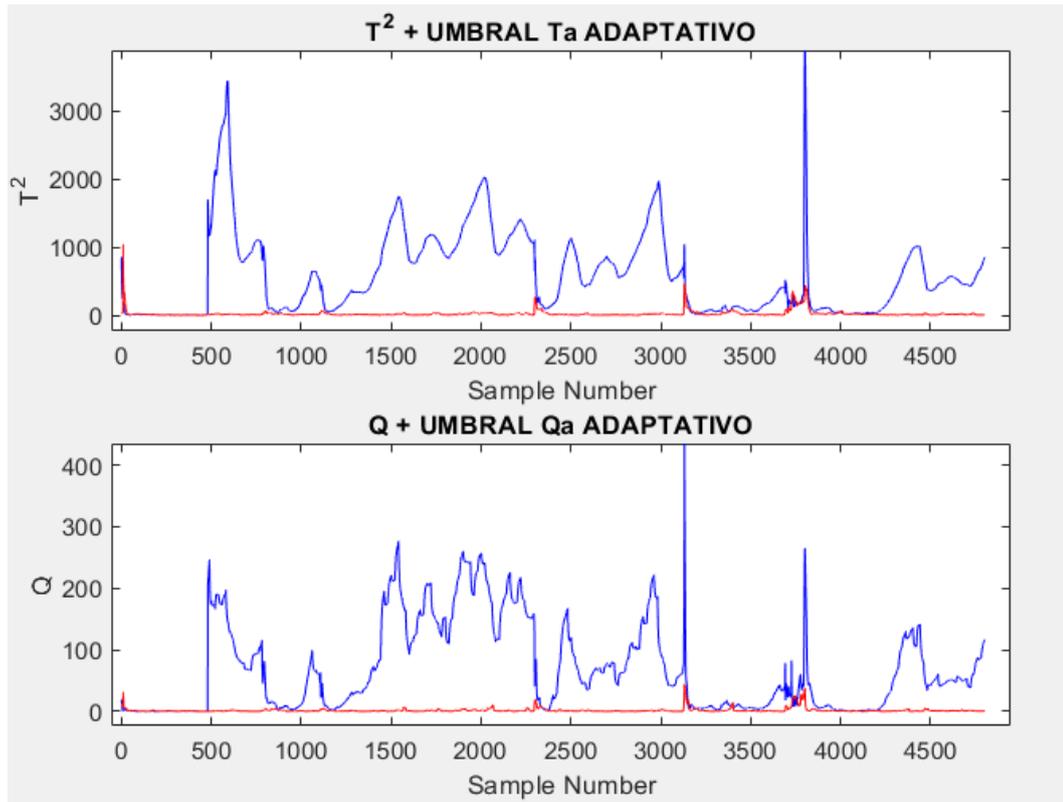


Figura 48. (a) Arriba. Gráfico de control T^2 para el fallo n5 (offset=2) con umbral adaptativo. (b) Abajo. Gráfico de control Q para el fallo n5 (offset=2) con umbral adaptativo.

En la Figura 48 (a) se observa la diferencia de orden de magnitud entre el umbral (rojo) y el estadístico (azul) tras provocar el fallo en la observación $n=480$. En Q, el umbral se encuentra sobre los valores 0 y 1, mientras que, tras el fallo, el estadístico Q alcanza 500 en el máximo de la curva. En T^2 , el umbral adaptativo se encuentra en torno al valor 6 en todo el intervalo de observaciones, mientras que, tras el fallo, se produce un máximo con valor 3228. Esta gran diferencia de orden de magnitud entre los umbrales y los estadísticos permite una gran rapidez para la detección del fallo, e ilustra de manera clara y concisa el momento de detección del fallo en la gráfica.

4.3. MODELOS PARA LA CLASIFICACIÓN DE FALLOS

Para el diagnóstico de los fallos detectados, se han desarrollado tres modelos distintos. Los resultados que aporten cada uno serán estudiados y comparados entre sí para poder determinar qué modelo clasifica de mejor manera las anomalías en la planta. Uno de los modelos estará basado en los árboles de decisión, y en los otros dos se aplicarán redes neuronales de distinto número de capas ocultas

4.3.1. MODELO DE RED NEURONAL MONOCAPA

Este modelo utiliza las redes neuronales como herramienta para el diagnóstico de los fallos. En este caso la red será de propagación hacia adelante, y con sólo una única capa oculta.

En primer lugar, se cargan los datos de entrenamiento en el programa. Estos datos de entrenamiento serán las matrices X_{fn} de n fallos provocados, pero en los que se eliminarán las observaciones del primer día, es decir, las primeras 480 observaciones. Esto se debe a que durante esas observaciones no se había provocado el fallo, por lo que ese intervalo de valores no se identifica con ningún fallo, y es inútil para el entrenamiento de la red. La red tendrá la misión de clasificar 5 fallos, por lo que serán 5 las matrices X_{fn} (X_{f1} ; X_{f2} ; X_{f3} ; X_{f4} ; X_{f5}).

La red creada tendrá una única capa oculta y será de propagación hacia adelante o *feedforward*, y únicamente se elegirá el número de neuronas de la capa oculta, ya que el número de neuronas en la capa de entrada vendrá determinado por el tamaño de los datos de entrada, y el de la capa de salida, por el tamaño de los datos objetivo. Los datos de entrada de la red son los datos de fallo agrupados en una matriz $X=[X_{f1} X_{f2} X_{f3} X_{f4} X_{f5}]$, por lo tanto el número de entradas de la red será 24, igual al número de filas de la matriz X . La matriz de datos de salida deseada consistirá en una matriz de 5 columnas (mismo número de columnas que de fallos a clasificar). En cuanto al número de filas:

$$n^{\circ} \text{ filas} = 4801 * 5 - (480 * 5) = 21605 \text{ filas}$$

Debido a que, tal como se ha comentado anteriormente, se han eliminado las primeras 480 observaciones. Los primeros 4321 valores de la primera columna tendrán el valor 1, siendo el resto 0. Para la segunda columna, serán los valores comprendidos entre la fila 4322 y 8642 los que tengan valor 1, siendo el resto de las filas 0. En la tercera columna, tendrán valor 1 de la fila 8643 a la 12963, y el resto 0. Para la cuarta columna, los valores unidad serán los comprendidos entre las filas 12964 y 17284, con el resto de filas valor 0. En cuanto a la quinta columna, todas sus filas tendrán valor 0 excepto las 4321 últimas.

Se elige un número de neuronas de la capa oculta, y se entrena la red con los datos de entrada X y los datos de salida deseados Y. En función del número de neuronas en la capa oculta, el entrenamiento de la red será mejor o peor.

	1	2	3	4	5	
1	4321 20.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	4321 20.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	4321 20.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	4321 20.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4321 20.0%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
	1	2	3	4	5	
	Target Class					

Figura 49. Matriz de Confusión del entrenamiento con 18 neuronas.

Si se eligen 18 neuronas como tamaño de la capa oculta de la red, el entrenamiento es excelente, pues se logra clasificar la totalidad de los datos correctamente (Figura 49). Las Figuras 50 y 51 muestran la calidad de los entrenamientos con distinto número de neuronas, siendo más ineficientes que el de 18.

Output Class	1	2	3	4	5	
1	4303 19.9%	0 0.0%	18 0.1%	0 0.0%	0 0.0%	99.6% 0.4%
2	4090 18.9%	0 0.0%	100 0.5%	121 0.6%	10 0.0%	0.0% 100%
3	33 0.2%	0 0.0%	4272 19.8%	16 0.1%	0 0.0%	98.9% 1.1%
4	11 0.1%	0 0.0%	16 0.1%	4292 19.9%	2 0.0%	99.3% 0.7%
5	31 0.1%	0 0.0%	13 0.1%	4268 19.8%	9 0.0%	0.2% 99.8%
	50.8% 49.2%	NaN% NaN%	96.7% 3.3%	49.4% 50.6%	42.9% 57.1%	59.6% 40.4%
Target Class	1	2	3	4	5	

Figura 50. Matriz de confusión del entrenamiento con 4 neuronas.

Output Class	1	2	3	4	5	
1	4321 20.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	7 0.0%	4283 19.8%	31 0.1%	0 0.0%	0 0.0%	99.1% 0.9%
3	0 0.0%	1 0.0%	4317 20.0%	3 0.0%	0 0.0%	99.9% 0.1%
4	0 0.0%	0 0.0%	0 0.0%	4321 20.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4321 20.0%	100% 0.0%
	99.8% 0.2%	100.0% 0.0%	99.3% 0.7%	99.9% 0.1%	100% 0.0%	99.8% 0.2%
Target Class	1	2	3	4	5	

Figura 51. Matriz de confusión del entrenamiento con 10 neuronas.

Otra forma para comprobar que el entrenamiento de la red es correcto, consiste en ilustrar mediante gráficas las salidas de la red, como se ha hecho en la Figura 52.

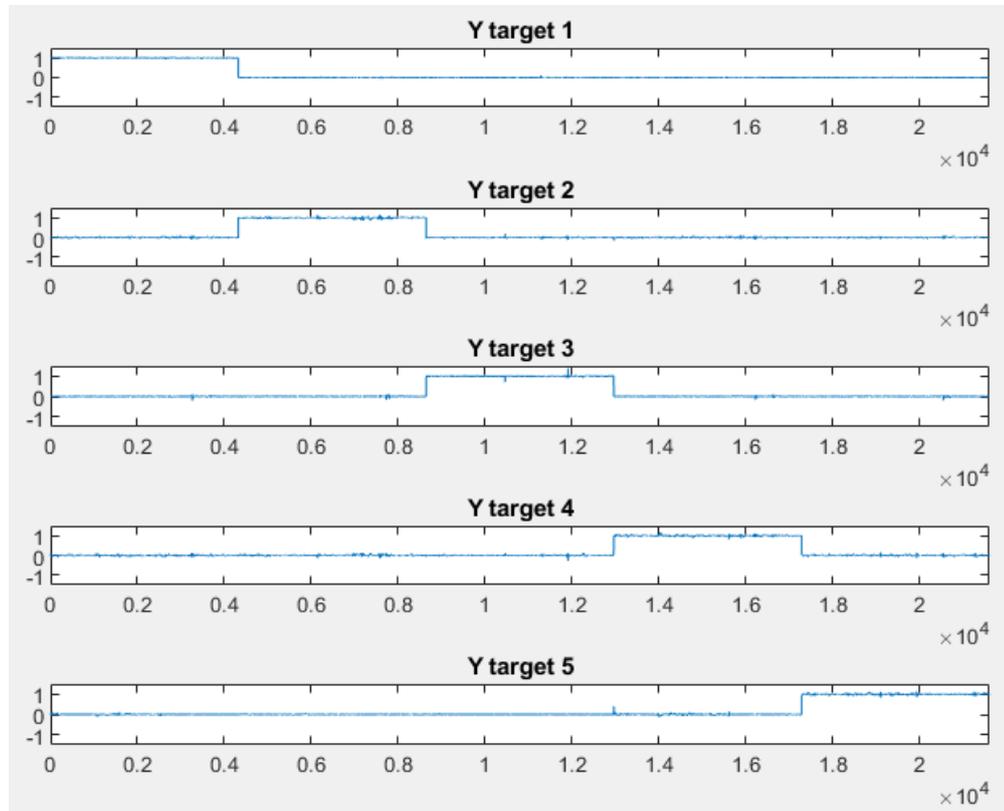


Figura 52. Entrenamiento de la red.

La Figura 52 muestra que, efectivamente, la salida de la red es la misma que la salida deseada, ya que tiene el valor 1 en el lugar que le corresponde a cada fallo de entrenamiento, y es 0 en el resto.

Para validar este modelo de clasificación de fallos, se introducen fallos de la misma clase que los de entrenamiento, pero de distinta magnitud, y así se puede observar si el modelo es capaz de diagnosticar los fallos de la misma manera que con los fallos de entrenamiento. Primero se va a trabajar bajo la hipótesis de que únicamente se produce un fallo en la planta al mismo tiempo, es decir, el modelo sólo tendrá que clasificar una clase de fallo a la vez. Para la visualización de los resultados, se prueba con un fallo de mayor magnitud en el caudal 8. Los resultados son excelentes ya que q8 tiene el valor 1 durante

todas las observaciones, y no se detecta ningún otro tipo de fallo, salvo pequeñas variaciones (Figura 53).

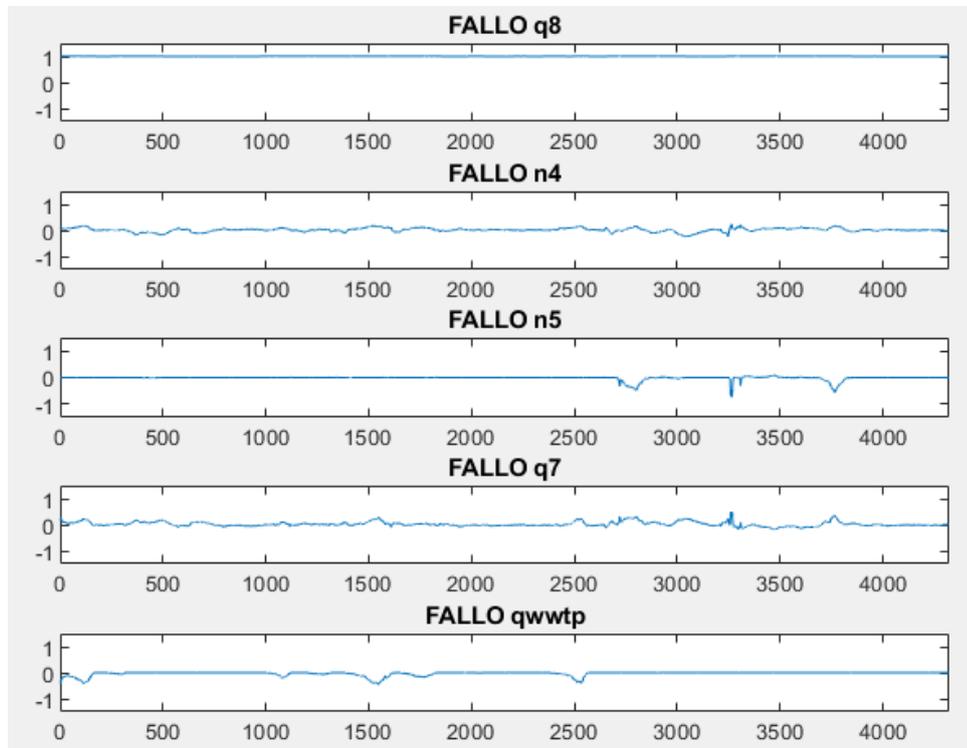


Figura 53. Validación del fallo en q8.

Los resultados numéricos obtenidos con el resto de fallos se resumen en la tabla de resultados de la Figura 54.

FALLO	PORCENTAJE CLASIFICACIÓN (%)
q8	100
n4	76,9729
n5	81,1618
q7	100
qwwtp	0,1157

Figura 54. Resumen de resultados para la clasificación de 1 fallo por parte del modelo monocapa.

Se introducen ahora 2 fallos en el modelo, con el objetivo de comprobar la efectividad del funcionamiento del modelo cuando trabaja con cantidades más elevadas de datos. La Figura 55 muestra un ejemplo de los resultados tras

introducir los fallos q8 y n5 simultáneamente. La red confunde ligeramente los fallos n5 y q7, pero en líneas generales la clasificación es buena.

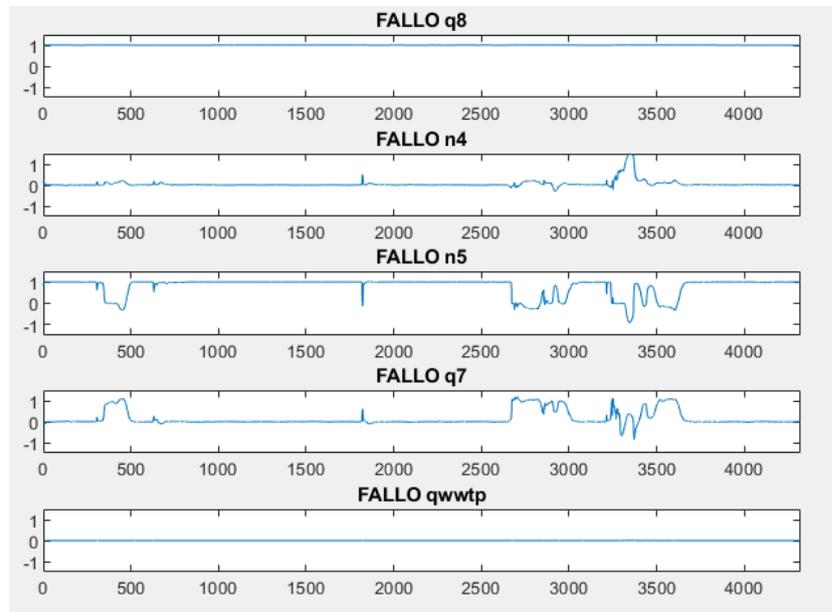


Figura 55. Validación del fallo en la pareja de fallos q8 y n5.

Las Figuras 56 y 57 muestran más ejemplos gráficos de la validación con parejas de fallos.

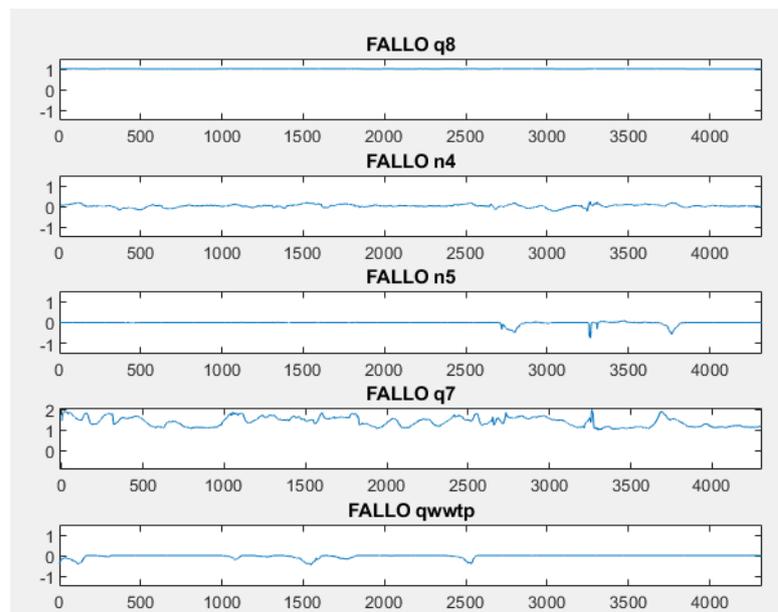


Figura 56. Validación del fallo en la pareja de fallos q8 y q7 con el modelo monocapa.

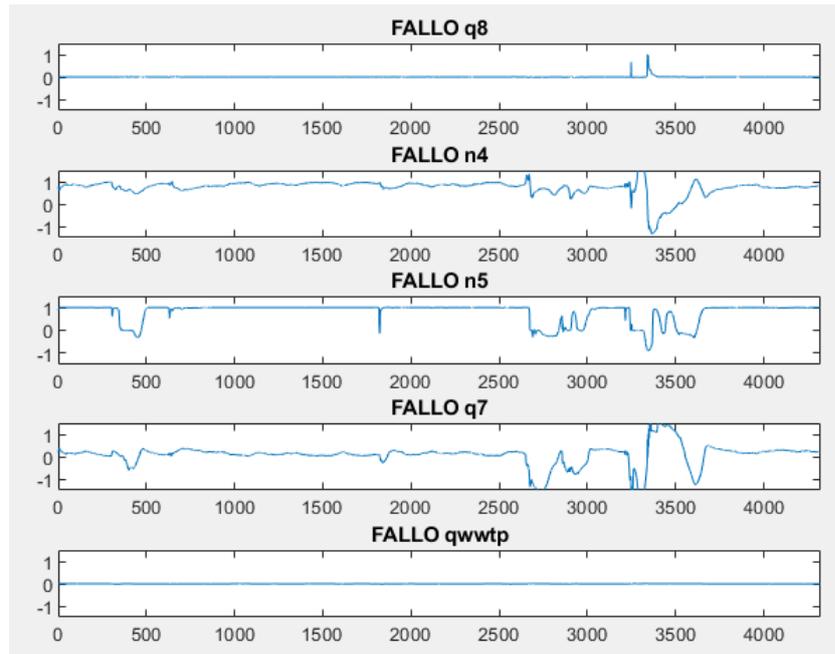


Figura 57. Validación del fallo en la pareja de fallos n4 y n5 con el modelo monocapa.

El funcionamiento del modelo con parejas de fallos es bueno, salvo en aquellas parejas con el fallo en qwwtp, ya que la red no consigue clasificar este fallo. Existen fallos que son clasificados al 100%, como el fallo en q8 o el fallo en q7. Los fallos en los niveles también se clasifican correctamente, con porcentajes por encima del 70%. En la tabla de la Figura 58 se muestra el resumen de resultados de todas las configuraciones de parejas posibles.

	q8 (%)	n4 (%)	OTRO (%)
q8 + n4	50,0694	38,4865	8,9910
	q8 (%)	n5 (%)	OTRO (%)
q8 + n5	50,2430	40,5809	7,3825
	q8 (%)	q7 (%)	OTRO (%)
q8 + q7	50,0000	50,0000	0,0000
	q8 (%)	qwwtp (%)	OTRO (%)
q8 + qwwtp	50,0000	0,0579	33,9274
	n4 (%)	n5 (%)	OTRO (%)
n4 + n5	39,3312	47,3964	9,0257
	n4 (%)	q7 (%)	OTRO (%)
n4 + q7	38,4865	52,1754	6,8850
	n4 (%)	qwwtp (%)	OTRO (%)
n4 + qwwtp	61,0275	0,0579	20,4466
	n5 (%)	q7 (%)	OTRO (%)
n5 + q7	40,5809	56,5378	1,0877
	n5 (%)	qwwtp (%)	OTRO (%)
n5 + qwwtp	45,4409	0,0579	36,6929
	q7 (%)	qwwtp (%)	OTRO (%)
q7 + qwwtp	56,5263	0,0579	27,4011

Figura 58. Resumen de resultados para parejas de datos con modelo monocapa.

Los porcentajes mostrados en la Figura 58 hacen referencia a la clasificación de los datos con respecto del total. Por lo tanto, el reparto ideal de porcentajes sería un 50-50% entre los dos fallos introducidos. Un parámetro importante de la tabla es el porcentaje que hace referencia a los datos que han sido clasificados en otros fallos (columna OTROS). Este porcentaje aumenta en aquellas parejas donde se encuentra el fallo qwwtp, ya que este modelo no es capaz de clasificarlo. La existencia de porcentajes superiores al 50% se debe a que hay datos mal clasificados, pero al ser clasificados dentro del fallo que se está validando aumenta su valor. Sin embargo, estos datos son una minoría en la mayoría de los casos.

Si se pide a la red que clasifique los 5 fallos a la vez, es decir, que trabaje con todos los datos de fallo simultáneamente, se puede observar los resultados obtenidos a través de una matriz de confusión. Dicha matriz de confusión se muestra en la Figura 59. Como se comentó en el Capítulo II, las celdas que

forman la diagonal representan los datos correctamente clasificados para cada uno de los fallos, y la última celda de la diagonal calcula el comportamiento global de la clasificación. El porcentaje global de datos correctamente clasificados es del 73,2%, un porcentaje elevado si se tiene en cuenta que, como se ha comprobado anteriormente y se comprueba en la Figura 59 de nuevo, el modelo no es capaz de clasificar el fallo en qwwtp.

Output Class	1	2	3	4	5	
1	4321 20.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	5 0.0%	3567 16.5%	530 2.5%	219 1.0%	0 0.0%	82.6% 17.4%
3	24 0.1%	79 0.4%	3575 16.5%	643 3.0%	0 0.0%	82.7% 17.3%
4	0 0.0%	0 0.0%	0 0.0%	4321 20.0%	0 0.0%	100% 0.0%
5	5 0.0%	2549 11.8%	552 2.6%	1194 5.5%	21 0.1%	0.5% 99.5%
	99.2% 0.8%	57.6% 42.4%	76.8% 23.2%	67.8% 32.2%	100% 0.0%	73.2% 26.8%
Target Class	1	2	3	4	5	

Figura 59. Matriz de confusión para la clasificación de 5 fallos simultáneamente con el modelo monocapa.

4.3.2. MODELO DE RED NEURONAL MULTICAPA

En este modelo se aumentará el número de capas ocultas (Deep Learning) con el objetivo de que la red pueda trabajar con un número mayor de datos, y así poder mejorar los resultados obtenidos en el modelo anterior.



La entrada y la salida de la red serán las mismas que en la red monocapa, ya que el objetivo de ambos modelos es el mismo. Por lo tanto, la entrada será la matriz $X=[X_{f1} X_{f2} X_{f3} X_{f4} X_{f5}]$, y la salida será la matriz Y , idéntica que en el modelo de red neuronal monocapa.

La diferencia entre ambos modelos se encuentra en la arquitectura de la red. En este modelo, la red se diseñará con Autoencoders, y una última capa cuyo número de neuronas será igual al tamaño de la salida. El proceso de creación de la red comienza con el entrenamiento de un Autoencoder, cuyos datos de entrada para el entrenamiento será la matriz X normalizada entre -1 y $+1$, por lo que el tamaño de las capas de entrada y salida será de 24 neuronas. La capa oculta tendrá un tamaño de 40 neuronas. Tras el entrenamiento, se codifica la capa oculta de este Autoencoder, que será la entrada para el entrenamiento del segundo. El segundo Autoencoder tendrá por lo tanto un tamaño de 40 neuronas en las capas de entrada y salida, y su capa oculta tendrá 15. Tras el entrenamiento del segundo Autoencoder, se vuelve a extraer o codificar su capa oculta, que será la entrada para el entrenamiento del tercer Autoencoder. Este será el último, con un tamaño en la capa oculta de 8 neuronas. Para finalizar las capas, se entrena la última capa, denominada softmax, cuyos datos de entrada para el entrenamiento será la capa oculta del tercer Autoencoder. Esta capa también precisa de datos de salida o deseados para el entrenamiento, que será la matriz Y . De esta manera, el tamaño de la entrada de la softmax es de 8, y la salida tiene tamaño 5. Finalmente se apilan los tres Autoencoders y la softmax, obteniendo la red completa. En la Figura 60 se representa el resultado del entrenamiento de cada uno de los 3 Autoencoders por separado. En rojo está representada la entrada, y en azul la salida que el Autoencoder intenta aproximar a la entrada.

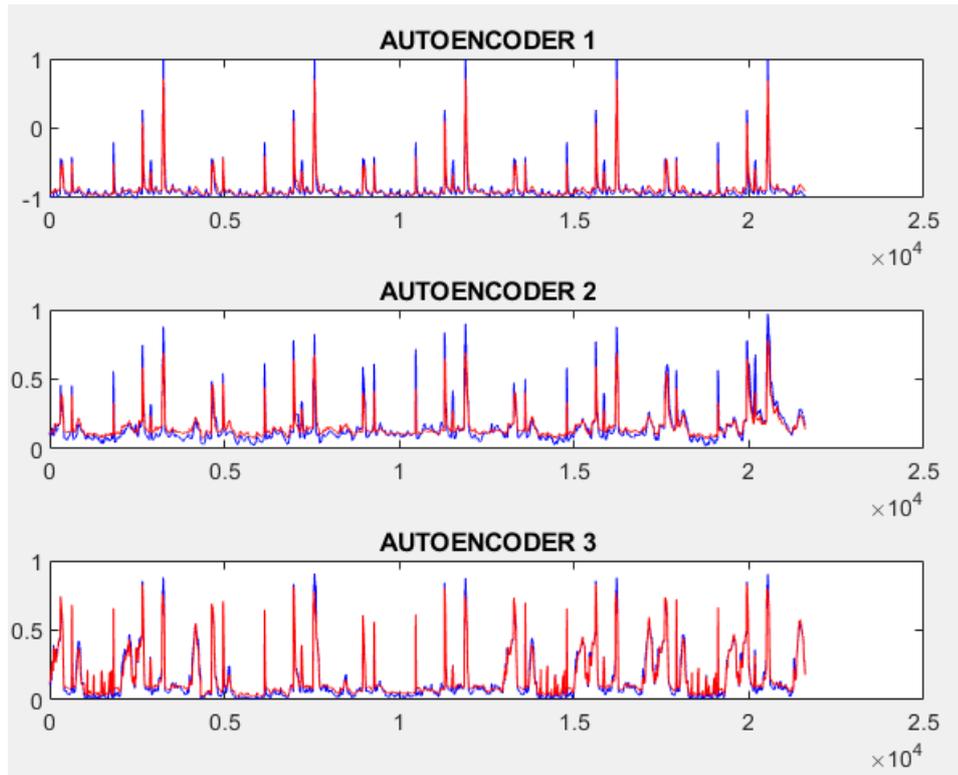


Figura 60. Resultado del entrenamiento de los Autoencoders por separado.

La red final también debe entrenarse. Sus datos de entrada para el entrenamiento será la matriz X normalizada entre -1 y $+1$, y los datos objetivos o deseados serán los correspondientes a la matriz Y . Para comprobar el entrenamiento de la red completa, se obtiene la salida de la red una vez entrenada, y se compara con la salida deseada. La herramienta ideal para visualizar dicha comparación es la matriz de confusión (Figura 61).

Output Class	1	2	3	4	5	
1	4321 20.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	4321 20.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	4321 20.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	4321 20.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4321 20.0%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
	1	2	3	4	5	
	Target Class					

Figura 61. Matriz de confusión de la efectividad del entrenamiento de la red completa.

El entrenamiento del modelo multicapa también consigue clasificar la totalidad de los datos correctamente, al igual que sucedía en el modelo monocapa.

Se procede ahora a la validación del método de clasificación, que al igual que anteriormente, consistirá en introducir fallos de la misma clase pero de distinta magnitud, para determinar si la red es capaz de clasificar estos nuevos datos de fallo que se introducen en ella. De la misma manera que en el modelo monocapa, primero se aumentará la magnitud de un solo fallo, y después se irá aumentando el número de fallos en la entrada a la red.

Se aumenta la magnitud en el fallo q7 (offset=3500), y se introducen estos datos de fallo en la red. La salida de la red aporta los resultados que se muestran en la Figura 62.

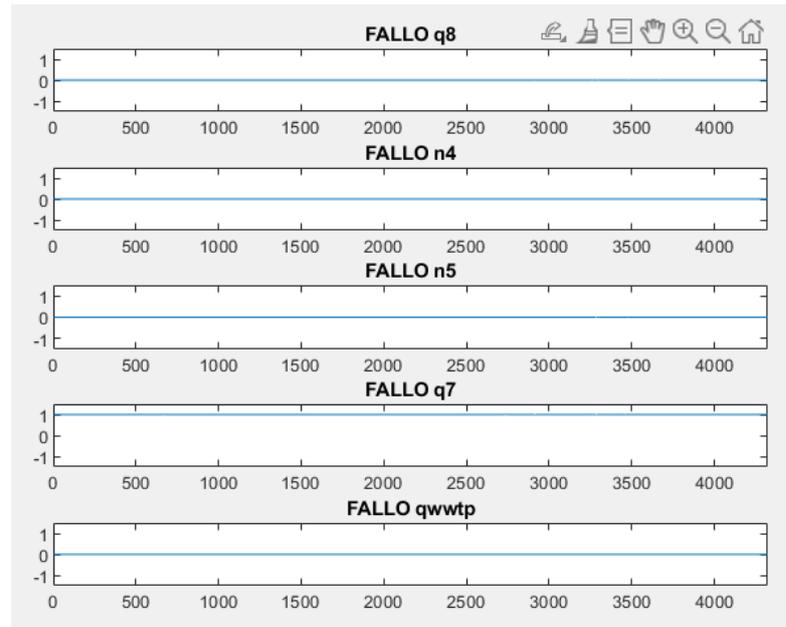


Figura 62. Validación del fallo en q7.

Los resultados de la gráfica son excelentes, ya que no existe ningún tipo de perturbación en todo el intervalo de datos. El fallo q7 se mantiene en 1 mientras que el resto de los fallos no son clasificados, y su valor es 0 para todo el intervalo. Los resultados numéricos para el resto de los fallos se incluyen en la tabla de la Figura 63.

FALLO	PORCENTAJE CLASIFICACIÓN (%)
q8	100,0000
n4	50,5439
n5	78,5003
q7	100,0000
qwwtp	57,5793

Figura 63. Resumen de resultados para la clasificación de 1 fallo por parte del modelo multicapa.

Se observa una gran mejora en la clasificación del fallo en el caudal de la depuradora (qwwtp), aunque la clasificación del fallo n4 es de peor calidad. Sin embargo, con este modelo se consigue clasificar todos los fallos en al menos un 50%, lo cual supone una característica diferencial con el modelo anterior.

En la validación con datos de entrada a la red correspondientes a dos fallos distintos, los resultados siguen siendo positivos (Figura 64 y 65). A continuación se muestran algunos ejemplos gráficos y la tabla de resultados.

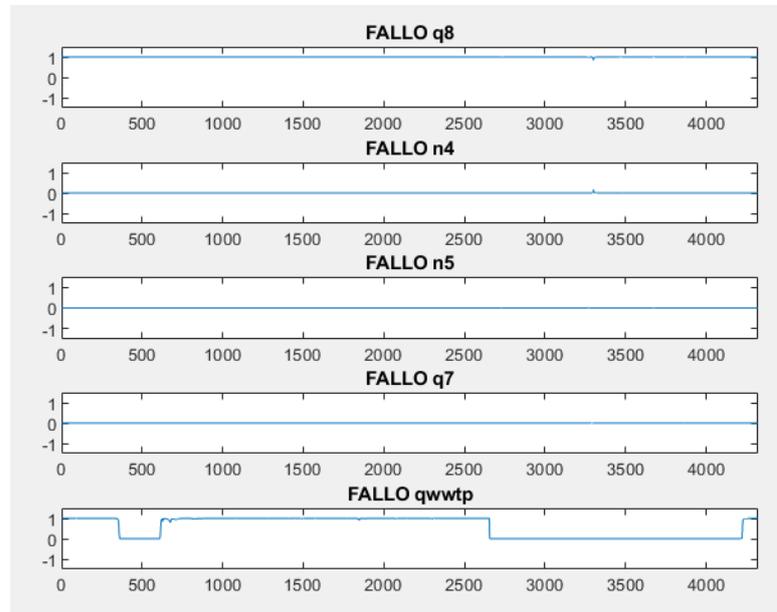


Figura 64. Validación del fallo en la pareja de fallos q8 y qwwtp con el modelo multicapa.

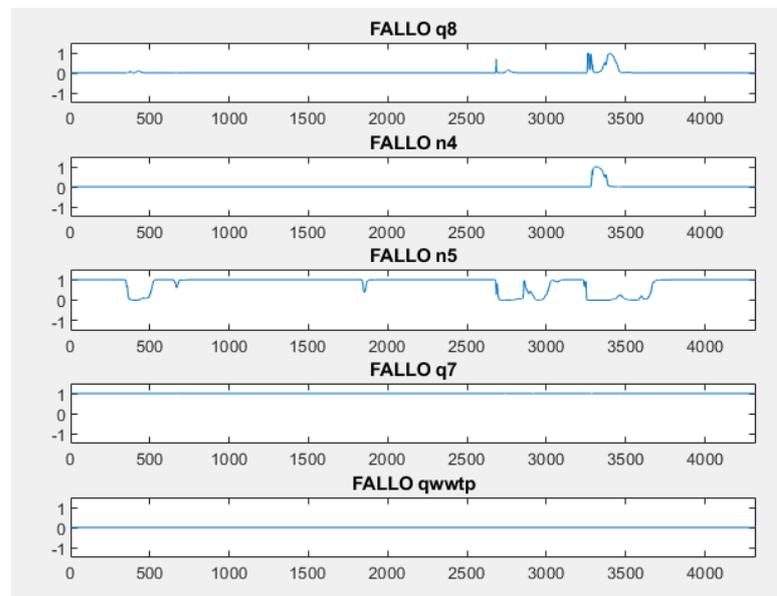


Figura 65. Validación del fallo en la pareja de fallos n5 y q7 con el modelo multicapa.

	q8 (%)	n4 (%)	OTRO (%)
q8 + n4	52,0713	25,2719	18,7457
	q8 (%)	n5 (%)	OTRO (%)
q8 + n5	50,7406	39,2502	7,6140
	q8 (%)	q7 (%)	OTRO (%)
q8 + q7	50,0000	50,0000	0,0000
	q8 (%)	qwwtp (%)	OTRO (%)
q8 + qwwtp	62,4508	28,7896	7,0933
	n4 (%)	n5 (%)	OTRO (%)
n4 + n5	26,0472	52,4300	15,2164
	n4 (%)	q7 (%)	OTRO (%)
n4 + q7	25,2719	54,6286	16,1884
	n4 (%)	qwwtp (%)	OTRO (%)
n4 + qwwtp	31,2659	29,7269	33,4298
	n5 (%)	q7 (%)	OTRO (%)
n5 + q7	39,2502	56,8387	1,5159
	n5 (%)	qwwtp (%)	OTRO (%)
n5 + qwwtp	39,9097	28,7896	27,2328
	q7 (%)	qwwtp (%)	OTRO (%)
q7 + qwwtp	50,4397	28,7896	19,1044

Figura 66. Resumen de resultados para parejas de datos con modelo multicapa.

Los porcentajes mayores del 50%, al igual que antes, se deben a los datos mal clasificados en fallos que son perfectamente clasificados, por lo que el porcentaje que debería ser 50 sube ligeramente. Vuelve a ser alto el porcentaje de datos mal clasificados en aquellas parejas donde interviene el fallo qwwtp, pero la clasificación de este fallo es superior al 25% en todas las situaciones. Es decir, más de la mitad de los datos correspondientes a ese fallo son clasificados correctamente. La suma de los 3 porcentajes de cada situación no suma 100% debido a que hay algunos datos que no son clasificados en ninguno de los 5 fallos.

Para finalizar con la validación, se introduce como entrada en la red los datos de los 5 fallos con aumento de magnitud. Los resultados pueden verse en la matriz de confusión mostrada en la Figura 67. El porcentaje de clasificación global es del 77,8%. A pesar de que consigue diagnosticar el fallo en qwwtp, este porcentaje únicamente sufre un aumento del 5% respecto del modelo

anterior. Esto se debe a que clasifica algo peor los fallos en los niveles (n4 y n5), pero es un modelo más completo puesto que consigue clasificar al menos un 50% de los datos de cada uno de los fallos.

Output Class	1	2	3	4	5	
1	4321 20.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	202 0.9%	2224 10.3%	1319 6.1%	458 2.1%	118 0.5%	51.5% 48.5%
3	94 0.4%	89 0.4%	3448 16.0%	690 3.2%	0 0.0%	79.8% 20.2%
4	0 0.0%	0 0.0%	0 0.0%	4321 20.0%	0 0.0%	100% 0.0%
5	1156 5.4%	560 2.6%	64 0.3%	49 0.2%	2492 11.5%	57.7% 42.3%
	74.8% 25.2%	77.4% 22.6%	71.4% 28.6%	78.3% 21.7%	95.5% 4.5%	77.8% 22.2%
	1	2	3	4	5	
	Target Class					

Figura 67. Matriz de confusión para la clasificación de 5 fallos simultáneos con el modelo multicapa.

4.3.3. MODELO DE ARBOLES DE DECISIÓN

El tercer y último modelo que se realizó para la clasificación de los fallos en una planta industrial está basado en árboles de decisión. Al igual que sucedía con los modelos basados en redes neuronales, en este modelo también existe una etapa de entrenamiento y otra de comprobación o validación del modelo.

En primer lugar, se crea una matriz $X=[X_{f1} X_{f2} X_{f3} X_{f4} X_{f5}]$, idéntica a la creada en los anteriores modelos de clasificación de fallos. A continuación, se crea un vector Y con 21605 elementos, mismo número de elementos que filas tiene la matriz X. Estos elementos son categóricos en vez de numéricos, y representan cada una de las cinco clases de fallos a clasificar. Puesto que existen cinco

clases distintas, los 4321 primeros elementos son de clase 1, del 4322 al 8642 son de clase 2, del 8643 al 12963 son de clase 3, del 12964 al 17284 son de clase 4, y del 17285 al 21605 son de clase 5. Con las matrices X e Y, se entrena un conjunto de árboles de decisión o bosque, especificando el número de árboles que lo forman.

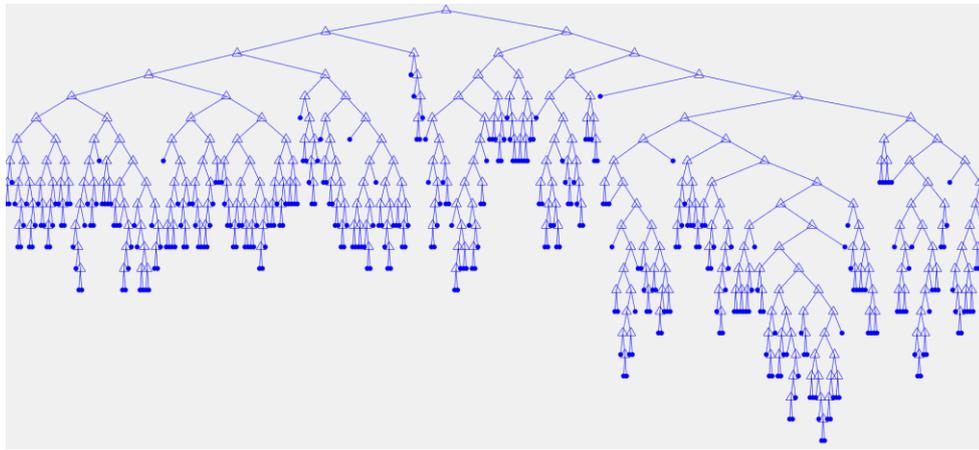


Figura 68. Estructura de un árbol ensamblado en el bosque.

El entrenamiento se ha realizado para distinto número de árboles, con el objetivo de encontrar un entrenamiento óptimo del bosque. La matriz de confusión que representa la calidad del entrenamiento no varía para el número de árboles, pero el tamaño del bosque si influirá en la validación.

Confusion Matrix

Output Class	1	4321 20.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	0 0.0%	4321 20.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	3	0 0.0%	0 0.0%	4321 20.0%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	4321 20.0%	0 0.0%	100% 0.0%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4321 20.0%	100% 0.0%
			100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
		Target Class					

Figura 69. Matriz de confusión del entrenamiento del bosque con 200 árboles.

Primero se llevará a cabo la validación para un tamaño de bosque de 200 árboles. En la validación se incluyen los 5 fallos con aumento de magnitud de forma simultánea para poder obtener resultados a partir de la matriz de confusión (Figura 70). El porcentaje global de datos clasificados es del 63,4%. Es un valor inferior comparado con los resultados obtenidos con el resto de modelos. Sin embargo, variando el número de árboles que componen el bosque, este resultado puede cambiar.

Output Class	1	2	3	4	5	
1	4318 20.0%	653 3.0%	209 1.0%	126 0.6%	3449 16.0%	49.3% 50.7%
2	0 0.0%	2839 13.1%	1373 6.4%	0 0.0%	427 2.0%	61.2% 38.8%
3	2 0.0%	58 0.3%	2350 10.9%	0 0.0%	22 0.1%	96.6% 3.4%
4	1 0.0%	771 3.6%	389 1.8%	4194 19.4%	423 2.0%	72.6% 27.4%
5	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0.0% 100%
	99.9% 0.1%	65.7% 34.3%	54.4% 45.6%	97.1% 2.9%	0.0% 100%	63.4% 36.6%
	1	2	3	4	5	
	Target Class					

Figura 70. Matriz de confusión para bosque de 200 árboles.

Si aumentamos el tamaño del bosque a 600 árboles, el porcentaje global de datos clasificados es de 64%. Con un bosque 3 veces más grande únicamente se ha conseguido un aumento del 0,7%.

Output Class	1	2	3	4	5	
1	4309 19.9%	642 3.0%	165 0.8%	99 0.5%	3377 15.6%	50.2% 49.8%
2	0 0.0%	2873 13.3%	1342 6.2%	0 0.0%	377 1.7%	62.6% 37.4%
3	2 0.0%	56 0.3%	2415 11.2%	0 0.0%	52 0.2%	95.6% 4.4%
4	10 0.0%	750 3.5%	399 1.8%	4221 19.5%	515 2.4%	71.6% 28.4%
5	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0.0% 100%
	99.7% 0.3%	66.5% 33.5%	55.9% 44.1%	97.7% 2.3%	0.0% 100%	64.0% 36.0%
	1	2	3	4	5	
	Target Class					

Figura 71. Matriz de confusión para bosque de 600 árboles.

Por último, se ha aumentado el número de árboles en el entramiento, duplicando el bosque anterior. Con 1200 árboles el porcentaje global de datos

clasificados es de 65,7%. La clasificación mejora suavemente con aumentos grandes del tamaño del bosque, lo que significa que el esfuerzo computacional aumenta drásticamente para mejoras en la clasificación pequeñas.

Output Class	1	2	3	4	5	
1	4319 20.0%	619 2.9%	188 0.9%	98 0.5%	3462 16.0%	49.7% 50.3%
2	0 0.0%	2844 13.2%	922 4.3%	0 0.0%	378 1.7%	68.6% 31.4%
3	2 0.0%	60 0.3%	2811 13.0%	0 0.0%	36 0.2%	96.6% 3.4%
4	0 0.0%	798 3.7%	400 1.9%	4222 19.5%	445 2.1%	72.0% 28.0%
5	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0.0% 100%
	100.0% 0.0%	65.8% 34.2%	65.1% 34.9%	97.7% 2.3%	0.0% 100%	65.7% 34.3%
	1	2	3	4	5	
	Target Class					

Figura 72. Matriz de confusión para bosque de 1200 árboles.

El hecho de que el porcentaje a penas aumente se debe a que ninguno de los 3 bosques validados consigue clasificar el fallo en qwwtp, por lo que el porcentaje máximo que puede alcanzar es del 80%. Queda demostrado que un aumento en el número de árboles no alcanza a conseguir clasificar el fallo en el caudal de la depuradora, pues el porcentaje de clasificación es del 0%. Por lo tanto, no se puede considerar una mejora de ninguno de los dos modelos anteriores.



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES



CAPÍTULO V: CONCLUSIONES Y TRABAJO FUTURO



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES

5.1. CONCLUSIONES

La irrupción de la Industria 4.0 ha facilitado la monitorización de procesos y con ello la obtención de grandes cantidades de datos, que coincidiendo con el auge del BigData, hace posible un tratamiento y análisis de estos grupos inmensos de datos a través de técnicas específicas para el procesamiento de datos. En este trabajo se ha pretendido hacer uso de estas técnicas para detectar y clasificar fallos a través de datos provenientes de la monitorización de un proceso, concretamente de un colector de aguas residuales.

Para la detección de fallos, se han empleado 3 modelos distintos, basados los 3 en la misma técnica: el Análisis de Componentes Principales. El primer modelo consistía en una aplicación lineal de la técnica, y los resultados con umbral adaptativo han sido muy positivos. Todos los fallos han sido detectados, y 9/10 de manera instantánea por alguno de los estadísticos. El segundo modelo estaba basado en una aplicación no lineal de PCA en el que se hacía uso de redes neuronales. A pesar de la complejidad de este método, los resultados han sido negativos, pues con umbral constante tan sólo un fallo ha sido detectado, mientras que con umbral adaptativo los fallos eran detectados antes siquiera de que el fallo se provocase, lo que indica que el modelo está fallido. El tercer modelo se centró en el residuo de los datos de fallo, con el objetivo de amplificar la señal de detección. Tras aplicar PCA al residuo, se obtuvieron resultados muy positivos y que pueden desmarcar a este método como el mejor de los 3 desarrollados. 10/10 fallos fueron detectados de forma instantánea, y el porcentaje de fallo a lo largo del tiempo alcanzó en todos los fallos cotas muy altas (en torno al 80%). El porcentaje de falsas alarmas fue muy bajo en todos los fallos, por lo que este modelo se ha considerado como el mejor de los 3 desarrollados, siendo el modelo del PCA no lineal el peor en la detección.

Tras la detección de un fallo, identificar la anomalía supone poder encontrar el origen y solucionar el problema en poco tiempo. Para la clasificación se han desarrollado de nuevo otros 3 modelos diferentes. El primer modelo estaba



basado en redes neuronales monocapa (machine learning), y sus resultados en la clasificación no fueron malos. El modelo clasificó 4 de los 5 fallos con un porcentaje mayor del 75%, pero sin embargo hubo un fallo que no se logró clasificar. El funcionamiento con los datos de los 5 fallos de manera conjunta fue positivo teniendo en cuenta que uno de los cinco a penas se clasificó, con un porcentaje global de datos clasificados del 73.2%. El segundo modelo para la clasificación estaba basado en redes neuronales multicapa (Deep Learning). A pesar de que este método tuvo porcentajes de clasificación más bajos en algún fallo, sí que consiguió clasificar los 5 fallos por encima del 50%, lo que coloca este modelo por encima del anterior. Con grandes cantidades de datos el modelo obtuvo un porcentaje de clasificación global de 77,8%, superior al modelo anterior precisamente por la capacidad de clasificar los 5 fallos. El tercer y último modelo se basó en los árboles de decisión y los bosques aleatorios. Tras probar con 200, 600 y 1200 árboles, tan sólo se pudo llegar a un porcentaje de clasificación global del 65,7%. Además, este modelo tuvo los peores resultados en la clasificación del fallo en el caudal de la depuradora (qwwtp), siendo incapaz de clasificarlo. El modelo Deep-Learning se considera por lo tanto el mejor en la clasificación de fallos, siendo el modelo basado en árboles de decisión el peor por su baja clasificación.

5.2. TRABAJO FUTURO

Los trabajos futuros que sigan la línea de lo desarrollado en este trabajo podrían centrarse en el desarrollo de un modelo basado en redes neuronales multicapa con distintas arquitecturas y tamaño de capas ocultas, y en encontrar una disposición de neuronas y un entrenamiento que consiga una detección de fallos efectiva, evitando el sobreajuste, utilizando otro tipo de redes dentro del área del Deep-learning, como redes convolucionales, redes recurrentes LSTM, etc. Para la clasificación podría desarrollarse un método que tuviera en cuenta una gran cantidad de clases de fallos para su clasificación, y poder obtener información sobre el comportamiento del modelo con enormes cantidades de datos, como lo que se ha desarrollado pero utilizando algunos



paradigmas nuevos de Deep-Learning. Por otro lado, sería interesante desarrollar métodos capaces de identificar, separar y clasificar fallos que se produzcan al mismo tiempo en un proceso productivo. Otra idea sería trabajar con más datos de fallos, es decir, provocar otros fallos distintos en la planta y observar si se pueden detectar y/o clasificar.



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES



REFERENCIAS

- [1] Gestión de producción y operaciones: ejemplos, sistema y etapas. (s. f.). www.beetrack.com. Recuperado 1 de junio de 2022, de <https://www.beetrack.com/es/blog/gestion-de-produccion-y-operaciones>
- [2] CARBONERAS CHECA, M. A. & BESTRATÉN BELLOVÍ, M. *Integración de sistemas de gestión: prevención de riesgos laborales, calidad y MA*. Ministerio de Trabajo y Asuntos Sociales, España, 2008.
- [3] AENOR (2009). *Sistemas de gestión de la calidad. Requisitos*. UNE-EN ISO 9001:2015 Madrid: AENOR.
- [4] GONZÁLEZ VELÁZQUEZ, M. (2020). *Mejora en la calidad de un proceso mediante la detección de anomalías basadas en datos*. Trabajo Fin de Grado. EII, Universidad de Valladolid.
- [5] PRAT BARTÉS, A., TORT-MARTONELL LLABRÉS, X. & GRIMA CINTAS, P. (2000). *Métodos estadísticos. Control y mejora de la calidad*. Barcelona: Universidad Politècnica Catalunya.
- [6] CARRO PAZ, R. & GONZÁLEZ GÓMEZ, D. (2012). *Control Estadístico de Procesos*. Buenos Aires: Universidad Nacional de Mar de la Plata.
- [7] L. (2021, 19 agosto). Los graficos de control de Shewhart: principios basicos. LeanSherpa. Recuperado 9 de junio de 2022, de <https://leansherpa.es/los-graficos-control-shewhart-principios-basicos/>
- [8] GOMEZ, A., & VILLOLDO, A. G. (s. f.). *Diagrama o gráfico de control: herramienta control de procesos. Manual de gestión de calidad paso a paso*. Recuperado 9 de junio de 2022, de <http://asesordecualidad.blogspot.com/2017/07/diagrama-o-grafico-de-control.html#.YqG25hrP1D8>

- [9] CUEVAS DÍAZ, A. (2012). *Aplicación de la técnica de componentes principales en el análisis del comportamiento mecánico dinámico funcional de las turbinas a vapor de 100MW*. Marianao, Cuba: Instituto Superior Politécnico José Antonio Echeverría.
- [10] D. GARCIA-ÁLVAREZ & M. FUENTE. *Análisis Comparativo de Técnicas de Detección de Fallos utilizando PCA*. Revista Iberoamericana de Automática e Informática Industrial RIAI, vol. 8, pp. 182-195, 2011.
- [11] KELLEHER, J. D. (2019). Deep learning. MIT press.
- [12] ABELIUK, A., & GUTIÉRREZ, C. (2021). Historia y evolución de la inteligencia artificial. Revista Bits de Ciencia, (21), 14-21.
- [13] ARISTIZÁBAL, M. C. (2006). Evaluación asimétrica de una red neuronal artificial: aplicación al caso de la inflación en Colombia. Lecturas de Economía, (65), 75-116.
- [14] X. (2022, 6 febrero). Redes Neuronales en Metatrader II. X-Trader.net - Trading en Estado Puro. Recuperado 15 de junio de 2022, de <https://www.x-trader.net/redes-neuronales-en-metatrader-ii/>
- [15] LÓPEZ FLÓREZ, R. & FERNÁNDEZ FERNÁNDEZ, J. M. (2008). *Las redes neuronales artificiales. Fundamentos teóricos y aplicaciones prácticas*. La Coruña, España: Gesbiblo, S. L.
- [16] KIM KIM, J. H. (2018). Aplicación de técnicas de aprendizaje no supervisado al modelado de ciclos cinemáticos urbanos. Proyecto fin de Grado, E.T.S.I. Industriales, Universidad Politécnica de Madrid.
- [17] PEQUEÑO ALONSO, A. (2020). *Mejora del control de calidad de un proceso mediante técnicas de aprendizaje automático*. Trabajo Fin de Grado. Valladolid: Universidad de Valladolid
- [18] SKANSI, S. (2018). Introduction to Deep Learning: from logical calculus to artificial intelligence. Springer.



- [19] BERLANGA, V., RUBIO HURTADO, M. J., & VILÀ BAÑOS, R. (2013). Cómo aplicar árboles de decisión en SPSS. REIRE. Revista d'Innovació i Recerca en Educació, 2013, vol. 6, num. 1, p. 65-79.
- [20] MARTÍNEZ, R. E. B., RAMÍREZ, N. C., MESA, H. G. A., SUÁREZ, I. R., TREJO, M. D. C. G., LEÓN, P. P., & MORALES, S. L. B. (2009). Árboles de decisión como herramienta en el diagnóstico médico. Revista médica de la Universidad Veracruzana, 9(2), 19-24.
- [22] SAAGI, R., FLORES ALSINA, X., FU, G., BUTLER, D., GERNAEY, K. V., & JEPPSSON, U. (2016). *Catchment & sewer network simulation model to benchmark control strategies within urban wastewater systems. Environmental Modelling & Software*, 78, 16-30. <https://doi.org/10.1016/j.envsoft.2015.12.013>
- [21] CEMPELLÍN, A., FRANCISCO, M., & VEGA, P. (2020). *Distributed Model Predictive Control Applied to a Sewer System. Processes*, 8(12), 1595. <https://doi.org/10.3390/pr8121595>