



Universidad de Valladolid



ESCUELA DE INGENIERÍAS  
INDUSTRIALES

UNIVERSIDAD DE VALLADOLID  
ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería en Tecnologías Industriales

# Monitorización distribuida de un proceso para mejora de su calidad mediante técnicas de inteligencia computacional /deep learning

**Autor:**

**Domínguez González, Juan**

**Tutores:**

**De La Fuente Aparicio,  
María Jesús**

**Departamento de Ingeniería  
de Sistemas y Automática**

# TABLA DE CONTENIDO

ÍNDICE DE FIGURAS.....	4
ÍNDICE DE TABLAS .....	6
RESUMEN .....	7
ABSTRACT.....	8
1. CAPÍTULO I: INTRODUCCIÓN Y OBJETIVOS .....	9
1.1.    Introducción.....	10
1.2.    OBJETIVOS .....	11
1.3.    ESTRUCTURA DE MEMORIA .....	12
2. CAPÍTULO II: ESTUDIO TEÓRICO.....	14
2.1.    CONTROL DE CALIDAD.....	15
2.2.    DETECCIÓN DE FALLOS POR CONTROL ESTADÍSTICO DE PROCESOS .....	18
2.2.1.    VARIABILIDAD EN LOS PROCESOS DE PRODUCCIÓN.....	19
2.2.2.    CAUSAS DE VARIABILIDAD .....	21
2.2.3.    GRAFICAS DE CONTROL .....	22
2.2.4.    ANÁLISIS DE COMPONENTES PRINCIPALES (PCA).....	26
2.2.5.    ESTADÍSTICOS.....	30
2.3.    DETECCIÓN DE FALLOS CON TÉCNICAS DE MACHINE LEARNING Y DEEP LEARNING.....	32
2.3.1.    REDES NEURONALES ARTIFICIALES.....	35
2.3.2.    REDES NEURONALES DE PROPAGACIÓN HACIA DELANTE .....	38
2.3.3.    AUTOENCODERS .....	39
2.3.4.    AUTOENCODERS DISTRIBUIDOS.....	42
2.3.5.    ÍNDICE DE INFERENCIA BAYESIANA (BIC).....	44
3. CAPÍTULO III: PLANTA TENNESSEE EASTMAN .....	47
3.1.    DESCRIPCIÓN DE LA PLANTA.....	48
3.2.    DATOS DE LA PLANTA.....	50
4. CAPÍTULO IV: APLICACIÓN .....	54
4.1.    INTRODUCCIÓN A LA APLICACIÓN.....	55
4.2.    ANÁLISIS DE COMPONENTES PRINCIPALES .....	56
4.3.    AUTOENCODER GENERAL.....	62

*Juan Domínguez González (2024) - EII - Universidad de Valladolid*

4.4.	AUTOENCODER DISTRIBUIDO.....	67
4.4.1.	Umbral límite de relevancia igual a la media.....	68
4.4.2.	Umbral igual a la media de las relevancias ampliada un 50% .....	76
4.4.3.	Umbral igual a la media de las relevancias reducida un 50% .....	81
5.	CAPÍTULO V: CONCLUSIONES Y TRABAJO FUTURO.....	86
5.1.	CONCLUSIONES.....	87
5.2.	TRABAJO FUTURO.....	90
	BIBLIOGRAFÍA.....	92

# ÍNDICE DE FIGURAS

Figura 1: Ciclo PDCA [Elaboración propia] .....	16
Figura 2: Áreas responsables de la calidad [4] .....	17
Figura 3: Variabilidad del proceso. (a) Toma de muestras. (b) Distribuciones. (c) Variación de distribuciones. (d) Distribución con causas comunes de variación. (e) Distribución con causas especiales de variación [6] .....	20
Figura 4: Tipos de salidas del proceso. (a) Proceso fuera de control. (b) Proceso bajo control con causas comunes de variación fuera de los límites de especificación. (c) Proceso bajo control con causas de variación dentro de los límites de especificación. [6] .....	21
Figura 5: Relación de los límites de control con la distribución de muestreo [6] .....	23
Figura 6: Graficas de Control que presentan situaciones que pueden preceder a estados fuera de control estadísticos.[7] .....	24
Figura 7: Grafica de Control según Shewhart [Elaboración propia] .....	25
Figura 8: Interpretación Geométrica de PCA. Ejes de elipsoide [10] .....	27
Figura 9: Aprendizaje supervisado. A la izquierda representación de grafico de regresión lineal y a la derecha grafico de clasificación lineal [12] .....	33
Figura 10: Esquema de Neurona Artificial [13] .....	36
Figura 11: Funciones de activación. A la izquierda escalón y a la derecha tangente hiperbólica [13]. .....	37
Figura 12: Esquema de neurona con inclinación [13]. .....	37
Figura 13: Red de propagación hacia adelante multicapa [16] .....	39
Figura 14: Estructura de Autoencoder [18] .....	41
Figura 15: Diagrama de proceso de la planta TE [21] .....	49
Figura 16: Representación de los valores de los estadísticos partiendo de un modelo sin fallos con estudio mediante PCA .....	57
Figura 17: Representación de estadísticos para fallos IDV (7) y IDV (17) estudiados mediante PCA .....	59
Figura 18: Representación de los valores de los estadísticos partiendo de un modelo sin fallos con estudio mediante Autoencoder general .....	63
Figura 19: Representación de estadísticos para fallos IDV (7) y IDV (17) estudiados mediante Autoencoder general .....	65
Figura 20: Representación de los valores de los estadísticos partiendo de un modelo sin fallos para el estudio de cada uno de los bloques de un Autoencoder Distribuido .....	69
Figura 21: Representación de estadísticos para fallos IDV (7) y IDV (17) estudiados mediante cada red del Autoencoder Distribuido .....	71

*Juan Domínguez González (2024) - EII - Universidad de Valladolid*

Figura 22: Representación de estadísticos BIC para fallos IDV (7) e IDV (17) estudiados mediante Autoencoder distribuido cuando el umbral límite de relevancia es igual a la media .....	74
Figura 23: Representación de estadísticos BIC para fallos IDV (7) e IDV (17) estudiados mediante Autoencoder distribuido cuando el umbral es igual a la media de las relevancias ampliada un 50%.....	79
Figura 24: Representación de estadísticos BIC para fallos IDV (7) e IDV (17) estudiados mediante Autoencoder distribuido cuando el umbral es igual a la media de las relevancias reducida un 50%.....	83

# ÍNDICE DE TABLAS

Tabla 1: Variables manipuladas del proceso [21].....	50
Tabla 2: Variables medidas del proceso [3].....	51
Tabla 3: Fallos posibles en el proceso [22].....	52
Tabla 4: Parámetros estadísticos obtenidos para los 21 errores utilizando el método de PCA.....	61
Tabla 5: Parámetros estadísticos obtenidos para los 21 errores utilizando el método de Autoencoder general .....	66
Tabla 6: Número de alarmas para el estadístico Q, analizando los 21 errores en cada uno de los bloques del autoencoder distribuido cuando el umbral límite de relevancia es igual a la media.....	73
Tabla 7: Parámetros estadísticos obtenidos para los 21 errores utilizando el método de Autoencoder distribuido cuando el umbral límite de relevancia es igual a la media .....	75
Tabla 8: Número de alarmas para el estadístico Q, analizando los 21 errores en cada uno de los bloques del autoencoder distribuido cuando el umbral es igual a la media de las relevancias ampliada un 50%.....	78
Tabla 9: Parámetros estadísticos obtenidos para los 21 errores utilizando el método de Autoencoder distribuido cuando el umbral es igual a la media de las relevancias ampliada un 50%.....	80
Tabla 10: Número de alarmas para el estadístico Q, analizando los 21 errores en cada uno de los bloques del autoencoder distribuido cuando el umbral es igual a la media de las relevancias reducida un 50% .....	82
Tabla 11: Parámetros estadísticos obtenidos para los 21 errores utilizando el método de Autoencoder distribuido cuando el umbral es igual a la media de las relevancias reducida un 50% .....	84
Tabla 12: Valores medios excluyendo los fallos IDV (3), IDV (9) e IDV (15), para cada método estudiado.....	87

# RESUMEN

Este trabajo tiene como objetivo mejorar la calidad de un proceso en una planta industrial mediante distintas técnicas de detección y diagnóstico de fallos (FDD) basadas en datos. Esto es debido a que, con la cuarta revolución industrial, también llamada Industria 4.0, se ha dado paso a la automatización y digitalización de los procesos industriales utilizando las últimas tecnologías. Como consecuencia, surge la recogida masiva de datos a lo largo de dichos procesos, utilizando sensores que nos permitan la lectura de las variables deseadas.

En este trabajo, por tanto, se pretende estudiar diferentes métodos que permitan el tratamiento de los datos recogidos de un proceso industrial y poder detectar los fallos o anomalías que puedan suceder. Por un lado, se estudiará un método clásico de control estadístico multivariante como es el Análisis de Componentes Principales (PCA). Además, debido al creciente auge de la inteligencia artificial y el “deep-learning” o aprendizaje profundo, usaremos un método de FDD basado en redes neuronales de propagación hacia adelante, y en concreto en una red llamada Autoencoder. En este método, se usará de dos formas distintas, mediante un autoencoder general con todas las variables del proceso, y en un autoencoder distribuido, donde mediante un método basado en datos se dividen las variables del proceso en bloques y se implementa un autoencoder para detectar fallos en cada bloque, finalmente las salidas de cada bloque se fusionan para obtener un resultado final.

Los datos que se van a utilizar para la aplicación de los métodos mencionados son obtenidos de la planta química Tennessee Eastman, ya que es un proceso muy utilizado por la comunidad científica como banco de pruebas, para el control y la detección de fallos.

Finalmente se procederá a realizar una comparación entre los diferentes métodos estudiados, sacándose las conclusiones oportunas, y se realizará un breve estudio de trabajo futuro donde se plantean mejoras u otras líneas de acción asociadas a este estudio.

## Palabras Clave

Control de calidad, PCA, Autoencoder, Autoencoder Distribuido, Estadístico T<sup>2</sup>, Estadístico Q, Tennessee Eastman

# ABSTRACT

This work aims to improve the quality of a process in an industrial plant through various fault detection and diagnosis (FDD) techniques based on data. This is due to the fact that, with the fourth industrial revolution, also called Industry 4.0, there has been a shift towards the automation and digitalization of industrial processes using the latest technologies. As a consequence, there is massive data collection throughout these processes, using sensors that allow us to read the desired variables.

Therefore, this work aims to study different methods that allow the processing of data collected from an industrial process and to detect faults or anomalies that may occur. On the one hand, a classical method of multivariate statistical control, Principal Component Analysis (PCA), will be studied. Additionally, due to the increasing rise of artificial intelligence and deep learning, we will use an FDD method based on feedforward neural networks, more specifically a network called Autoencoder. This method will be used in two different ways: through a general autoencoder with all the process variables and a distributed autoencoder, where the process variables are divided into blocks using a data-based method, and an autoencoder is implemented to detect faults in each block. Finally, the outputs of each block are merged to obtain a final result.

The data to be used for the application of the mentioned methods are obtained from the Tennessee Eastman chemical plant, as it is a process widely used by the scientific community as a benchmark for control and fault detection.

Finally, a comparison will be made between the different methods, drawing appropriate conclusions, and a brief study of future work will be conducted where improvements or other lines of action associated with this study are proposed.

## Key Words

Quality control, PCA, Autoencoder, Distributed Autoencoder,  $T^2$  statistics, Q statistics, Tennessee Eastman,





# CAPÍTULO I: INTRODUCCIÓN Y OBJETIVOS

## 1.1. INTRODUCCIÓN

El concepto de control y mejora de la calidad lleva existiendo desde la antigüedad, los primeros homínidos ya buscaban la forma de mejorar sus tareas para hacerlas más efectivas. Sin saber que esa acción que ellos realizaban casi por inercia y sin pensarlo en exceso, sería uno de los grandes objetos de estudio de la actualidad.

Durante un largo periodo de tiempo se continuó realizando controles de calidad sin denominarlos de esta manera, hasta comienzos del siglo XX. Fue en este momento cuando se empezó a desarrollar el termino de Calidad propiamente dicho y se escribieron los primeros textos que trataban su control y mejora.

Hoy en día la calidad ha pasado a formar parte del sistema productivo, convirtiéndose en un eslabón más de la cadena. Este concepto no se cierra únicamente al ámbito industrial, sino que se aplica a todos los aspectos de la vida diaria. Por este motivo, se han desarrollado, y siguen en desarrollando, múltiples formas de estudio de la calidad. Cada una aplicada al ámbito que se trate. Es muy distinto un control de calidad de la producción de un elemento físico, frente a un control de calidad de un equipo humano de trabajo en I+D, por ejemplo.

De esta manera se puede observar que una vez conocido y comprendido el concepto de calidad es necesario una elevada especificación en el ámbito de trabajo para realizar un buen control de calidad.

De igual manera, siendo la calidad un elemento de estudio exhaustivo en la actualidad, los métodos y formas de realización de un control de calidad en un mismo ámbito pueden verse modificados o mejorados en cortos periodos de tiempo. De ahí que sea un elemento puntero en el trabajo de cualquier empresa, ya que cuenta con una rápida evolución.

Estos conceptos sitúan el estudio realizado en su espacio y tiempo. Particularizando para un ámbito de trabajo a nivel de planta industrial, el control calidad pretende asegurar que las especificaciones de un producto final se mantienen en un rango aceptable, asegurando su calidad, además, también se monitoriza el proceso en su conjunto para su correcto funcionamiento, para detectar anomalías que pueden suponer un riesgo para el propio proceso, y para los agentes externos que intervienen en él, y así evitar posibles daños humanos, materiales y/o medioambientales.

Por tanto, asegurar la calidad, así como la seguridad en la producción son dos objetivos de cualquier industria moderna, lo que se consigue mediante la implementación de métodos de detección y diagnóstico de fallos que detecten cualquier anomalía que aparezca en la planta. Este problema ha sido abordado desde diferentes puntos de vista, los métodos analíticos o basados en modelos, los métodos basados en datos y los métodos basados en el conocimiento. En el presente proyecto el estudio se basa únicamente en el estudio de magnitudes físicas, tales como dimensiones, presiones, temperaturas, etcétera. Variables disponibles en toda industria moderna debida a la implantación de las nuevas tecnologías digitales. De ahí que el proceso de control de calidad en este trabajo se pueda basar completamente en un estudio de datos objetivos, tomados en diferentes momentos de trabajo de la planta en cuestión.

Para realizar este estudio de monitorización de procesos basados en datos existen diversas técnicas. Una de las más utilizadas se basa en el control estadístico de los procesos, en concreto el Análisis de Componentes Principales (PCA), que se basa en la extracción de información de un conjunto de datos al cual se le pretende reducir su dimensión, sin perder información en el proceso.

Sin embargo, el campo con más futuro en el tratamiento de datos industriales para la detección de fallos y por tanto para el control de calidad basado en datos son las técnicas de inteligencia artificial, más en concreto, las redes neuronales donde las más complejas son capaces de extraer patrones y de procesar gran cantidad de datos complejos, también llamadas “deep-learning” o aprendizaje profundo, y entre estas técnicas se encuentran los autoencoders que son los que utilizaremos en este trabajo.

## 1.2. OBJETIVOS

Se conoce así el objetivo principal del estudio en cuestión, siendo este el de obtención de métodos relativamente perfeccionados para la detección de fallos en las variables de la planta de estudio. Para alcanzar este objetivo se han usado técnicas basadas en el control estadístico de procesos, como el Análisis de Componentes Principales y técnicas basadas en deep-learning como los Autoencoders.

Además, como las plantas industriales son complejas, se tratará de aportar también información sobre puntos o características más críticas de la planta, implementando métodos distribuidos que se centran en diferentes partes de la planta, ya que no todas las variables causan el mismo tipo de fallos.

Posteriormente, se contrastarán los resultados con las diferentes técnicas, identificando cuál de los métodos resulta más eficaz para detectar anomalías en una instalación industrial.

Los datos usados provienen de la planta Tennessee Eastman, utilizada como caso de estudio y banco de datos en la literatura científica permitiendo el desarrollar y validar técnicas en un entorno complejo y realista.

### 1.3. ESTRUCTURA DE MEMORIA

De esta manera la memoria se estructura comenzando por un estudio teórico en el que se incluye una síntesis de la información esencial que se ha tenido en cuenta por parte del redactor de este documento, para la realización del estudio en cuestión. Considerándose este apartado algo esencial para la propia realización del proyecto.

Continuado al estudio teórico, se encuentra un apartado de definición de la planta de trabajo. Se trata de la planta química Tennessee Eastman, comúnmente tomada como base para estudios de este estilo. En él se define de manera general el proceso realizado en dicha planta y los principales datos de estudio, tratados como variables del proceso.

El siguiente y más extenso capítulo de la memoria, trata en detalle los trabajos realizados en lenguaje de programación Python, para la creación de las técnicas de control de calidad. Dentro de este apartado se diferencian tres tipos de técnicas, uno más básico e inicial a través de PCA, y dos siguientes que basan su funcionamiento en el tipo de redes neuronales conocidas como Autoencoders. Diferenciándose entre sí en función de la parcialización realizada en el estudio de las diferentes variables. El primer Autoencoder trabaja de manera general con todas las variables del proceso y el segundo modelo, genera Autoencoders más específicos para el estudio de un número más reducido de variables, es decir, se implementa un método de detección de fallos distribuido.

Tras el capítulo de aplicación del trabajo se puede encontrar una sección definida para la localización de las principales conclusiones generadas a partir del estudio realizado. A mayores, animado por las conclusiones obtenidas sobre los modelos realizados, se plantean posibles modificaciones, cambios o mejoras en concepto de trabajo futuro. Gracias al cual se podrían sacar unas mejores conclusiones en el caso de que alguna idea planteada no haya quedado resuelta de la manera deseada.



*Juan Domínguez González (2024) - EII - Universidad de Valladolid*

Finalmente se encuentra un capítulo de referencias a los textos que han servido de base para la obtención de toda la información que se ha tenido en cuenta para la realización de dicha memoria.



# CAPÍTULO II: ESTUDIO TEÓRICO

## 2.1. CONTROL DE CALIDAD

La historia del control de calidad es más antigua que la propia industria. Desde el origen de las sociedades, la humanidad seleccionaba y perfeccionaba sus métodos de vida. Este proceso existe desde que el hombre primitivo trabajaba su vivienda y herramientas, pasando por las civilizaciones antiguas como la egípcia que cuidaban mucho los cálculos matemáticos en la construcción de las pirámides, hasta la edad media donde los gremios insistían en obtener grandes periodos de entrenamiento para mejorar la calidad de sus artesanías.

El termino propiamente dicho de control de calidad, no se acuñó hasta el siglo XX con la evolución de la industrialización. Con ella llegó el desarrollo de la producción en masa, la especialización, el incremento de la complejidad de los procesos productivos y la necesidad de optimización de recursos, para la reducción de costes. Conceptos para los cuales era necesario un aumento de la eficiencia en las líneas de producción.

Este trabajo se vio influenciado por el aumento del uso y desarrollo de la tecnología, como elemento base para la creación de métodos de supervisión más específicos. [1]

La creación de estos métodos surgió a raíz de las demandas de la industria de armamentos, que necesitaba producir un gran volumen de componentes, impulsando la implementación de estándares. Esto marcó el inicio de la instauración de normativas militares en los Estados Unidos conocidas como Z-1, así como los Estándares Británicos. Más tarde, se establecieron normativas en distintos ámbitos como la ingeniería, construcción e industria química.

Se incorporaron sistemas de medición (dispositivos de medición) y la aplicación de técnicas estadísticas en las labores de inspección y control. El objetivo era reducir los costos de inspección al limitarla a un número representativo de muestras del producto.

El aporte fundamental para desarrollar estas técnicas provino de Shewhart, quien aplicó principios estadísticos a los problemas de calidad, definiendo el concepto de variabilidad y, por ende, el de tolerancias. Shewhart también introdujo los gráficos de control para comprender la variabilidad y las causas asignables. Estos gráficos se empleaban en cada etapa del proceso, permitiendo una respuesta ágil a los cambios en el comportamiento del proceso (causas asignables) [2]

Fue Edwards Deming quien popularizó en la segunda mitad del siglo XX uno de los ciclos inicialmente desarrollado por Shewhart. Se trata del Ciclo PDCA

(Plan, Do, Check, Act) utilizado ampliamente en las áreas de gestión de calidad de las industrias actuales (Figura 1).

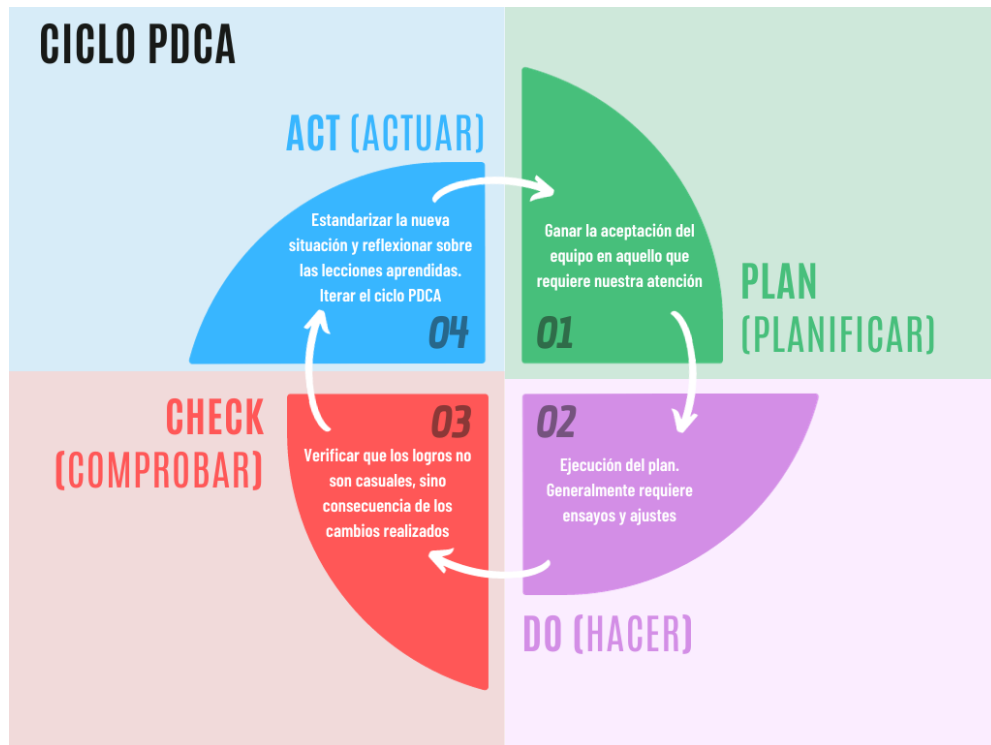


Figura 1: Ciclo PDCA [Elaboración propia]

Posteriormente, en los años 90, se incrementó la calidad y la productividad, siendo los países que aplicaron las normas de calidad y sistemas de certificación los que se incorporaron en una mejor posición al siglo XXI. Se popularizó el uso de un conjunto de normas sobre calidad y gestión de la calidad, que han sido implantadas por la ISO (Organización Internacional de Normalización), formada por órganos de normalización de prácticamente todos los países del mundo.

En concreto se habla de la familia de la norma ISO 9000, cuyo objetivo se centra en especificar los requerimientos que las organizaciones implementan a fin de cumplir sus objetivos en términos de calidad, medio ambiente, prevención de riesgos laborales, seguridad y responsabilidad social. [3]

En la actualidad el control de calidad se ha generalizado, siendo ahora necesario que todos los departamentos que intervienen en la fabricación o distribución de un producto o servicio (Figura 2) sean participes de este proceso.



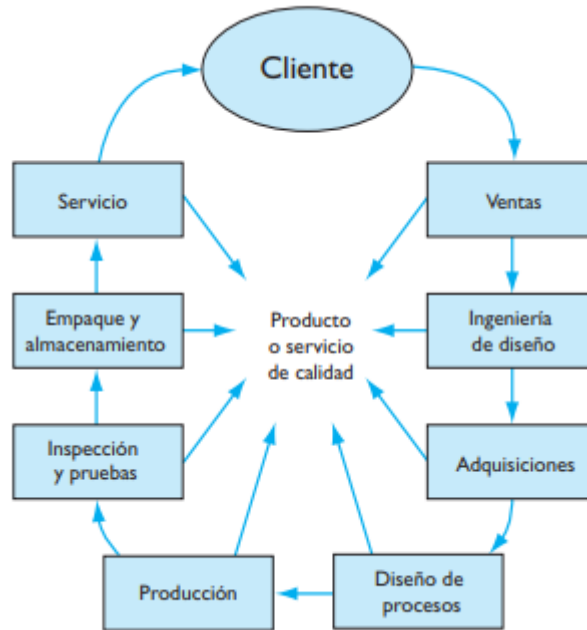


Figura 2: Áreas responsables de la calidad [4]

Para garantizar un control de calidad efectivo, se requiere contar con instrumentos de medida altamente precisos, un sistema de comunicación industrial sólido para recopilar y transmitir datos, la detección temprana de cualquier irregularidad y un diagnóstico adecuado de los errores.

La implementación de un esquema de detección y diagnóstico de fallos (FDD), integrado en la etapa de monitorización del proceso de control de calidad, contribuye a mejorar la fiabilidad de la planta.

El proceso de detección y diagnóstico de fallos (FDD) se divide en las siguientes etapas:

- Detección del fallo
- Identificación del fallo: buscar que variables de la planta son responsables del fallo
- Diagnóstico del fallo: identificar tipo y cause del fallo
- Recuperación del sistema: retornar la planta al funcionamiento normal [2]

Existen diversas técnicas disponibles para aplicar en la detección y diagnóstico de fallos (FDD). El Control Estadístico de Procesos (SPC) es una de las técnicas más comúnmente empleadas. Sin embargo, en la actualidad, está en auge la aplicación de técnicas de “Machine Learning” y “Deep

Learning” en plantas industriales modernas, ya que permiten manejar eficientemente la gran cantidad de datos generados en estos entornos. Estas técnicas posibilitan un análisis más avanzado y predictivo, contribuyendo así a la mejora continua y la optimización de los procesos industriales.

## 2.2. DETECCIÓN DE FALLOS POR CONTROL ESTADÍSTICO DE PROCESOS

El Control Estadístico de Procesos (SPC) se basa en la aplicación de técnicas estadísticas para comprobar y asegurar la calidad y repetibilidad de cualquier proceso productivo. Identificando los posibles problemas en una planta, que llevarán a las variaciones de un proceso que pueden derivar de dos tipos de causas, las asignables y las aleatorias.

El SPC fue desarrollado por Shewhart a comienzos del siglo XX (años 20), él planteó la posibilidad de identificación de los tipos de causas de las variaciones a partir de una variedad de herramientas; las principales que propone este método son:

- El histograma
- La plantilla de recogida de datos
- La grafica de Pareto
- El diagrama de causa y efecto
- Los diagramas bivariantes
- La Estratificación
- Los gráficos de control

El Control Estadístico de Procesos (SPC) cumple un rol esencial en la detección y manejo de anomalías en los procesos industriales. No solo se encarga de alertar al personal sobre estas anomalías, sino que también facilita la implementación de medidas y soluciones adecuadas para restablecer el proceso a su funcionamiento normal. Es un instrumento valioso para informar a la gerencia sobre cualquier cambio realizado en los procesos que pueda tener un impacto positivo en la producción resultante.

La necesidad de implementar el Control Estadístico de Procesos se fundamenta en las altas fuentes de variación presentes en cualquier proceso. Incluso cuando el proceso se desarrolla según lo previsto, no se obtendrán dos productos finales exactamente idénticos. Debido a esta variabilidad inherente en los procesos, destaca la importancia de monitorizar y controlar

de manera continua, para garantizar la consistencia y calidad del producto final. [5]

### 2.2.1. VARIABILIDAD EN LOS PROCESOS DE PRODUCCIÓN

Absolutamente todos los procesos industriales presentan variaciones, y es esencial comprender y distinguir entre dos tipos principales de variaciones: causas comunes de variación y causas especiales de variación. Cuando se dan únicamente las causas comunes el proceso sigue estando bajo control estadístico, mientras que cuando se suman las causas especiales, es cuando el proceso se encuentra fuera de control.

Las causas comunes de variación son fuentes normales de variabilidad que actúan de manera constante en un proceso. Estas causas generan variaciones dentro de un rango predecible y estable a lo largo del tiempo, dando lugar a una distribución estadística que es replicable y consistente. Por otro lado, las causas especiales de variación son perturbaciones inusuales concretas que por naturaleza son impredecibles e intermitentes.

Para evaluar la variabilidad del proceso, se toman pequeñas muestras a lo largo del tiempo y se representan en una gráfica donde el eje horizontal muestra la secuencia de las muestras y el eje vertical indica la frecuencia de cada muestra. Después de un número notable de muestras se le denomina distribución, si el patrón es estable, como se observa en la Figura 3 (b). Las distribuciones pueden diferir en función de lo que revelan las muestras como se observa en la Figura 3 (c).

Si el patrón resultante de estas muestras es estable y predecible, se dice que el proceso está bajo control estadístico (como se muestra en la Figura 3 [d]). Sin embargo, si aparecen causas especiales de variación, la salida del proceso se vuelve impredecible y no sigue un patrón estable a lo largo del tiempo (como se muestra en la Figura 3 [e]). Esta comprensión es fundamental para determinar la estabilidad y la predictibilidad del proceso, lo que a su vez guía la toma de decisiones para mejorar la calidad y eficiencia del proceso industrial.

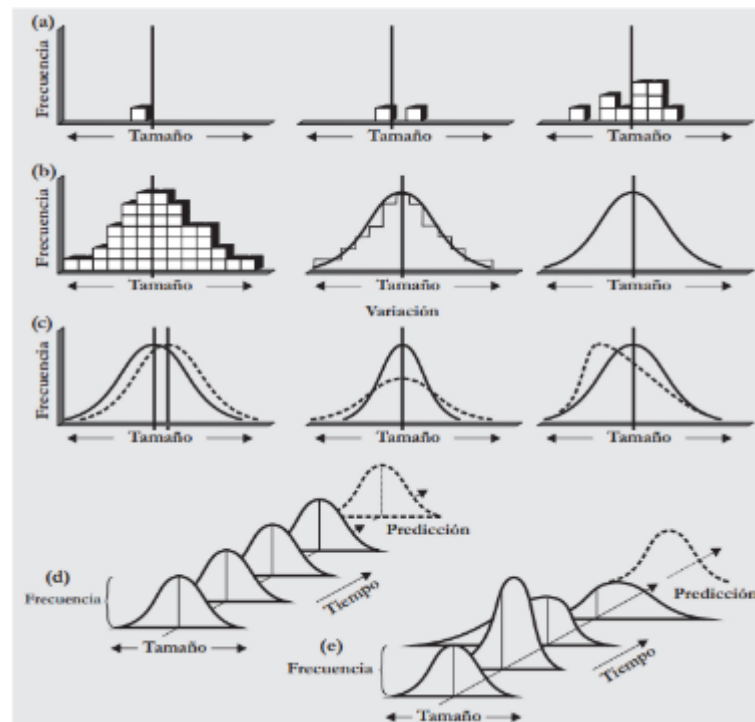


Figura 3: Variabilidad del proceso. (a) Toma de muestras. (b) Distribuciones. (c) Variación de distribuciones. (d) Distribución con causas comunes de variación. (e) Distribución con causas especiales de variación [6]

En la siguiente figura se muestran los tres tipos posibles de salidas del proceso, dependiendo del tipo de causas de variabilidad y su trabajo. En la Figura 4 (a) se muestra un proceso completamente fuera de control estadístico, debido a la existencia de causas especiales de variación (causas asignables). La Figura 4 (b) muestra un proceso bajo control estadístico ya que solo se comprenden causas comunes de variación, pero que se encuentra fuera de los límites de especificación. Mientras que la Figura 4 (c) centra las causas naturales únicamente y se encuentra dentro de los límites de calidad.

El objetivo de este tipo de control de calidad es la transformación progresiva de las salidas, hasta que nos encontremos en todo momento en una salida como se muestra en la Figura 4(c). Reduciendo todo lo posible o si fuera posible eliminando las causas especiales de variación.

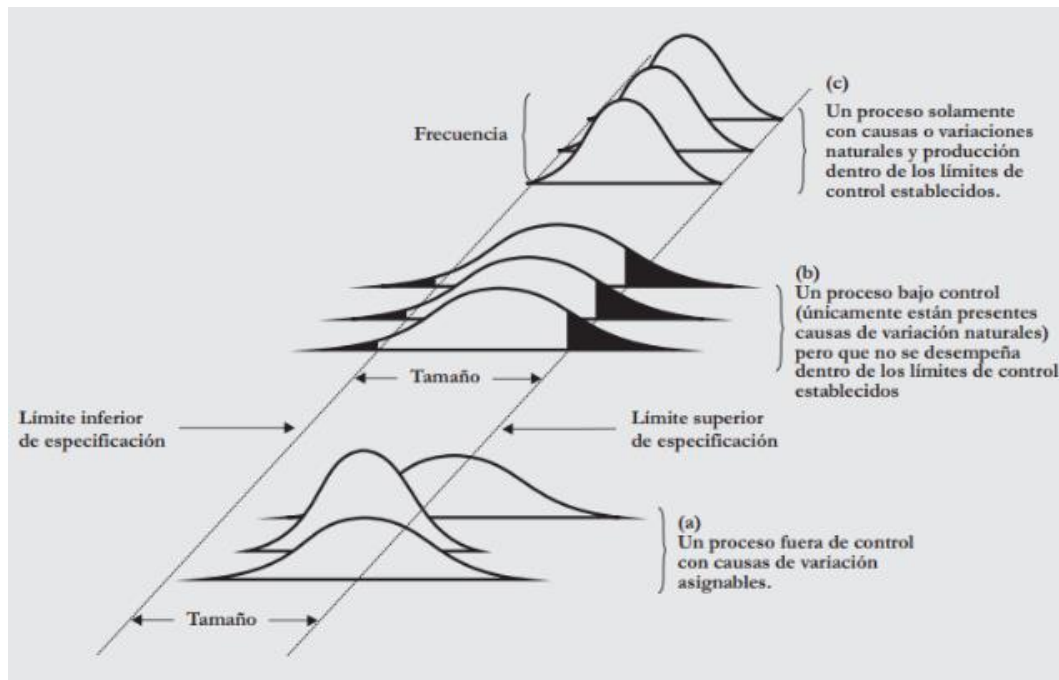


Figura 4: Tipos de salidas del proceso. (a) Proceso fuera de control. (b) Proceso bajo control con causas comunes de variación fuera de los límites de especificación. (c) Proceso bajo control con causas de variación dentro de los límites de especificación. [6]

## 2.2.2. CAUSAS DE VARIABILIDAD

Definida la variabilidad de un proceso productivo se puede profundizar en sus tipos de causas, existen dos tipos, las causas comunes de variación o causas naturales y las causas especiales de variación o causas asignables.

- Causas comunes de variación:

Se trata de elementos de variación completamente aleatorios, imposibles de identificar y evitar, aun así, no provocan una pérdida de control estadístico del proceso productivo.

- Causas especiales de variación:

Estas causas no son inherentes del proceso, y sí que provocan la pérdida de control estadístico de este, por lo tanto, deben ser identificadas y eliminadas, para poder devolver el proceso a su funcionamiento normalizado.

La existencia de este tipo de causas es la base del Control Estadístico de Procesos (SPC), ya que su función principal es la detección de causas especiales de variación en procesos que no se encuentran bajo control estadístico, y su eliminación, para devolverlos a su estado natural.

Ambas clases de causas forman distribuciones que se caracterizan por tres parámetros:

- La media: suma de valores de observaciones dividida entre el número total de observaciones.
- La varianza: medida de la dispersión de las observaciones en torno al valor de la media.
- La forma: se puede encontrar de dos formas, simétrica o asimétrica, respecto al valor central de la distribución (la media). En el caso en que sea simétrica el proceso tendrá el mismo peso a ambos lados de la media, es decir, presentará el mismo número de observaciones a cada lado. Mientras que, si no encontramos en un caso asimétrico, no se cumplirá este factor. [7]

### 2.2.3. GRAFICAS DE CONTROL

En el contexto del control estadístico de procesos, se emplean diversos tipos de gráficos para monitorizar y controlar la variabilidad del proceso. Estos gráficos se basan en distintas pautas de variabilidad y se interpretan de manera similar, considerándose el gráfico de control como una prueba de una hipótesis.

En términos generales, existen dos enfoques principales para la elaboración de gráficos de control en procesos industriales:

- Gráficos de Atributos:

En este tipo de gráficos, no se controlan las medidas directas del proceso, sino que se evalúa la fracción o porcentaje de productos defectuosos producidos. Se observa y registra si un artículo cumple o no con ciertas características o criterios de calidad, convirtiendo este enfoque en "atributos" que se pueden medir como defectuosos o no defectuosos.

- Gráficos de Dimensiones o Variables:

Estos gráficos se centran en el control de las medidas directas del proceso, como dimensiones físicas, pesos, temperaturas u otras variables cuantificables. Se registran y controlan las magnitudes numéricas para asegurar que se mantengan dentro de límites predefinidos.

Ambos tipos de gráficos desempeñan un papel crucial en la monitorización y control de la calidad en la producción industrial. La elección del tipo de gráfico a utilizar depende de la naturaleza y características específicas del proceso y de lo que se necesita controlar para garantizar la calidad del producto.[3]

Todo tipo de gráficos de control establecen límites, para su estudio. Existen límites superiores de control (LSC) y límites inferiores de control (LIC). En la mayoría de los casos también encontramos un valor que muestra la cadencia general de la distribución, conocido como línea central (LC).

Una distribución de muestras que se sitúa dentro de los límites establecidos señala que las variaciones en el proceso son normales y predecibles, conocidas como causas comunes de variación. Por otro lado, si la distribución se encuentra fuera de estos límites, indica que ha habido variaciones inusuales e impredecibles en el proceso, denominadas causas especiales de variación. La Figura 5 muestra la relación de los límites de control con las distribuciones.

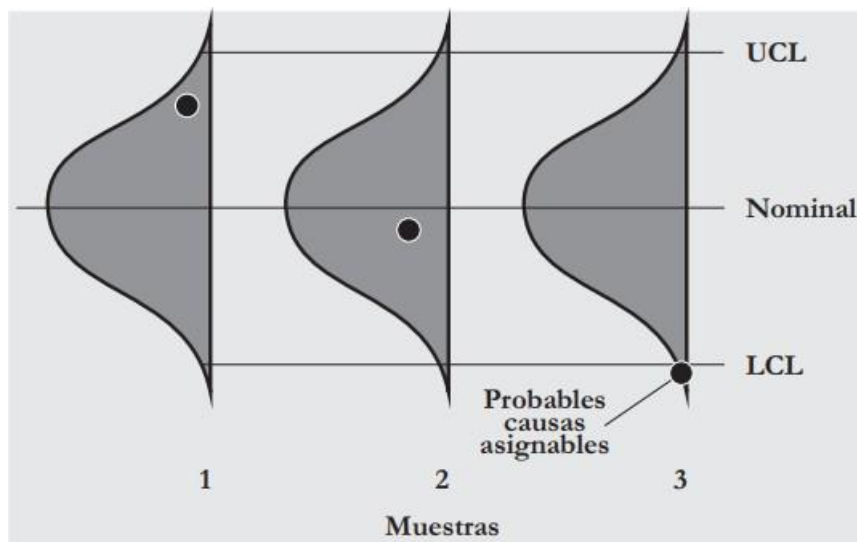


Figura 5: Relación de los límites de control con la distribución de muestreo [6]

Quienes supervisan un proceso suelen seguir este método utilizando gráficos de control:

1. Se toma una muestra al azar del proceso y se mide una característica relacionada con la calidad, calculando una medida que puede ser variable o de atributos.
2. Si la estadística se encuentra fuera de los límites establecidos en la gráfica de control, se investiga una causa específica para esa variación.
3. Si se identifica una causa y esta afecta negativamente la calidad, se elimina. Por otro lado, si se descubre que la causa puede mejorar la



Juan Domínguez González (2024) - EII - Universidad de Valladolid

calidad, se incorpora. Luego, se ajustan los datos y se actualiza la gráfica de control.

- Este proceso se repite de forma periódica para mantener una monitorización constante y efectivo del procedimiento. [7]

Existen dos posibles tipos de errores al utilizar este tipo de gráficos.

El error de 'Tipo I' (rechazo de un lote de alta calidad) ocurre al concluir que el proceso está descontrolado, basándose en un resultado de muestra que está fuera de los límites de control, cuando en realidad era simplemente un efecto completamente aleatorio.

El error de 'Tipo II' (aceptación de un lote de baja calidad) ocurre al concluir que el proceso está bajo control y solo muestra discrepancias aleatorias, cuando en realidad el proceso está fuera de control estadístico. [8]

A veces es posible detectar los problemas que afectan a un proceso, aun cuando los acotamientos de control no hayan sido rebasados. En la Figura 6 se presentan ejemplos de gráficas de control que muestran distintos comportamientos en los procesos que denotan situaciones anormales en el comportamiento de las muestras.

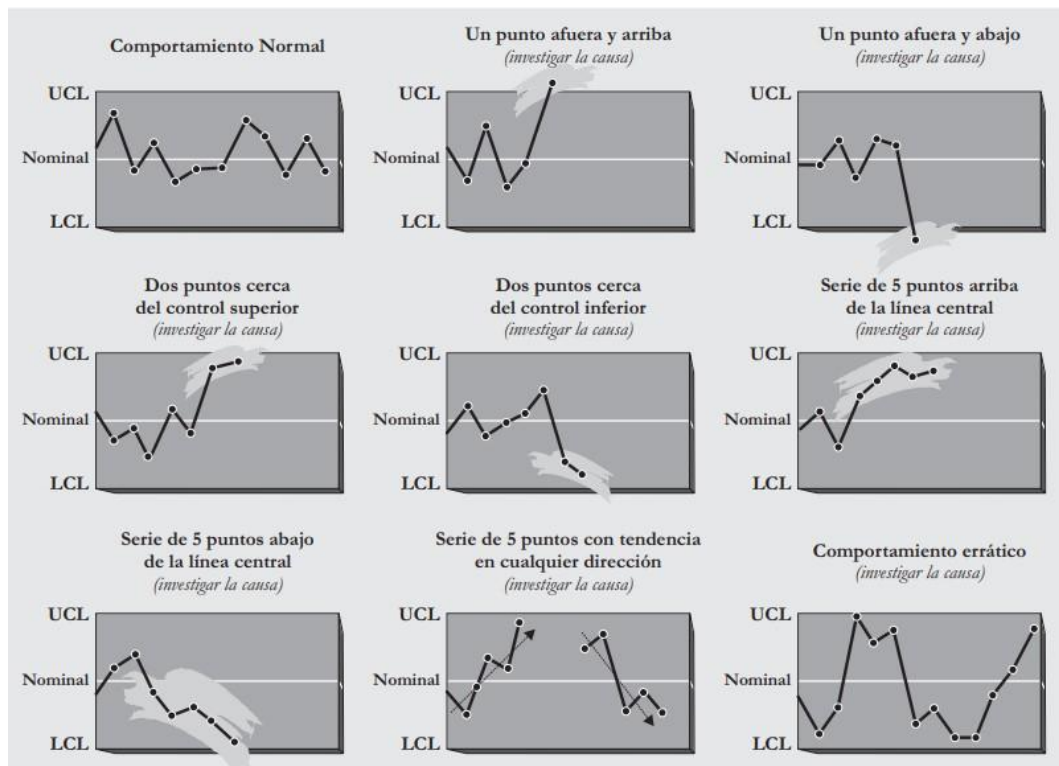


Figura 6: Gráficas de Control que presentan situaciones que pueden preceder a estados fuera de control estadísticos.[7]



Según Shewhart, la forma de una gráfica de control corresponde con la Figura 7 donde los límites de control y la línea central vienen definidas por las siguientes ecuaciones:

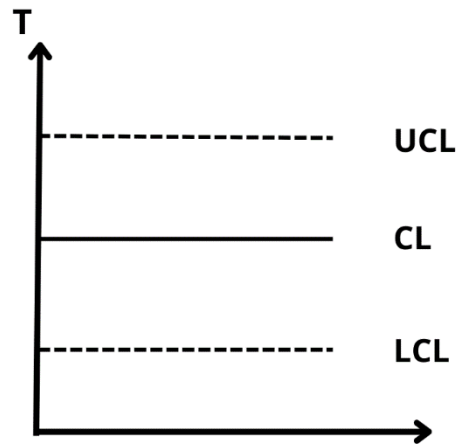


Figura 7: Gráfica de Control según Shewhart [Elaboración propia]

$$UCL = \mu_T + k \cdot \sigma_T \quad (Ec. 1)$$

$$CL = \mu_T \quad (Ec. 2)$$

$$LCL = \mu_T - k \cdot \sigma_T \quad (Ec. 3)$$

Siendo:

T el estadístico que mide el comportamiento de la característica de la cual se pretende conocer su calidad.

$\mu_T$  la media del estimador T.

$k$  la constante que indica la distancia entre el valor central y los límites de control.

$\sigma_T$  la desviación típica del estimador T.

#### 2.2.4. ANALISIS DE COMPONENTES PRINCIPALES (PCA)

El conjunto de métodos empleados para supervisar y detectar fallos en procesos se basa en el Control Estadístico de Procesos (SPC). Sin embargo, los resultados de estos métodos univariados no suelen ser muy satisfactorios, ya que, en caso de un fallo, varios de estos gráficos lo detectan de manera simultánea, dificultando la identificación de la causa del problema. Esto se debe a que, en general, las variables en un sistema controlado suelen estar interrelacionadas, y una causa especial puede afectar a más de una variable al mismo tiempo.

El Control Estadístico de Procesos univariante presenta diversas limitaciones notables. Una de ellas es que las variables se analizan de manera individual, lo que implica la necesidad de una gráfica de control para cada variable. Esto restringe la capacidad de monitorizar un número limitado de variables. Además, no se tiene en cuenta la correlación existente entre las variables, un aspecto crucial en los procesos de una planta industrial.

El Control Estadístico Multivariante (MSPC) aborda este desafío al utilizar, en muchos casos, variables latentes del proceso en lugar de considerar cada variable individualmente. Las técnicas que conforman el MSPC tienen la capacidad de construir modelos empíricos que describen la tendencia del sistema a partir de datos históricos del proceso. Este tipo de control se implementa con la intención de tener en cuenta la correlación de las variables entre sí.

El PCA reduce la dimensionalidad de un espacio original que se establece a partir de los datos históricos de las variables medidas del proceso mediante su proyección en un espacio de menor dimensionalidad. El PCA busca combinaciones lineales de las variables que ofrecen la descripción más precisa de la tendencia del proceso. Desde un punto de vista matemático, el PCA se fundamenta en la descomposición de la matriz de covarianza de las variables del proceso en las direcciones que mejor explican las principales fuentes de variabilidad de la información analizada. Este conjunto reducido de variables es lo que se llama 'componentes principales', las cuales no están correlacionadas entre sí.

Dentro de un espacio muestral que consta de  $m$  variables, el PCA nos permite identificar un número de factores subyacentes  $a$  (donde  $a$  es menor que  $m$ ) conocidos como componentes principales. Estas componentes principales logran contener casi toda la información presente en las  $m$  variables originales.

El objetivo fundamental del PCA es proyectar los puntos en un espacio de menor dimensión, conservando al máximo sus posiciones relativas. A través

Juan Domínguez González (2024) - EII - Universidad de Valladolid

de esta proyección, se busca preservar la variabilidad inherente a los puntos, tal como se ilustra en la Figura 8. [9]

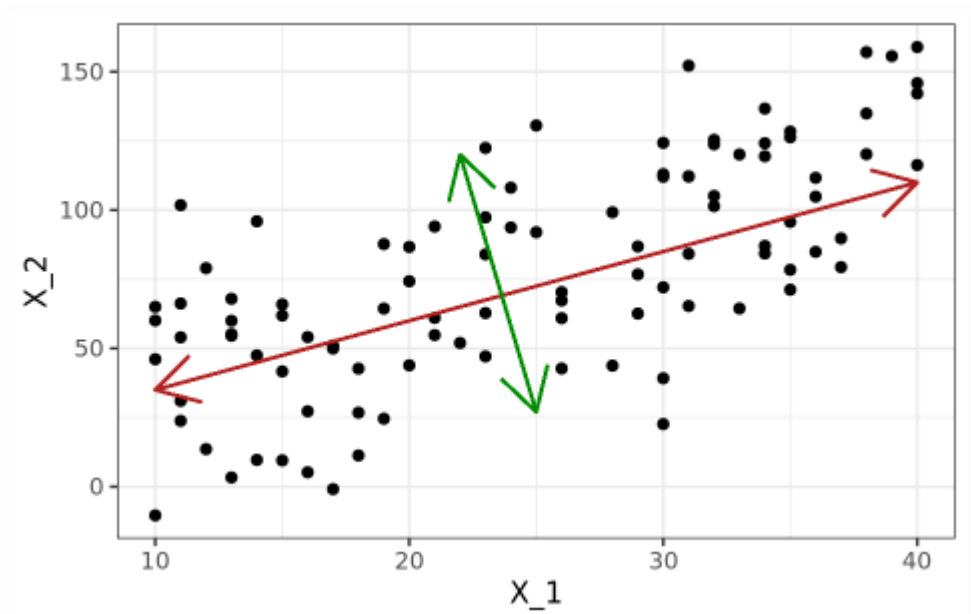


Figura 8: Interpretación Geométrica de PCA. Ejes de elipsoide [10]

Tal y como se ilustra en la Figura 8, una dirección permite conservar la variabilidad de los puntos manteniendo la información esencial. No obstante, en la dirección perpendicular (indicada por la línea verde), la variabilidad es reducida, resultando en una pérdida de información sobre las distancias entre los puntos.

El proceso de PCA, requiere de un tratamiento inicial de adaptación de los datos representativos del comportamiento normal de la planta. Eliminando variables inapropiadas, con errores de medida muy grandes y normalizando los datos para que todos tengan el mismo peso y puedan ser comparables.

La normalización de los datos de la planta se realiza con media 0 y varianza 1, para obtener la matriz de datos normalizados ( $X_n$ ). Es decir, a cada valor de la matriz de datos original ( $X$ ) se le resta la media ( $\mu_m$ ) y se divide por su desviación estándar ( $\sigma_m$ ), de su variable. Siendo  $X$  una matriz de dimensión  $n \times m$ :

$$X_n[n, m] = \frac{X[n, m] - \mu_m}{\sigma_m} \quad (\text{Ec. 4})$$

Conocida la matriz de datos normalizada ya podemos trabajar con el método que propone el PCA, el primer paso es calcular la matriz de covarianzas, denominada R:

$$R = \frac{1}{(n-1)} X^T X \quad (\text{Ec. 5})$$

Realizando la descomposición en valores singulares (SVD) de R, se obtiene:

$$R = V \Lambda V^T \quad (\text{Ec. 6})$$

Donde:

$\Lambda \in \mathbb{R}^{m \times m}$  es una matriz diagonal que contiene los valores propios reales no negativos de R en orden decreciente de manera que  $\lambda_1 \geq \lambda_2 \geq \lambda_m \geq 0$  y cuyo valor es  $\lambda_i = \sigma_i^2$ .

$V \in \mathbb{R}^{m \times m}$  es una matriz ortogonal ( $V^T = V^I \rightarrow V V^T = I$ ) cuyas columnas son los vectores propios de R.

Con el objetivo de mantener la máxima información de los datos, se requiere del trabajo con un número reducido  $a$ , de los valores propios de R, de tal forma que ( $a < m$ ). La elección del número apropiado de valores propios depende de la variabilidad que se quiera permitir al sistema. Mediante el Test del porcentaje de la varianza, se realiza una operación iterativa para descubrir el número de valores propios que son necesarios para cumplir con el requisito de variabilidad deseado. De la siguiente manera:

$$\text{variabilidad}_i = \frac{\sum_0^i \lambda_i}{\text{traza}(\lambda)} \quad (\text{Ec. 7})$$

Conocido el número  $a$ , de valores propios más grandes requerido para el estudio, se recogen los vectores propios (*loadings*) asociados a esos valores propios en la matriz  $P \in \mathbb{R}^{m \times a}$ . Posteriormente se proyectan los valores de la matriz de datos normalizada ( $X_n$ ) sobre el espacio de menor dimensión, obteniéndose un nuevo vector de componentes principales (*scores*) que se denomina  $T \in \mathbb{R}^{n \times a}$ . [9]

$$T = Xn P \quad (\text{Ec. 8})$$

Una vez definido este nuevo espacio T se pueden comprobar que se cumplen algunas propiedades. Definiendo  $t_i$  como la columna i-esima de T se cumple:

- $Var(t_1) \geq Var(t_2) \geq Var(t_a)$
- $Media(t_i) = 0$
- $t_i^T t_j = 0 \rightarrow i \neq j$
- No existe ninguna otra expansión ortogonal de a componentes que capture más variación de los datos. [8]

A partir de la multiplicación matricial de P y T, se puede volver al espacio de partida, obteniéndose la matriz  $\hat{X} \in \mathbb{R}^{n \times m}$ .

$$\hat{X} = T P^T \quad (\text{Ec. 9})$$

Conocida esta, se puede definir la matriz de residuos (ruido) como la diferencia entre el espacio original y el recalculado.

$$E = X - \hat{X} \quad (\text{Ec. 10})$$

Finalmente se puede obtener la expresión del espacio original de la siguiente manera:

$$X = T P^T + E \quad (\text{Ec. 11})$$

Lo que viene a expresar que se puede representar el espacio original a partir de los componentes principales seleccionados y el espacio del ruido.

Una vez obtenido el subespacio con los componentes principales se pasa a la fase de detección de fallos mediante PCA para ello se definen dos estadísticos que van a permitir definir un umbral para detectar anomalías en el proceso. El estadístico  $T^2$  o de Hotelling monitoriza el espacio de los

componentes principales mientras que el estadístico Q o SPE monitoriza el espacio de residuos. [9]

### 2.2.5. ESTADÍSTICOS

Para la monitorización y control estadístico del estado del proceso mediante el análisis de componentes principales, se emplean diversos estadísticos. Los estadísticos posibilitan la creación de gráficos de control para supervisar el estado del proceso. Este enfoque permite monitorizar el estado del proceso de manera univariable y visualizar el umbral de normalidad en un solo gráfico.

#### Estadístico $T^2$ o de Hotelling

El estadístico  $T^2$  monitoriza el espacio de dimensión  $a$ , de las componentes principales. Para calcular este estadístico se van a utilizar las matrices obtenidas en el método de PCA.

$$T^2 = X P \Lambda_a^{-1} P^T X^T \quad (\text{Ec. 12})$$

Siendo  $\Lambda_a \in \mathbb{R}^{a \times a}$  la matriz que contiene las primeras  $a$  filas y columnas de  $\Lambda$ , para  $a$  componentes principales elegidas.

Cuando este estadístico supera un cierto umbral, se puede inferir que el proceso está fuera de control, es decir, que se ha producido un fallo. Este umbral puede calcularse fuera del contexto operativo principal, es decir, fuera de línea, utilizando los datos de muestreo de la siguiente manera:

$$T_\alpha^2 = \frac{(n^2 - 1) a}{n (n - a)} F_\alpha(a, n - a) \quad (\text{Ec. 13})$$

Donde  $n$  es el número de muestras utilizado para el cálculo del PCA y  $F_\alpha(a, n - a)$  es el valor crítico de la distribución F (distribución de Fisher-Snedecor) con  $n$  y  $n - a$  grados de libertad. El parámetro  $\alpha$  representa el nivel de significancia, determinando el grado de fiabilidad para falsas alarmas, sus valores más comunes suelen ser 0.01 y 0.05.

El estadístico  $T^2$  se fundamenta en las primeras  $a$  componentes principales, ofreciendo así una medida de las desviaciones de las variables latentes. Estas variables latentes son aquellas que explican de manera óptima la variabilidad del proceso. De esta manera, este estadístico solo detecta un evento si la variación en las variables latentes es mayor que la variación explicada por las causas comunes. Sin embargo, pueden ocurrir otros tipos de eventos que generen variaciones significativas en el espacio de ruido descartado para el cálculo de este estadístico. Estos eventos se identifican mediante el estadístico  $Q$ , el cual se explica a continuación.[9]

### Estadístico $Q$ o SPE

Este estadístico es conocido con el nombre de error de predicción al cuadrado (SPE). Monitoriza el espacio de dimensión  $(m-a)$ , enfocado en los valores singulares más pequeños.

Se puede calcular a partir de las matrices obtenidas en el modelo de PCA, de la siguiente manera

$$Q = r r^T \quad (\text{Ec. 14})$$

Siendo  $r$  el vector de residuos, que se define así:

$$r = X (I - P P^T) \quad (\text{Ec. 15})$$

Este estadístico indica la diferencia, o residuo entre el vector de observación actual y su proyección en el espacio reducido de las  $a$  componentes principales. Cuando el sistema se encuentra bajo control estadístico los valores del estadístico son muy pequeños ya que sus variaciones se deben al ruido.

Al igual que en el caso del estadístico anterior también es necesario definir un umbral, a partir de las observaciones del comportamiento normal del proceso, de la siguiente manera:

$$Q_\alpha = \frac{\sigma_Q^2}{2 \mu_Q} \text{Chi} \left( p, 2 \frac{\mu_Q^2}{\sigma_Q^2} \right) \quad (\text{Ec. 16})$$

Para calcular el umbral se tiene en cuenta la media  $\mu_Q$  y la desviación estándar  $\sigma_Q$  del estadístico Q. Se requiere la utilización de la distribución Chi-cuadrado, en la que  $p$  es la probabilidad que se va a utilizar del 99% y  $2 \mu_Q^2 / \sigma_Q^2$  son los grados de libertad de la distribución.

La detección de fallos se hace mucho más robusta utilizando ambos estadísticos ya que trabajan en espacios de datos distintos. Para ambos casos se definirá un fallo, en el momento en que las observaciones de los estadísticos sobrepasen los umbrales  $k$  veces consecutivas. La elección del valor  $k$  debe ser precavida porque si se elige un número demasiado pequeño se puede incurrir en posibles falsas alarmas, mientras que si se elige un número demasiado grande se pueden pasar por alto ciertos fallos. [9]

### 2.3. DETECCIÓN DE FALLOS CON TÉCNICAS DE MACHINE LEARNING Y DEEP LEARNING

El “Machine Learning” es una disciplina científica que hace que los sistemas “aprendan” a partir de datos. En vez de programar, paso a paso, cada solución específica para cada necesidad planteada, tal y como se realiza en el enfoque de la programación convencional, el área de machine Learning está dedicada al desarrollo de algoritmos genéricos que pueden extraer patrones de diferentes tipos de datos.

En este punto se podría pensar que el proceso completo de machine Learning es fácilmente automatizable, cuando realmente no es el caso, un ingeniero de datos debe llevar a cabo numerosas tareas específicas. Algunas de las que ya se ha comentado en este proyecto como pueden ser la identificación de la fuente de datos y su limpieza, la eliminación de información que este fuertemente correlacionada, la búsqueda de información sesgada, la realización de las normalizaciones necesarias... [11]

Con el objetivo de poder abordar cualquier tipo de tarea específica, es necesario conocer los distintos tipos de machine learning.

- Aprendizaje supervisado:
- Aprendizaje no supervisado
- Aprendizaje semi-supervisado
- Aprendizaje por refuerzo

En este texto nos centráremos en el desarrollo únicamente de los dos primeros tipos propuestos el supervisado y el no supervisado.



Juan Domínguez González (2024) - EII - Universidad de Valladolid

El aprendizaje supervisado en machine learning se aplica cuando cada dato, o conjunto de datos de entrada al algoritmo (muestra) tiene asociada una etiqueta por intervención humana. A partir de estos datos de entrada el algoritmo puede generar datos de salida esperados ya que los datos introducidos están etiquetados y clasificados.

Se observan una gran división entre los dos procedimientos mayoritarios de aprendizaje supervisado (Figura 9), en función del tipo de variable de salida que se está prediciendo:

La clasificación es un tipo de problema en el que el objetivo es predecir una etiqueta de categoría o clase discreta. En otras palabras, se busca asignar una instancia de datos a una categoría predefinida. Ejemplos comunes de clasificación incluyen predecir si un correo electrónico es spam o no, clasificar imágenes en diferentes clases (por ejemplo, perros, gatos, autos) o predecir si un paciente tiene una enfermedad en función de ciertos síntomas.

La regresión es un tipo de problema en el que el objetivo es predecir un valor numérico continuo en lugar de una etiqueta de clase. Por lo tanto, en lugar de clasificar en categorías discretas, se intenta predecir una cantidad real. Ejemplos típicos de problemas de regresión incluyen predecir el precio de una casa basándose en sus características, predecir la temperatura para el día siguiente o estimar el tiempo que tomará completar una tarea en función de varios factores.

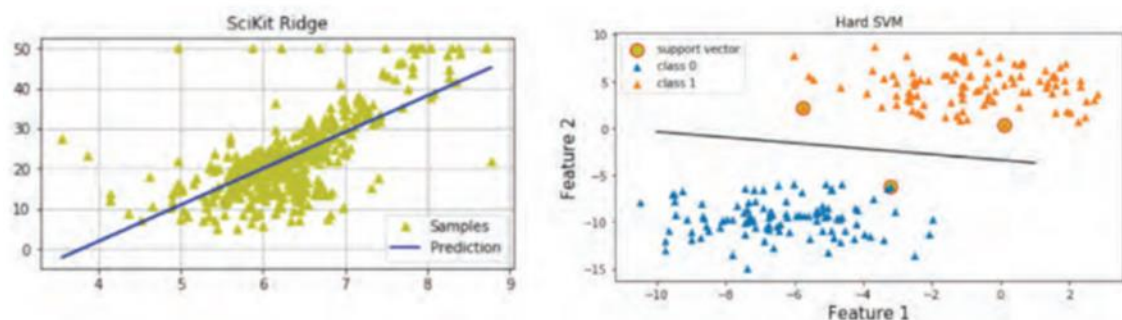


Figura 9: Aprendizaje supervisado. A la izquierda representación de gráfico de regresión lineal y a la derecha gráfico de clasificación lineal [12]

El aprendizaje no supervisado se basa en algoritmos que aprenden de datos con elementos no etiquetados, buscando patrones entre ellos. Los datos se introducen en el sistema, sin etiquetar, por tanto, no es necesaria la intervención humana.

Dentro de este tipo de aprendizaje automático, como el caso anterior, también se encuentran dos grandes bloques, referentes a técnicas distintas.

El clustering es un proceso que se basa en la agrupación de un conjunto de datos en subgrupos homogéneos, donde los elementos dentro de cada subgrupo son más similares entre sí en comparación con los elementos de otros grupos. El objetivo es encontrar patrones y estructuras ocultas en los datos basados en similitudes y diferencias en sus características. Por ejemplo, la agrupación de los clientes de un supermercado según sus preferencias de compra para personalizar campañas de marketing. Algunos algoritmos de clustering comunes son el K-means, el clustering jerárquico y el DBSCAN (Density-Based Spatial Clustering of Applications with Noise).

La asociación implica encontrar relaciones interesantes o patrones entre variables en un conjunto de datos. Estos patrones pueden implicar que ciertos elementos tienden a aparecer juntos en transacciones o eventos. Uno de los métodos más conocidos en este ámbito es el algoritmo Apriori. Los algoritmos de asociación intentan encontrar reglas de asociación, que son declaraciones en la forma "Si X, entonces Y", donde X e Y son conjuntos de elementos. Estas reglas indican la probabilidad de que Y ocurra dado X. Un ejemplo puede ser la identificación de patrones de compra en un historial de transacciones, como la tendencia de que los clientes que compran pan también compren mantequilla.

Dentro del Machine Learning aparece, como un subconjunto de este, un concepto denominado Deep Learning. Se caracteriza por el uso de redes neuronales profundas para aprender automáticamente representaciones complejas de datos, mientras que Machine Learning es un enfoque más amplio que incluye una variedad de algoritmos y técnicas para aprender patrones de datos. El Deep Learning se ha destacado en problemas con grandes conjuntos de datos y complejidad, pero requiere de mayor poder computacional que el Machine Learning convencional.

El Deep Learning ha revolucionado diversos sectores y ha encontrado aplicaciones en una amplia gama de campos.

- Visión por computadora: reconocimiento de imágenes, detección de objetos...
- Procesamiento de lenguaje natural: traducción automática, procesamiento de texto, chatbots y asistentes virtuales...
- Reconocimiento de voz y procesamiento de audio: generación de voz sintética.
- Automatización industria y robótica:
- Sistemas de recomendación personalizados

*Juan Domínguez González (2024) - EII - Universidad de Valladolid*

- Salud y medicina: diagnóstico médico y descubrimiento de medicamentos
- Finanzas: prevención de mercado, gestión de riesgos...
- Automóviles autónomos y transporte: conducción autónoma, gestión del tráfico... [12]

### 2.3.1. REDES NEURONALES ARTIFICIALES

Las redes neuronales artificiales son una destacada manifestación de la inteligencia artificial (IA) y se sitúan en el ámbito del aprendizaje automático o "machine learning." Esto permite que las máquinas puedan llevar a cabo tareas o funciones que anteriormente no podíamos automatizar con los sistemas convencionales.

Las investigaciones relacionadas con las redes neuronales artificiales o neuroredes, se enfocan en modelar la forma en que los sistemas nerviosos biológicos procesan información. Este interés proviene, en gran medida, de la manera en que funciona el cerebro humano, que es significativamente diferente al funcionamiento de una computadora digital convencional.

El cerebro humano se caracteriza por ser un sistema altamente complejo, no lineal y paralelo. En términos simples, esto significa que puede llevar a cabo numerosas operaciones simultáneas, a diferencia de las computadoras convencionales que operan de forma secuencial, es decir, realizan una sola operación a la vez.

En este contexto, una red neuronal artificial se asemeja a un procesador de información con una distribución altamente paralela, compuesta por múltiples unidades de procesamiento simples llamadas neuronas. Son unidades o nodos que reciben información del exterior o de otras neuronas, de manera similar a los impulsos nerviosos que reciben las neuronas del cerebro humano, las procesan y generan un valor de salida que alimenta a otras neuronas de la red o son la salida hacia el exterior de la red. [13]

De manera análoga al funcionamiento de las neuronas del cerebro humano, la neurona artificial se compone de tres partes o fases:

- Un conjunto de entradas que son las interconexiones por donde reciben información del exterior de la red o de otras neuronas.
- Un conjunto de funciones como son la de propagación, la de activación y la de transferencia.

Juan Domínguez González (2024) - EII - Universidad de Valladolid

- La salida de la neurona que es la interconexión por donde entrega el resultado al exterior de la red neuronal o a otras neuronas. [14]

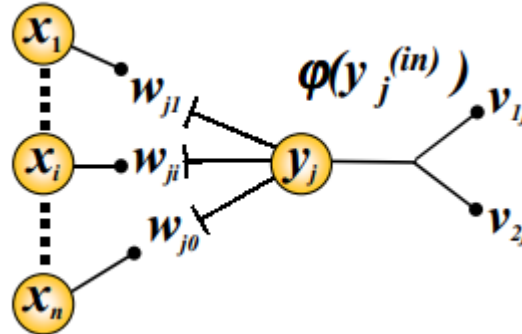


Figura 10: Esquema de Neurona Artificial [13]

En la Figura 10, se representa un esquema de una neurona, donde la neurona de interés es  $y_j$ . Un total de "n" neuronas  $x_i$  envían señales de entrada, que son valores numéricos con información. Los valores  $w_{ji}$  corresponden a los pesos sinápticos en las dendritas de la neurona  $y_j$ . La notación utilizada es la siguiente: el primer índice indica la neurona hacia la cual se dirige la información, y el segundo índice indica la neurona de la cual proviene la información.

Lo que hace cada peso sináptico es simplemente multiplicar a su entrada correspondiente y define la importancia relativa de cada entrada. Entonces tenemos que la entrada total a la neurona  $y_j$  es:

$$y_{j(\text{entrada})} = \sum_{i=1}^n w_{ji} x_i \quad (\text{Ec. 17})$$

La neurona se activa cuando la suma total de las entradas supera un umbral específico. Para lograr esto, se aplica una función de activación sobre  $y_{j(\text{entrada})}$ , que podría ser, por ejemplo, una función tipo escalón o sigmoidea, como la función tangente hiperbólica. Estas son las funciones de activación no lineales, más típicas (Figura 11).

Como se mencionó anteriormente, una neurona se activa cuando su entrada total supera un umbral específico. Sin embargo, en muchas ocasiones, es útil poder ajustar este umbral para que sea más difícil o fácil que la neurona se active (es decir, subir o bajar el umbral). Si bien es posible realizar esta modificación directamente, a menudo resulta más conveniente y eficiente añadir lo que se llama una "neurona de inclinación," denotada como  $x_0$ , a la

Juan Domínguez González (2024) - EII - Universidad de Valladolid

cual se le asigna un valor constante de 1. Luego, se utiliza un peso sináptico  $w_{j0}$  para esta neurona de inclinación (Figura 12).

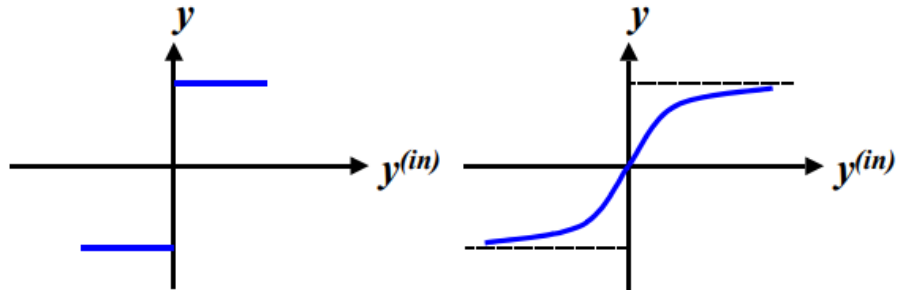


Figura 11: Funciones de activación. A la izquierda escalón y a la derecha tangente hiperbólica [13].

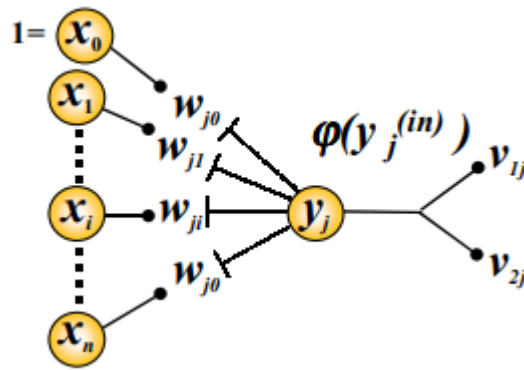


Figura 12: Esquema de neurona con inclinación [13].

De esta forma obtenemos que la entrada a la neurona artificial tendría la siguiente forma:

$$y_{j(\text{entrada})} = \sum_{i=0}^n w_{ji} x_i ; \text{siendo } x_0 = 1 \tag{Ec. 18}$$

Una vez se obtiene la expresión que contiene la información de entrada, se le aplica la función de activación ( $\sigma$ ) [13]. De la forma:

$$y_j = \sigma(y_{j(\text{entrada})}) \tag{Ec. 19}$$

En base a su estructura las neuronas artificiales pueden diferenciarse en:



*Juan Domínguez González (2024) - EII - Universidad de Valladolid*

- **Monocapa:** la red neuronal consiste en una sola capa de neuronas que recibe valores externos, los procesa y produce valores de salida como resultado. Las neuronas de la capa no están interconectadas entre sí
- **Multicapa:** se trata del tipo de red neuronal más utilizado. Consiste en una capa de entrada que recibe valores del exterior, seguida de una serie de capas intermedias u ocultas que procesan la información, y finalmente una capa de salida que proporciona los resultados de la red neuronal. En este tipo de red, las neuronas de una capa están conectadas con todas las neuronas de la capa siguiente, pero no se conectan entre sí.
- **Convolutiva:** son similares a las redes neuronales multicapa, pero se diferencian en que las neuronas de cada capa no se conectan con todas las neuronas de la capa siguiente, sino solo con un subconjunto de estas. En este caso, se habla de especialización de las neuronas en grupos, lo que reduce la cantidad de unidades necesarias y la complejidad de los sistemas que utilizan este tipo de redes.
- **Recurrentes:** en estas redes neuronales, las neuronas no están organizadas en capas estructuradas, sino que están interconectadas de manera no estructurada entre sí. Esta disposición permite que estas redes tengan memoria, lo que significa que la información generada en iteraciones anteriores puede influir en el procesamiento en momentos futuros. [14]

### 2.3.2. REDES NEURONALES DE PROPAGACIÓN HACIA DELANTE

Una red de propagación hacia adelante es un tipo de red neuronal en la que las conexiones entre los nodos no tienen recirculación. Estas redes son el tipo más básico de red neuronal y pueden dividirse en redes de propagación hacia adelante sin capa oculta o perceptrón, y redes con una o más capas ocultas.

El perceptrón se considera el modelo más básico de red neuronal artificial, con una capa de entrada y una capa de salida. Se trata de un algoritmo de aprendizaje supervisado utilizado para clasificación binaria, que agrupa los elementos de un conjunto en dos clases diferentes según una regla de clasificación. Fue desarrollado en 1958 por Frank Rosenblatt para el reconocimiento de imágenes. Sin embargo, pronto se descubrió que solo era capaz de reconocer patrones simples debido a su naturaleza como clasificador lineal. Esto significa que utiliza combinaciones lineales de las entradas para tomar decisiones, lo que limita su capacidad. A pesar de estas

limitaciones, se ha demostrado que algunos modelos lineales producidos por el perceptrón se asemejan bastante a una neurona real.

Una red de propagación hacia adelante multicapa (Figura 13) consta de una capa de entrada, una o más capas ocultas de neuronas y una capa de salida. A excepción de los nodos de entrada, todas las neuronas utilizan funciones de activación no lineales. Estas redes son más complejas que los perceptrones, ya que emplean algoritmos de entrenamiento basados en el cálculo de gradientes.

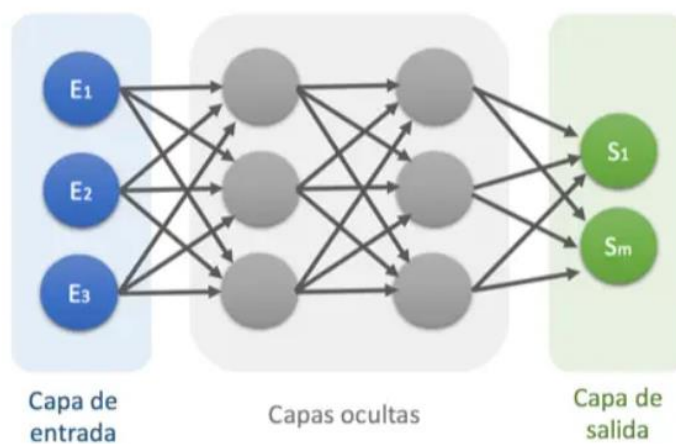


Figura 13: Red de propagación hacia adelante multicapa [16]

El método de entrenamiento más comúnmente utilizado para estas redes es la retropropagación, que implica comparar la señal de salida deseada con cada una de las salidas reales de la red. El error resultante se propaga hacia las capas inmediatamente anteriores, distribuyéndose entre cada neurona según su contribución a la salida original. A medida que la red se entrena, se ajustan los parámetros (pesos y bias) para minimizar el error y mejorar el rendimiento de la red. [15]

### 2.3.3. AUTOENCODERS

Un autoencoder es un algoritmo de aprendizaje no supervisado que pertenece al tipo específico de red neuronal de propagación hacia adelante. En un autoencoder, la entrada es la misma que la salida, y consta de tres partes principales: el codificador, la capa oculta y el decodificador. El proceso de codificación implica proyectar las variables de entrada en la capa oculta, mientras que la decodificación consiste en proyectar la capa oculta en la salida.





*Juan Domínguez González (2024) - EII - Universidad de Valladolid*

Inicialmente, el autoencoder comprime la entrada en una representación latente a través del codificador y luego reconstruye la salida utilizando esta representación. El modelo se entrena minimizando los errores de reconstrucción entre las variables de entrada y salida.

La aplicación más común de los autoencoders es la reducción de dimensionalidad y el aprendizaje de características en conjuntos de datos para el reconocimiento de imágenes. Sin embargo, en la última década, se han utilizado para desarrollar potentes redes neuronales de múltiples capas ocultas capaces de procesar grandes cantidades de datos. Enfoque conocido como aprendizaje profundo o "deep learning".

Los algoritmos de codificación automática buscan la reducción o compresión de dimensionalidad. Basándose en una serie de propiedades importantes:

- **Específico para un tipo de datos:** los codificadores automáticos solo pueden comprimir de manera significativa datos similares a aquellos con los que han sido entrenados. Dado que aprenden funciones específicas para los datos de entrenamiento dados, son diferentes de un algoritmo de compresión de datos estándar.
- **Con pérdida:** la salida del codificador automático no será exactamente la misma que la entrada, será una representación cercana pero degradada.
- **Sin supervisión:** para entrenar un codificador automático no se necesita una atención sofisticada, únicamente arrojarle los datos de entrada sin procesar. Los codificadores automáticos se consideran una técnica de aprendizaje no supervisado, ya que no necesitan etiquetas explícitas para entrenar. Sin embargo, se pueden considerar como auto supervisados ya que generan sus propias etiquetas a partir de los datos de entrenamiento. [17]

La Figura 14 muestra una visualización detallada de un autoencoder. En primer lugar, la entrada pasa a través del codificador, que es una red neuronal artificial (ANN) completamente conectada, para producir el código. El decodificador, que tiene una estructura ANN similar, genera la salida utilizando solo el código. El objetivo es obtener una salida que sea idéntica a la entrada.

Es importante destacar que la arquitectura del decodificador es la imagen especular del codificador en esta representación. Aunque esto no es un requisito estricto, suele ser el caso en muchos autoencoders. Lo único que se



necesita es que la dimensionalidad de la entrada y la salida sea la misma. Se puede experimentar con cualquier aspecto intermedio de la red.[18]

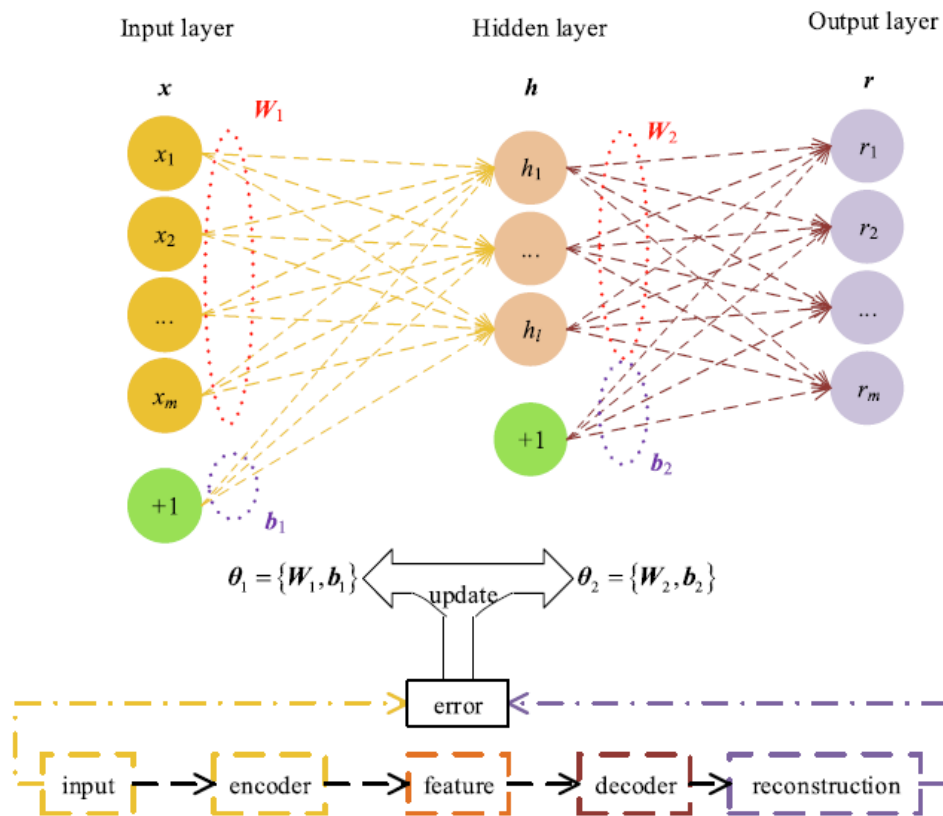


Figura 14: Estructura de Autoencoder [18]

Existen cuatro parámetros que se deben configurar para el correcto entrenamiento de un codificador automático:

- Tamaño del código: número de nodos en la capa intermedia (capa oculta). Se debe tener en cuenta que una cantidad de nodos más pequeña da como resultado una mayor compresión.
- Número de capas: el autoencoder puede ser tan profundo como queramos, pero debe ser simétrico, es decir, contener el mismo número de capas de codificación que de decodificación.
- Número de nodos por capa: el número de nodos por capa disminuye con cada capa posterior del codificador y aumenta nuevamente en el decodificador. Además, debido a la propiedad de simetría las capas de codificación y decodificación correspondientes entre sí tendrán que tener el mismo número de nodos.

Juan Domínguez González (2024) - EII - Universidad de Valladolid

- Función de pérdida: generalmente se utiliza el error cuadrático medio (mse) o la entropía cruzada binaria. Si los valores de entrada se encuentran en el rango [0, 1], normalmente se usa entropía cruzada; de lo contrario, se aplica el error cuadrático medio. [17]

Una vez entrenado el autoencoder, para su utilización en la detección de fallos se definen los mismos dos estadísticos ( $T^2$  y  $Q$ ) que para la detección de fallos mediante PCA, el estadístico  $T^2$  usará la salida de la capa más oculta, que aquí se llamará  $h$ , y para el estadístico  $Q$  se usarán los residuos. Se definen los estadísticos mediante las siguientes expresiones.

$$T^2 = (h - \bar{h}_{med})^T \Sigma_{h,med}^{-1} (h - \bar{h}_{med}) \quad (Ec. 20)$$

Donde  $\bar{h}_{med}$  se trata de la media de las características locales extraídas de la capa oculta y  $\Sigma_{h,med}^{-1}$  la matriz de covarianza de las mismas.

$$Q = (res - \overline{res}_{med})^T \Sigma_{res,med}^{-1} (res - \overline{res}_{med}) \quad (Ec. 21)$$

Donde  $\overline{res}_{med}$  se trata de la media de las características locales extraídas de la predicción y  $\Sigma_{res,med}^{-1}$  la matriz de covarianza de las mismas.

Siendo el residuo ( $res$ ) la diferencia entre los datos de la capa de salida y los datos de la entrada. [18]

$$res = x - \hat{x} \quad (Ec. 22)$$

Los umbrales de cada estadístico, al igual que en el caso de PCA, se establecen utilizando el percentil 99, lo que significa que el 99% de los datos de cada estadístico estarán por debajo de ese umbral.

#### 2.3.4. AUTOENCODERS DISTRIBUIDOS

Se introduce una nueva estrategia de monitorización denominada Autoencoder Distribuido, la cual se basa en la tecnología de aprendizaje profundo. Está diseñada para procesos industriales de gran escala que cuentan con múltiples unidades operativas.

Este método propuesto tiene la capacidad de identificar con precisión las observaciones donde se encuentran los fallos, al mismo tiempo que proporciona información esencial a los operadores del proceso. Esta información indica si el fallo detectado es solamente local, afectando a una unidad operativa específica, o si se trata de un fallo global que impacta en todo el proceso.

El concepto de autoencoder distribuido se define como la aplicación de varios codificadores automáticos sobre la agrupación de las variables de un proceso completo.

Su fundamento es la agrupación de las variables presentes en el proceso de diversas formas, para poder aplicar un autoencoder a cada agrupación. De esta manera el entrenamiento y la detección de fallos se realizará sobre grupos de variables menos numerosos y que a priori tengan algo que ver entre ellas.

Por lo general se pretende obtener un mejor rendimiento de la detección de los fallos, aunque la contribución principal del autoencoder distribuido es su capacidad para detectar claramente si ocurre un fallo en la unidad local o la global. [18]

### Mínima redundancia máxima relevancia

Para buscar la mejor agrupación de variables, el objetivo es encontrar un conjunto de variables  $S$ , con  $m$  variables, que en conjunto tengan la mayor dependencia de la variable objetivo  $c$ . Obviamente, es difícil calcular la densidad multivariada.

Se recurre al método de mínima redundancia máxima relevancia mRMR. La relevancia máxima busca la mayor interrelación de las variables que componen el conjunto  $S$  con la variable  $c$ .

$$\max D(S, c); D = \frac{1}{|S|} \sum_{x^i \in S} I(x^i, c) \quad (\text{Ec. 23})$$

Siendo  $|S|$  el número de variables que componen el conjunto  $S$  e  $I(x^i, c)$ , la función que evalúa la relación entre la variable  $x^i$  en cada caso y la variable  $c$ , en este caso la información mutua entre las variables.

Las variables seleccionadas para formar el conjunto podrían representar gran redundancia entre ellas, debido a su máxima relevancia con la variable  $c$ . Por

tanto, se incorpora la regla de mínima redundancia para seleccionar variables mutuamente excluyentes.

$$\min R(S); R = \frac{1}{|S|^2} \sum_{x^i, x^j \in S} I(x^i, x^j) \quad (\text{Ec. 24})$$

Se define el operador  $\Phi(D, R)$  que combina ambas reglas optimizándolas de la siguiente manera:

$$\max \Phi(D, R); \Phi = D - R \quad (\text{Ec. 25})$$

Si se satisface la condición impuesta por el operador, la variable evaluada formará parte del mismo grupo o bloque que la variable objetivo  $c$ . [19]

### 2.3.5. ÍNDICE DE INFERENCIA BAYESIANA (BIC)

Después de obtener los resultados de la monitorización de cada bloque de variables evaluadas con su propio autoencoder, se encuentra la problemática de que no es factible observar constantemente tantas estadísticas,  $T^2$  y  $Q$ , por cada bloque.

Para ofrecer una indicación clara del estado del proceso, se necesita fusionar dichas estadísticas. En este estudio, se emplea la estrategia de inferencia bayesiana para combinar los resultados de la monitorización. En esta estrategia de fusión, la probabilidad de fallo del estadístico  $T^2$  en cada bloque se puede calcular mediante ("N" para la condición normal, "F" para la condición de fallo).

De esta forma se pueden calcular las probabilidades condición normal o fallo para el estadístico  $T^2$  para cada bloque,  $b=1, \dots, B$ , siendo  $B$  el número total de bloques en que se ha dividido la planta:

$$P_{T^2}(F|x_b) = \frac{P_{T^2}(x_b|F) P_{T^2}(F)}{P_{T^2}(x_b)} \quad (\text{Ec. 26})$$

Donde:

$$P_{T^2}(x_b) = P_{T^2}(x_b|N) P_{T^2}(N) + P_{T^2}(x_b|F) P_{T^2}(F) \quad (\text{Ec. 27})$$

$$P_{T^2}(x_b|N) = e^{-T_b^2(x_b)/T_{b,lim}^2} \quad (\text{Ec. 28})$$

$$P_{T^2}(x_b|F) = e^{T_{b,lim}^2/-T_b^2(x_b)} \quad (\text{Ec. 29})$$

$P_{T^2}(N)$  y  $P_{T^2}(F)$  se determinan a partir del nivel de confianza  $\alpha$  y  $1 - \alpha$  respectivamente.

$T_b^2(x_b)$  representa el estadístico T2 de la muestra actual en el bloque b, y  $T_{b,lim}^2$  muestra el valor del umbral al 99% establecido para el bloque b.

De manera similar se calculan las probabilidades de fallo y condición normal para el estadístico Q.

$$P_Q(F|x_b) = \frac{P_Q(x_b|F) P_Q(F)}{P_Q(x_b)} \quad (\text{Ec. 30})$$

Donde:

$$P_Q(x_b) = P_Q(x_b|N) P_Q(N) + P_Q(x_b|F) P_Q(F) \quad (\text{Ec. 31})$$

$$P_Q(x_b|N) = e^{-Q_b(x_b)/Q_{b,lim}} \quad (\text{Ec. 32})$$

$$P_Q(x_b|F) = e^{Q_{b,lim}/-Q_b(x_b)} \quad (\text{Ec. 33})$$

$P_Q(N)$  y  $P_Q(F)$  se determinan a partir del nivel de confianza  $\alpha$  y  $1 - \alpha$  respectivamente.

Juan Domínguez González (2024) - EII - Universidad de Valladolid

$Q_b(x_b)$  representa el estadístico  $Q$  de la muestra actual en el bloque  $b$ , y  $Q_{b,lim}$  muestra el valor del umbral al 99% establecido para el bloque  $b$ .

Por lo tanto, las estadísticas por probabilidades finales se pueden escribir como:

$$BIC_{T^2} = \sum_{b=1}^B \left\{ \frac{P_{T^2}(x_b|F) P_{T^2}(F|x_b)}{\sum_{b=1}^B P_{T^2}(x_b|F)} \right\} \quad (Ec. 34)$$

$$BIC_Q = \sum_{b=1}^B \left\{ \frac{P_Q(x_b|F) P_Q(F|x_b)}{\sum_{b=1}^B P_Q(x_b|F)} \right\} \quad (Ec. 35)$$

Durante el proceso de monitorización, el límite de confianza se establece como  $\alpha$  y si el valor de  $BIC_{T^2}$  o  $BIC_Q$  es mayor que  $1-\alpha$ , se considera que se ha detectado un fallo.



# CAPÍTULO III: PLANTA TENNESSEE EASTMAN

### 3.1. DESCRIPCIÓN DE LA PLANTA

Durante la década de 1980, surgió un interés académico por el desarrollo de un problema real que sirviera de referencia común en el control de procesos industriales de todo tipo, pero especialmente químicos.

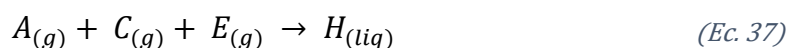
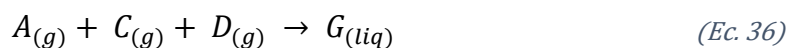
En 1993, la empresa estadounidense Eastman Chemical Company colaboró con la Universidad de Tennessee para desarrollar un proyecto centrado en la creación de una planta experimental basada en una planta real de la propia compañía química. Este proyecto, conocido como la planta Tennessee Eastman (TEP), fue ideado por Downs y Vogel con el propósito de plantear un problema que sirviera como referencia para el desarrollo y la evaluación de diversas tecnologías de control de procesos, especialmente en estudios relacionados con el control total de la planta y el control multivariable.

La planta se distingue por su alta no linealidad y complejidad, así como por la gran cantidad de variables que la componen, lo que la hace capaz de simular una multitud de escenarios distintos. Esto permite la investigación y análisis de diversos métodos de detección y diagnóstico de fallos.

En la Figura 15 se muestra un diagrama simplificado de la planta TE. El proceso consta de cinco operaciones unitarias: una reacción que tiene lugar en un reactor, una condensación, una separación líquido-vapor, una compresión de reciclaje y una desorción, también conocida como stripping, que se realiza en una columna stripper.

El reactor es un tanque de agitación continua (CSTR) de dos fases, donde se generan los productos G y H y un subproducto F a partir de cuatro reactivos A, C, D y E, y un inerte B. Las reacciones que tienen lugar en el reactor para obtener los productos mencionados son las siguientes: [20]

- Reacciones de los productos:



- Reacciones de los subproductos:





Todas las reacciones en la planta se caracterizan por ser reversibles, exotérmicas y exhibir una relación lineal entre la concentración y el tiempo. Además, la velocidad de estas reacciones depende de la temperatura. En particular, la reacción que produce el producto G se distingue por tener una energía de activación más alta, lo que indica una mayor sensibilidad a los cambios de temperatura.

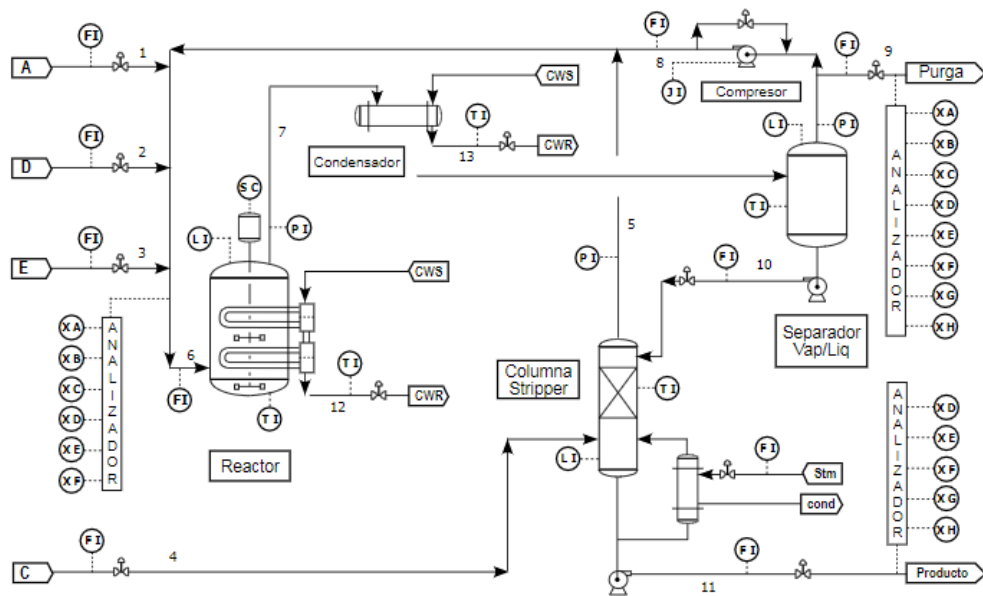


Figura 15: Diagrama de proceso de la planta TE [21]

El proceso comienza con la introducción de los reactivos gaseosos en el reactor, donde reaccionarán para formar los productos en fase líquida. Para acelerar esta reacción en la fase gaseosa, se añadirá un catalizador no volátil disuelto en la fase líquida, que permanecerá dentro del reactor en todo momento. Además, se empleará un sistema de refrigeración interno para eliminar el calor generado durante la reacción.

Los productos y los reactivos no reaccionados salen del reactor en forma de vapor y se dirigen a un condensador para enfriar la fase gaseosa y convertirla en líquida. Luego, pasan por un separador de líquido y vapor. Los componentes no condensados se reciclan mediante un compresor centrífugo y se devuelven al reactor para mejorar la eficiencia del proceso. Los componentes condensados son dirigidos a una columna de extracción de productos, donde, junto con la alimentación del reactivo C, se realiza una operación de desorción para eliminar los reactivos restantes.

Finalmente, los productos G y H se separan en la base de la columna y se llevan a una sección de refinación aguas abajo que no se incluye en el problema, ya que no es relevante para el estudio. El inerte y el subproducto F se eliminan principalmente como vapor al pasar por el separador líquido-vapor.

### 3.2. DATOS DE LA PLANTA

El modelo del proceso cuenta con 12 variables manipuladas (Tabla 1) y 41 variables medidas (Tabla 2), de las cuales, las 22 primeras son variables del proceso y las 19 restantes son variables procedentes del muestreo de la planta cada cierto intervalo de tiempo.

Las 12 variables manipuladas proporcionan 12 grados de libertad, que corresponden a 9 válvulas de flujo, 2 válvulas de control de temperatura y una de velocidad de agitación.

Número de Variable	Variable	Unidades
XMV (1)	Flujo de alimentación D	kg h <sup>-1</sup>
XMV (2)	Flujo de alimentación E	kg h <sup>-1</sup>
XMV (3)	Flujo de alimentación A	kscmh
XMV (4)	Flujo de alimentación A y C	kscmh
XMV (5)	Válvula de recirculación del compresor	%
XMV (6)	Válvula de carga	%
XMV (7)	Flujo de líquido del separador LV	m <sup>3</sup> h <sup>-1</sup>
XMV (8)	Flujo de líquido de la columna stripping	m <sup>3</sup> h <sup>-1</sup>
XMV (9)	Válvula de vapor de la columna stripping	%
XMV (10)	Flujo de agua de refrigeración del reactor	m <sup>3</sup> h <sup>-1</sup>
XMV (11)	Flujo de agua en el condensador	m <sup>3</sup> h <sup>-1</sup>
XMV (12)	Velocidad del agitador del reactor	rpm

*Tabla 1: Variables manipuladas del proceso [21]*

<b>Número de Variable</b>	<b>Variable</b>	<b>Unidades</b>
XMEAS (1)	Flujo de alimentación A	kscmh
XMEAS (2)	Flujo de alimentación D	kscmh
XMEAS (3)	Flujo de alimentación E	kscmh
XMEAS (4)	Flujo de alimentación A y C	kscmh
XMEAS (5)	Flujo de recirculación	kscmh
XMEAS (6)	Flujo de alimentación al reactor	kscmh
XMEAS (7)	Presión del reactor	kPa
XMEAS (8)	Nivel del reactor	%
XMEAS (9)	Temperatura del reactor	°C
XMEAS (10)	Flujo de purga	kscmh
XMEAS (11)	Temperatura del separador	°C
XMEAS (12)	Nivel del separador	%
XMEAS (13)	Presión del separador	kPa
XMEAS (14)	Corriente del separador	m <sup>3</sup> h <sup>-1</sup>
XMEAS (15)	Nivel de destilador (stripper)	%
XMEAS (16)	Presión del destilador (stripper)	kPa
XMEAS (17)	Corriente del destilador (stripper)	m <sup>3</sup> h <sup>-1</sup>
XMEAS (18)	Temperatura del destilador (stripper)	°C
XMEAS (19)	Flujo de vapor del destilador (stripper)	kg h <sup>-1</sup>
XMEAS (20)	Potencia del compresor	kW
XMEAS (21)	Temp. de salida de agua de refrigeración del reactor	°C
XMEAS (22)	Temp. de salida de agua de refrigeración del separador	°C
XMEAS (23-28)	Concentración de la alimentación del reactor	% mol
XMEAS (29-36)	Concentración de la purga	% mol
XMEAS (37-41)	Concentración aguas abajo del destilador	% mol

*Tabla 2: Variables medidas del proceso [3]*

También se dispone de los datos correspondientes a las distintas anomalías que pueden suceder. Se trata de 21 fallos distintos, que en este proyecto se tiene como objetivo detectar.

<b>Número de Fallo</b>	<b>Fallo</b>	<b>Tipo de Fallo</b>
IDV (1)	Relación de flujo de alimentaciones A/C, composición de B constante	Escalón
IDV (2)	Composición de B con relación A/C constante	Escalón
IDV (3)	Temperatura de alimentación D	Escalón
IDV (4)	Temperatura de entrada de agua refrigerante al reactor	Escalón
IDV (5)	Temperatura de entrada de agua refrigerante al condensador	Escalón
IDV (6)	Pérdida de alimentación de A	Escalón
IDV (7)	Pérdida de presión en la corriente C	Escalón
IDV (8)	Composición de las alimentaciones A, B y C	Variación aleatoria
IDV (9)	Temperatura de alimentación D	Variación aleatoria
IDV (10)	Temperatura de alimentación C	Variación aleatoria
IDV (11)	Temperatura de entrada de agua refrigerante al reactor	Variación aleatoria
IDV (12)	Temperatura de entrada de agua refrigerante al condensador	Variación aleatoria
IDV (13)	Cinética de las reacciones	Variación lenta
IDV (14)	Válvula de agua refrigerante del reactor	Bloqueo
IDV (15)	Válvula de agua refrigerante del condensador	Bloqueo
IDV (16)	Desconocido	No especificado
IDV (17)	Desconocido	No especificado
IDV (18)	Desconocido	No especificado
IDV (19)	Desconocido	No especificado
IDV (20)	Desconocido	No especificado
IDV (21)	Desconocido	Constante

*Tabla 3: Fallos posibles en el proceso [22]*



*Juan Domínguez González (2024) - EII - Universidad de Valladolid*

En este proyecto se ha trabajado con dos conjuntos de datos, el primero se trata de una simulación de la planta con 960 observaciones de cada variable, donde el fallo ocurre a partir de la observación número 160. Los datos del modelo teórico de Tennessee Eastman son públicos y pueden obtenerse a través del siguiente enlace:

<http://web.mit.edu/braatzgroup/links.html>.

Para el entrenamiento de las redes neuronales que tienen muchos parámetros a calcular (los AE entran dentro de la denominada deep-learning) es necesario usar muchos más datos de los mostrados en ese link. Dichos datos están formados por 500 simulaciones de la planta TE en condiciones normales de funcionamiento durante 48 horas, de las cuales se han cogido datos de dimensiones (240000, 52), que se creen suficientes para un buen entrenamiento. Estos datos se pueden obtener del siguiente enlace.

(<https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/6C3JR1>)



# CAPÍTULO IV: APLICACIÓN

## 4.1. INTRODUCCIÓN A LA APLICACIÓN

Como se ha indicado anteriormente el objetivo de este estudio es comprobar la capacidad de detección de errores de distintos métodos. Para ello cada método se trabaja y entrena con datos en situaciones normales, sin fallos. Posteriormente se utilizan las características obtenidas de ese comportamiento en situación normal para el análisis de datos con fallos.

La planta de trabajo cuenta con 52 variables conocidas que se analizarán. Aun así, se ha comprobado en investigaciones previas que ciertos fallos, como los fallos 3, 9 y 15, son más difíciles de detectar debido a que con esos fallos las variables no experimentan cambios significativos en comparación con el comportamiento normal del proceso. Por esta razón en algunos métodos obviaremos el comportamiento de dichos fallos.

Se conoce que todos los ficheros de fallo de la planta Tennessee se han generado contando con un fallo teórico en la observación 160. Sabiendo que se realiza una observación cada 3 minutos, el tiempo de detección teórico será de 480 minutos.

Se iniciará la aplicación estudiando el método de PCA, continuando con un análisis a partir de una red neuronal de propagación hacia adelante llamada Autoencoder. Se profundizará más el análisis dividiendo las variables de estudio en varios bloques y estudiando un autoencoder por bloque. Finalmente se utilizará el método estadístico BIC, para formatear los datos obtenidos del estudio de autoencoders distribuidos y transformarlos en los parámetros comparables con los resultados de los estudios de los otros métodos.

La aplicación al completo ha sido diseñada en lenguaje Python. Mediante la herramienta Anaconda y más concretamente el software de Spyder, se ha trabajado el código a partir de funciones incluidas en la propia librería del programa, como puede ser *numpy*, *pandas* o *matplotlib*. Utilizadas para la realización de cálculos matemáticos, algebraicos y su representación gráfica. Además, se han utilizado otras librerías más complejas como pueden ser *tensorflow* y *keras*, utilizadas para la creación, entrenamiento y testeo de las redes neuronales.

## 4.2. ANALISIS DE COMPONENTES PRINCIPALES

El estudio comienza con la aplicación del método de análisis de componentes principales. Su funcionamiento como algoritmo es sencillo, conocidas las variables y las ecuaciones definidas en el apartado del estudio teórico, se pueden obtener valores representativos de un conjunto elevado de variables.

Como se ha visto previamente este método comienza con el estudio sobre una matriz de datos extraídos de la planta, cuyo comportamiento se asemeja lo más posible a un comportamiento sin fallos. Se trata de un fichero que contiene 960 muestras de datos para cada una de las 52 variables definidas en la planta Tennessee Eastman.

La matriz se normaliza a media cero y varianza uno, por columnas, es decir, por variables. Posteriormente se obtienen sus parámetros característicos, valores y vectores propios que permiten obtener el número de componentes principales para un porcentaje de variabilidad requerido.

En el caso de estudio, inicialmente este porcentaje de variabilidad se fija en el 90%, obteniendo así, un número de componentes principales  $a=32$ , dentro de las 52 variables iniciales.

A partir de este estudio se puede construir la matriz de componentes principales y la matriz de residuo. Las cuáles serán necesarias para el posterior estudio sobre los dos estadísticos con los que se va a trabajar, el Estadístico  $T^2$  o de Hotelling y el Estadístico Q o SPE.

Conocida esta información y siguiendo el procedimiento indicado en el segundo apartado de este texto, se pueden obtener valores de cada uno de los estadísticos, para cada observación. En concreto se obtienen 960 valores de cada estadístico.

Siendo este el comportamiento que se asume como “normal” o sin fallos, se obtiene también un parámetro denominado umbral para cada estadístico, el cual marcará un percentil al 99%. Es decir, de todos los valores de un estadístico obtenidos, este umbral marca el límite, asegurando que el 99% de ellos son inferiores al valor del umbral.

Los valores que toman los estadísticos en cada observación y su umbral o percentil al 99%, se representan en sendas gráficas (Figura 16), con el objetivo de comprender mejor el comportamiento y la dispersión de estos valores.

Como aclaración, en todo el estudio se ha representado en color azul el estadístico  $T^2$  y en color verde el estadístico Q, para favorecer la comprensión del estudio completo a simple vista.



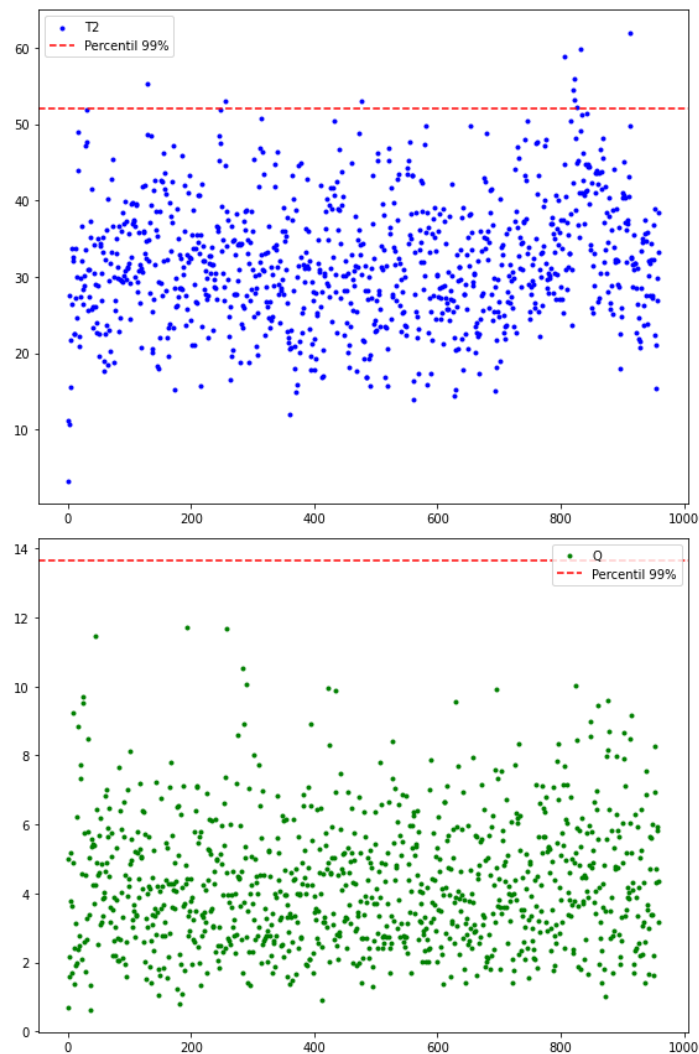


Figura 16: Representación de los valores de los estadísticos partiendo de un modelo sin fallos con estudio mediante PCA

Realizado el estudio del comportamiento de la planta en condiciones normales, se ha obtenido mucha información necesaria para su aplicación posterior.

Por tanto, se programa el guardado de los valores de los umbrales obtenidos para cada estadístico, los vectores de medias y desviaciones de la matriz de datos inicial por columnas (variables), y las matrices de valores,  $\Lambda_a$  y vectores propios,  $P$ , que corresponden únicamente al número de componentes principales.

### Detección de Fallos

Se aplica el método de análisis de componentes principales a conjuntos de observaciones de la planta Tennessee Eastman. Se conoce que estos conjuntos contienen fallos y se pretende observar la capacidad de reconocimiento por parte del método PCA.

Para ello, serán necesarios los valores provenientes del análisis de la planta en comportamiento normal guardados en el apartado anterior. Es decir, primero se cargan en el programa los umbrales obtenidos para cada estadístico, los vectores de medias y desviaciones de la matriz de datos inicial por columnas (variables), y las matrices  $P$  (de vectores propios) y  $\Lambda_a$  (de valores propios) correspondientes a los  $a$  componente principales seleccionados.

Posteriormente se carga uno a uno los ficheros de datos de fallos localizados para la planta y se realiza su estudio particular en cada caso.

Se normalizan de nuevo por variables a media cero y varianza uno, sin embargo, en este caso los valores de medias y desviaciones son los que se obtuvieron para los datos de comportamiento normal. Pretendiendo así que la normalización sea en base a los mismos valores para todos los ficheros con fallos.

Conocida la matriz normalizada en cada caso y las matrices de valores y vectores propios que corresponden al número de componentes principales obtenidas para los datos de comportamiento normal, se pueden obtener los valores de los estadísticos  $T^2$  y  $Q$  para cada observación, de manera similar a como se realizó para los datos de comportamiento normal.

Se obtienen sendas graficas que muestran los valores de los estadísticos para cada muestra, lo que permite observar más fácilmente su comportamiento. En todas ellas se muestran además los umbrales obtenidos en el estudio de datos de comportamiento normal, para su comparación.

Debido al gran número de ficheros de errores con los que se está realizando la experimentación, el estudio se va a centrar concretamente en el análisis de los fallos 7 y 17. En el caso de PCA, la representación de sus estadísticos será la de la Figura 17.

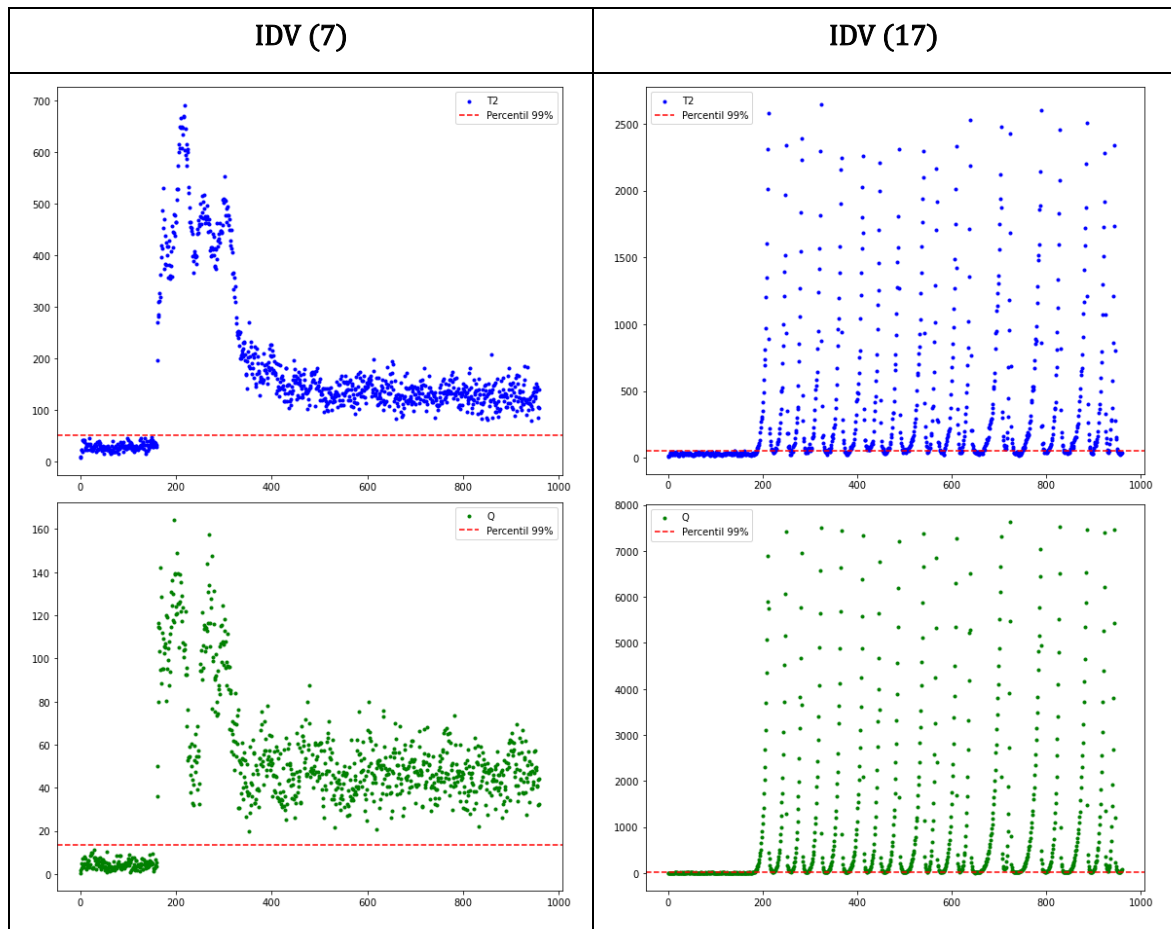


Figura 17: Representación de estadísticos para fallos IDV (7) y IDV (17) estudiados mediante PCA

Además de las representaciones gráficas, se calculan ciertos parámetros característicos del reconocimiento de los errores por parte del método concreto, que puedan ser comparables con los obtenidos con otros métodos.

En concreto se trata de las siguientes características, calculadas para ambos estadísticos:

- **Tiempo de detección** (td): medición temporal que define la observación en la que se detecta el fallo para cada fichero. Debido a la posibilidad de existencia de falsas alarmas, se requiere de la presencia consecutiva de cinco observaciones del estadístico que superen el valor del umbral, para considerarlo como fallo. Se toma el valor de la primera de las cinco observaciones consecutivas como valor del tiempo de detección en cada caso. Aunque se considere una medición temporal, para facilitar el estudio, en este texto se va a definir el tiempo de detección como la observación en la que se detecta el fallo.



*Juan Domínguez González (2024) - EII - Universidad de Valladolid*

De todas formas, si en un momento se requiere el cálculo de un valor temporal real, únicamente habrá que multiplicar dicha observación por tres, conocido que se realiza una observación cada tres minutos, obteniendo de esta manera un valor en minutos.

- **Número de alarmas** (na): se trata del valor porcentual que relaciona el número de observaciones que superan el umbral, desde el tiempo de detección teórico (160), con el número de observaciones existentes después del mismo, concretamente 800.
- **Falsas alarmas** (fa): este valor porcentual relaciona el número de alarmas, valores que superan el umbral, que se dan en las primeras 160 observaciones, tiempo de detección teórico, donde el comportamiento de la planta es normal.

En la Tabla 4 se muestran los valores de los parámetros a estudiar, pero como se ha comentado anteriormente se analizará particularmente los fallos 7 y 17.

En cuanto al IDV (7), se puede observar en la Figura 17 y la Tabla 4 que su reconocimiento es prácticamente perfecto, se determina el fallo en la observación 160 para ambos estadísticos, y se definen un 100% de alarmas detectadas y un 0% de falsas alarmas.

La detección del fallo denominado como IDV (17) se considera que no se realiza de manera totalmente correcta. De nuevo, la Figura 17 y la Tabla 4, nos aportan la información necesaria para este análisis. Se puede observar que el comportamiento de los estadísticos hasta el fallo se encuentra muy cerca del umbral, llegando a superarlo en ciertas ocasiones, de ahí que exista un porcentaje, aunque pequeño, de falsas alarmas en ambos casos. Respecto al tiempo de detección para el estadístico  $T^2$  se encuentra en la observación 188 y para el Q la 181, ambas por encima de la observación de detección de fallo teórica. Por otro lado, debido a su comportamiento asintótico aproximadamente cada 50 muestras, se observa que el porcentaje del número de alarmas es inferior al caso anterior, destacando sobre todo en el caso del estadístico  $T^2$ , que desciende casi hasta el 80%.

Número de Fallo	Estadístico T2			Estadístico Q		
	Tiempo de detección	Número de alarmas	Falsas alarmas	Tiempo de detección	Número de alarmas	Falsas alarmas
IDV (1)	166	99.25	1.25	162	99.75	1.25
IDV (2)	172	98.50	1.88	174	98.25	0.00
IDV (3)	200	2.50	2.50	960	0.75	0.00
IDV (4)	234	43.25	0.63	160	100.00	0.00
IDV (5)	170	27.25	0.63	160	21.63	0.00
IDV (6)	166	99.25	0.00	160	100.00	0.00
IDV (7)	160	100.00	0.00	160	100.00	0.00
IDV (8)	182	97.25	0.00	179	93.25	0.00
IDV (9)	960	3.63	3.13	960	0.25	0.63
IDV (10)	256	32.75	1.25	267	25.38	0.00
IDV (11)	190	52.50	0.00	215	54.38	1.25
IDV (12)	166	98.50	1.25	182	91.50	0.00
IDV (13)	206	94.88	1.88	201	94.88	0.00
IDV (14)	160	100.00	0.63	161	99.88	0.00
IDV (15)	960	2.88	1.25	960	0.38	0.00
IDV (16)	356	17.00	5.63	355	25.25	0.00
IDV (17)	188	79.13	0.63	181	94.88	0.63
IDV (18)	251	89.50	1.25	243	90.00	0.00
IDV (19)	904	12.25	1.88	960	6.13	1.25
IDV (20)	246	33.50	1.25	244	46.75	0.63
IDV (21)	442	41.75	1.25	662	40.38	0.63
Media Aritmética		58.36	1.34		61.13	0.30
Media Aritmética sin fallos IDV (3) (9) (15)		67,58	1,18		71,24	0,31

Tabla 4: Parámetros estadísticos obtenidos para los 21 errores utilizando el método de PCA

### 4.3. AUTOENCODER GENERAL

El estudio mediante este método se introduce en el mundo de las redes neuronales, su entrenamiento y la utilidad de sus parámetros de salida para el objetivo general del trabajo.

Inicialmente se crea un fichero denominado “Autoencoder” que contiene la función propiamente dicha de la red neuronal. En este caso se trabaja con un autoencoder de 3 capas intermedias, siendo la primera de codificación, seguida de la capa oculta y por último la de decodificación. Los parámetros de entrada que es necesario aportar a este fichero son los siguientes:

- **X\_train:** Se trata del fichero de datos de comportamiento normal que se aporta para el entrenamiento y testeo de la red neuronal. Para este proceso se trata de valores normalizados entre 0 y 1 por columnas, variables. Dado que es un autoencoder, el objetivo es que el modelo aprenda a reproducir la entrada, por lo que tanto los datos de entrada como los datos de salida son los mismos. En este caso se utiliza un fichero de 240000 observaciones.
- **nb\_epoch:** Especifica el número de épocas para entrenar el modelo. Una época es un ciclo completo a través del conjunto de datos de entrenamiento
- **batch\_size:** Define el número de muestras que se utilizarán en cada actualización del modelo.
- **hidden\_layer1:** Valor numérico que recoge el número de neuronas que se encontraran en la capa de codificación y como es un autoencoder, por simetría, también en la capa de decodificación.
- **hidden\_layer2:** Define el número de neuronas de la capa oculta.

A partir de estos parámetros y haciendo uso de funciones importadas de las librerías keras y tensorflow, se podrá crear la red neuronal, entrenarla y testearla. Se define una característica a mayores, el porcentaje de datos de entrada que se utiliza para entrenamiento de la red (80%) aplicando el 20% restante para el testeo.

Esta función aporta como salida los datos de la capa oculta y la salida de la red neuronal. Estos valores son necesarios para calcular los estadísticos T2 y Q, según se ha visto en el capítulo 2 de este trabajo.

Juan Domínguez González (2024) - EII - Universidad de Valladolid

En este caso el entrenamiento del autoencoder referente al fichero de 240.000 observaciones, de comportamiento normal y el estudio de los correspondientes estadísticos lleva a la siguiente representación gráfica (Figura 18).

El autoencoder generado se ha entrenado durante 100 épocas para el (nb\_epoch) y utilizando 64 muestras en cada actualización del mismo (batch\_size). La red neuronal se compone de una capa de entrada y otra de salida de 52 neuronas cada una y tres capas intermedias. Debido a la simetría que presenta este tipo de red, la primera y tercera capa se componen de 30 neuronas y la segunda capa (hidden\_layer) está formada por 20 neuronas, es decir tiene una estructura 52-30-20-20-52.

Esta estructura se ha definido tras sucesivos ensayos en los que se han variado estos parámetros, obteniéndose estos como los parámetros óptimos para este autoencoder concreto.

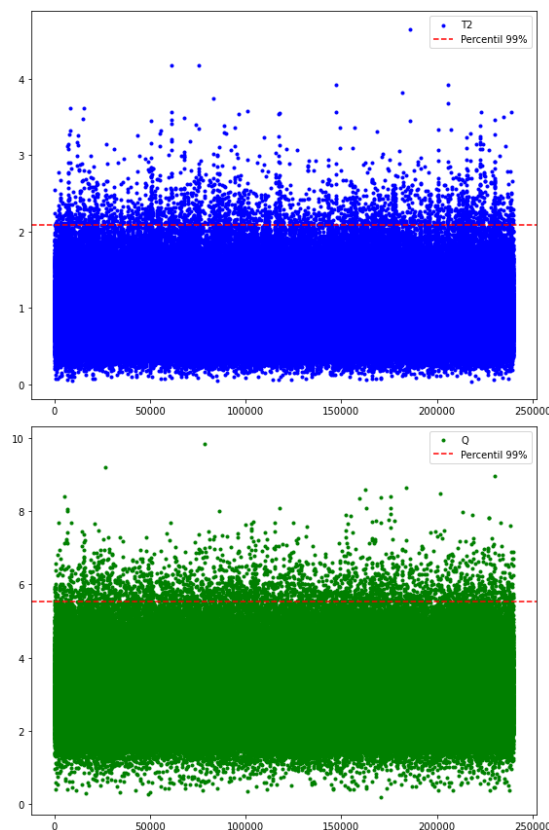


Figura 18: Representación de los valores de los estadísticos partiendo de un modelo sin fallos con estudio mediante Autoencoder general

La Figura 18 muestra los valores que toman los estadísticos analizados para cada una de las observaciones y el umbral (percentil al 99%) en cada caso.

Igual que se realizó en el modelo PCA, de este estudio para datos sin fallo se guardan ciertos parámetros que serán necesarios para el análisis de los ficheros de fallos. Estos son en este caso, los valores mínimos y máximos utilizados para la normalización, los vectores de medias de las características principales extraídas de la capa oculta y la predicción, las matrices de covarianzas de las mismas y los umbrales de cada estadístico.

También es necesario guardar los modelos del autoencoder y del encoder entrenados con datos de comportamiento normal para poder aplicarles a los datos de fallo.

### **Detección de Fallos**

En el proceso de detección de fallos mediante la aplicación de la red neuronal autoencoder general, son necesarios los modelos del autoencoder y del encoder entrenados en el paso anterior. Por tanto, inicialmente se cargan tanto los parámetros como los modelos que se han indicado en el final del apartado anterior.

Teniendo en cuenta esos datos comienza el análisis uno a uno de cada fichero de datos con fallos, al igual que se realizó para el fichero de comportamiento normal.

Se cargan los datos, se normalizan entre 0 y 1 y se aplican los modelos autoencoder y encoder. Con los datos que se obtienen de la capa oculta y la salida de la red en cada caso y los parámetros de medias y covarianzas de comportamiento normal, se calculan los valores de los estadísticos para cada observación.

Al igual que en el caso anterior, se grafica la evolución de los estadísticos para cada observación. Cada fichero de fallo tendrá su comportamiento, que debería ser similar al comportamiento representado en las gráficas del método PCA, de no ser así, es posible que alguno de los métodos, o incluso los dos, no estén detectando del todo bien el fallo.

Como se ha indicado previamente el estudio se va a centrar en la representación de dos fallos concretos, que son el IDV (7) e IDV (17), para su posterior análisis. En la Figura 19, se puede observar el comportamiento de los valores de sus estadísticos  $T^2$  y  $Q$ , una vez aplicado el autoencoder general.

Para completar el análisis, al igual que en el método de PCA, se calculan los mismos parámetros de cada estadístico para cada fichero de fallo. Se trata del tiempo de detección, el número de alarmas (%) y las falsas alarmas (%). Todos ellos pueden verse representados en la Tabla 5 que se encuentra a continuación.



En cuanto al IDV (7), se puede observar en la Figura 19 y la Tabla 5 que su detección por parte de este método al igual que ocurría con PCA es prácticamente perfecto, se determina el fallo en la observación 160 para ambos estadísticos, definiéndose un 100% de alarmas detectadas. Sin embargo, se puede observar que en cuanto al estadístico  $T^2$ , las falsas alarmas son distintas de 0%, suponen un valor pequeño permisible pero característico.

La detección del fallo IDV (17) por ambos estadísticos es más tardía respecto a la detección mediante PCA. También se observa que para ambos estadísticos el número de alarmas es un porcentaje más pequeño y las falsas alarmas se reducen al 0%.

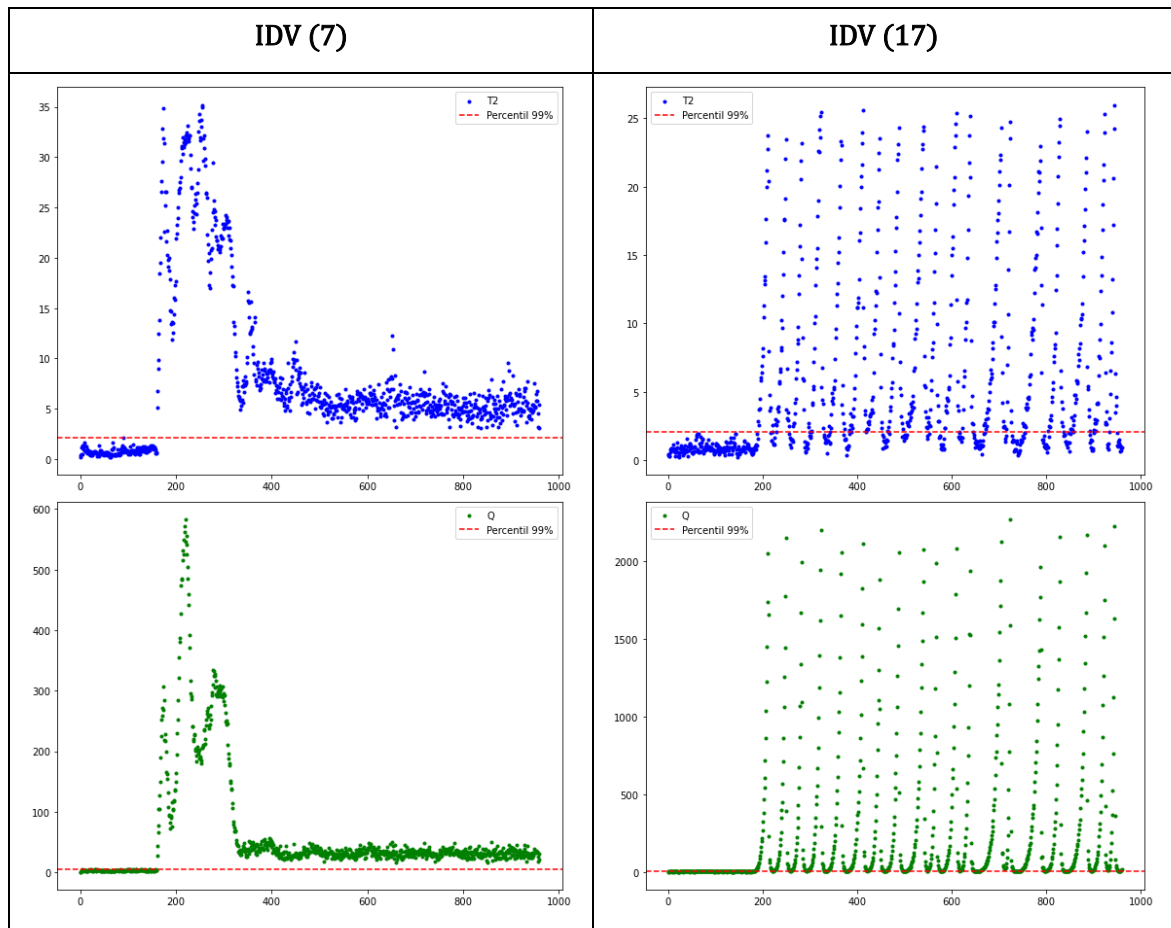


Figura 19: Representación de estadísticos para fallos IDV (7) y IDV (17) estudiados mediante Autoencoder general

Comparando los valores de las medias de los parámetros para el caso de PCA (Tabla 4) y Autoencoder general (Tabla 5). Se puede observar que el autoencoder realiza una mejor detección mediante el estadístico Q, sin embargo, el análisis mediante el estadístico  $T^2$  es peor.

Número de Fallo	Estadístico T2			Estadístico Q		
	Tiempo de detección	Número de alarmas	Falsas alarmas	Tiempo de detección	Número de alarmas	Falsas alarmas
IDV (1)	167	99.13	0.00	162	99.75	0.00
IDV (2)	173	98.38	0.00	172	98.50	0.63
IDV (3)	960	0.25	0.63	960	1.25	0.00
IDV (4)	960	3.50	0.00	162	99.88	1.25
IDV (5)	174	23.25	0.00	160	100.00	1.25
IDV (6)	168	99.00	0.63	160	100.00	1.25
IDV (7)	160	100.00	0.63	160	100.00	0.00
IDV (8)	186	96.50	0.00	177	97.88	0.00
IDV (9)	960	1.00	0.63	960	1.50	0.63
IDV (10)	257	32.00	0.00	207	46.50	0.00
IDV (11)	463	16.88	0.63	165	72.25	0.63
IDV (12)	181	97.88	1.25	162	99.00	0.63
IDV (13)	208	94.13	0.00	197	95.38	0.63
IDV (14)	163	82.13	0.00	160	100.00	1.25
IDV (15)	960	1.63	0.00	960	1.38	0.63
IDV (16)	472	13.50	2.50	355	39.75	0.00
IDV (17)	190	70.25	0.00	183	92.88	0.00
IDV (18)	254	88.50	0.63	243	89.88	0.63
IDV (19)	960	9.75	0.00	960	12.25	0.63
IDV (20)	246	26.25	0.00	242	51.00	0.63
IDV (21)	686	34.25	0.63	647	41.88	2.50
Media Aritmética		51.82	0.39		68.61	0.63
Media Aritmética sin fallos IDV (3) (9) (15)		60.29	0.38		79.82	0.66

*Tabla 5: Parámetros estadísticos obtenidos para los 21 errores utilizando el método de Autoencoder general*

#### 4.4. AUTOENCODER DISTRIBUIDO

El estudio del comportamiento de la planta mediante el método de autoencoder distribuido, se basa en una concreción del estudio realizado para un autoencoder general.

El procedimiento de entrenamiento de la red neuronal sigue los mismos pasos descritos en el apartado anterior. Sin embargo, en esta ocasión, en lugar de entrenar un único autoencoder para las 52 variables de la planta, se crean bloques de variables relacionadas entre sí. Esta agrupación se realiza según el método de Máxima Relevancia y Mínima Redundancia (mRMR), que se detalla en el apartado 2.3.4 del estudio teórico.

La agrupación de variables como se ha visto anteriormente depende de la relación que establezca cada una de ellas con la variable objetivo que define un bloque. De esta manera se define un umbral como el límite inferior de la relación que puede existir entre una variable y la variable objetivo de un bloque para que estas pertenezcan al mismo bloque.

No existe un número mínimo ni máximo de variables que pueden formar parte de un mismo bloque, únicamente depende de la relevancia y la redundancia que exista entre ellas, de ahí la importancia del método.

Para calcular este número de variables por bloque, lo que se hace es escoger una primera variable objetivo, por ejemplo,  $x_1$ , y se calcula el mRMR de todas las variables del proceso con esta variable  $x_1$ , de entre todas se eligen las variables con un valor mayor de mRMR para formar bloque con  $x_1$ , lo cual se fija con un umbral. Una vez elegido el primer bloque, se borran de la lista de variables, todas las variables elegidas, y se elige otra para ser la variable objetivo, por ejemplo, la variable con menor valor de mRMR con  $x_1$ , y se repite el proceso hasta que todas las variables están asignadas a un bloque.

El umbral se puede definir de múltiples maneras, una posible opción es simplemente calcular el valor medio de mRMR entre todas las variables con respecto a la variable objetivo, todas las variables por encima de ese umbral pertenecen al mismo bloque que la variable objetivo, y el resto no.

Dentro de este estudio se van a barajar tres posibilidades a la hora de definir los bloques en los que se van a dividir las variables. En el primer caso se va a asumir que el umbral es la propia media, posteriormente se realizará un segundo estudio en el que la media se incrementará un 50% y el tercer caso se reducirá un 50%.

#### 4.4.1. Umbral límite de relevancia igual a la media

Aplicando este valor del umbral se obtienen 7 bloques de variables, encontrándose en cada uno de los bloques las variables siguientes:

- **B1:** 1, 12, 15, 44, 48, 49
- **B2:** 3, 4, 7, 8, 10, 11, 13, 16, 18, 19, 22, 25, 31, 35, 43, 47, 50
- **B3:** 5, 17, 42, 46, 52
- **B4:** 2, 9, 21, 51
- **B5:** 20, 27, 28, 33, 34, 36
- **B6:** 6, 23, 24, 29, 38, 39, 41
- **B7:** 14, 26, 30, 32, 37, 40, 45

Definidos cada uno de los bloques como un conjunto de variables, se define y entrena un autoencoder para cada uno de ellos. Dividiendo los valores del fichero de 240.000 observaciones de comportamiento normal en función de la variable para tenerlos en cuenta únicamente en el bloque del que forme parte.

Tras sucesivos ensayos variando la posible estructura de cada uno de los autoencoders se define el número de neuronas por capa para la capa de entrada, las tres capas intermedias y la capa de salida de cada bloque de la siguiente manera:

- **B1:** 6-4-2-4-6
- **B2:** 17-10-5-10-17
- **B3:** 5-4-2-4-5
- **B4:** 4-3-2-3-4
- **B5:** 6-4-2-4-6
- **B6:** 7-5-3-5-7
- **B7:** 7-5-3-5-7

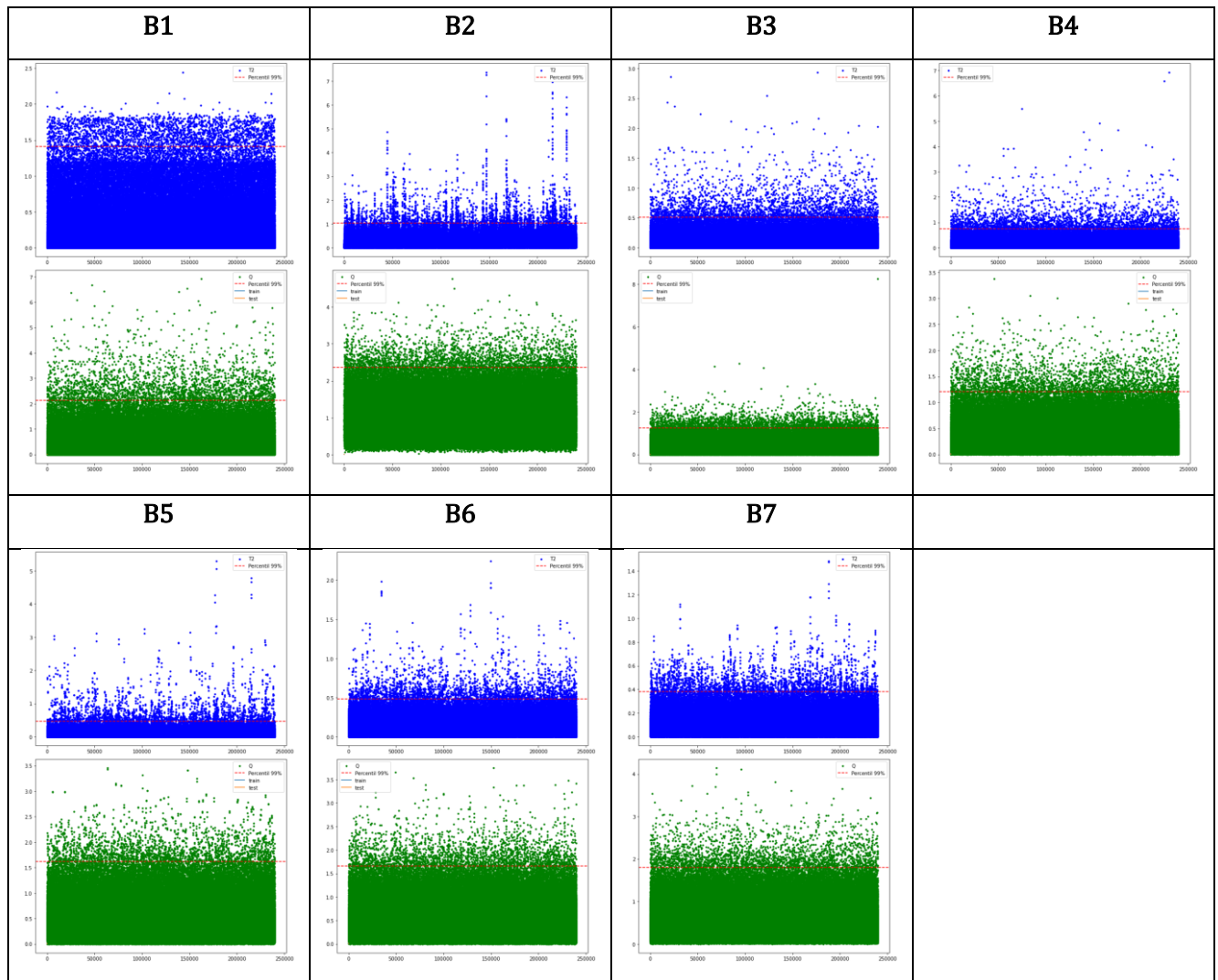


Figura 20: Representación de los valores de los estadísticos partiendo de un modelo sin fallos para el estudio de cada uno de los bloques de un Autoencoder Distribuido

Al igual que en el caso del autoencoder general, para cada bloque del autoencoder distribuido se realiza el estudio de los estadísticos  $T^2$  y  $Q$  y se representa sus valores y umbral en sendas gráficas (Figura 20).

La Figura 20 muestra los valores que toman los estadísticos analizados para cada una de las observaciones y el umbral (percentil al 99%) en cada caso.

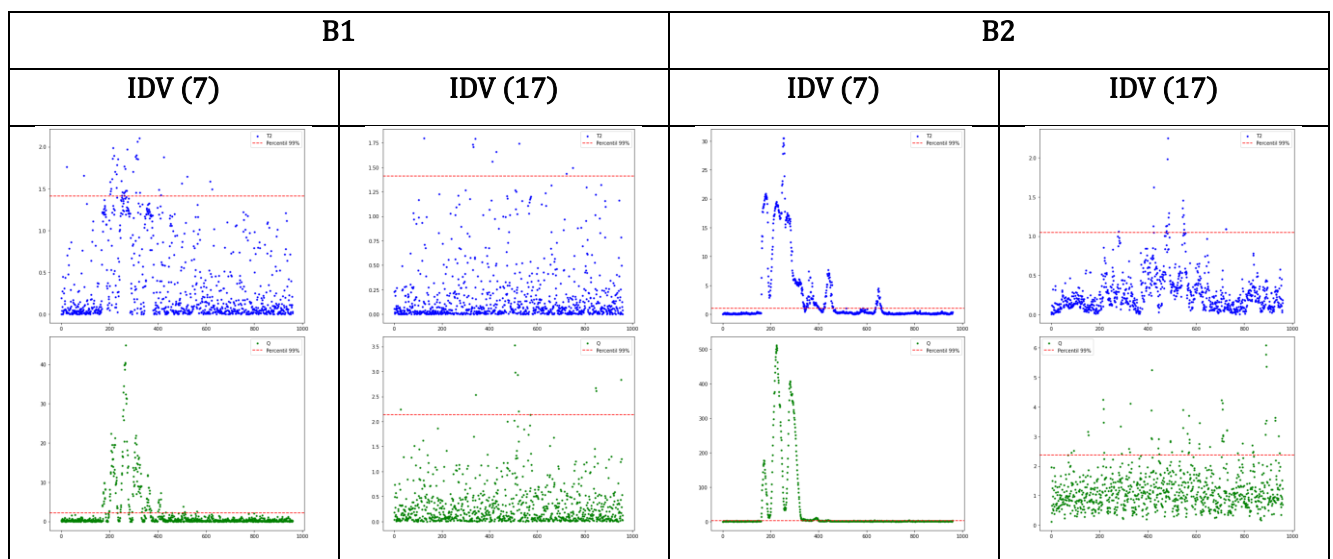
Igual que se realizó en el modelo del autoencoder general, de este estudio para datos sin fallo también se guardan los parámetros que serán necesarios para el análisis de los ficheros de fallos. Teniendo en cuenta que en este caso se han entrenado 7 redes, una por bloque, por tanto, es necesario guardar la información de cada red.

## Detección de fallos

La detección de fallos en este modelo se lleva a cabo de manera similar al método del autoencoder general. Sin embargo, en este caso, el proceso debe ser iterativo para identificar fallos en cada una de las redes entrenadas para cada bloque de variables.

Se grafica la evolución de los estadísticos para cada fallo, sin embargo, en este caso también se realiza este proceso para cada red. Cada fichero de fallo tendrá su comportamiento y posiblemente todos los fallos no se detecten en todas las redes.

Como se ha indicado previamente el estudio se va a centrar en la representación de dos fallos concretos, que son el IDV (7) e IDV (17), para su posterior análisis. En la Figura 21, se puede observar el comportamiento de los valores de sus estadísticos  $T^2$  y  $Q$ , para cada bloque del autoencoder distribuido.



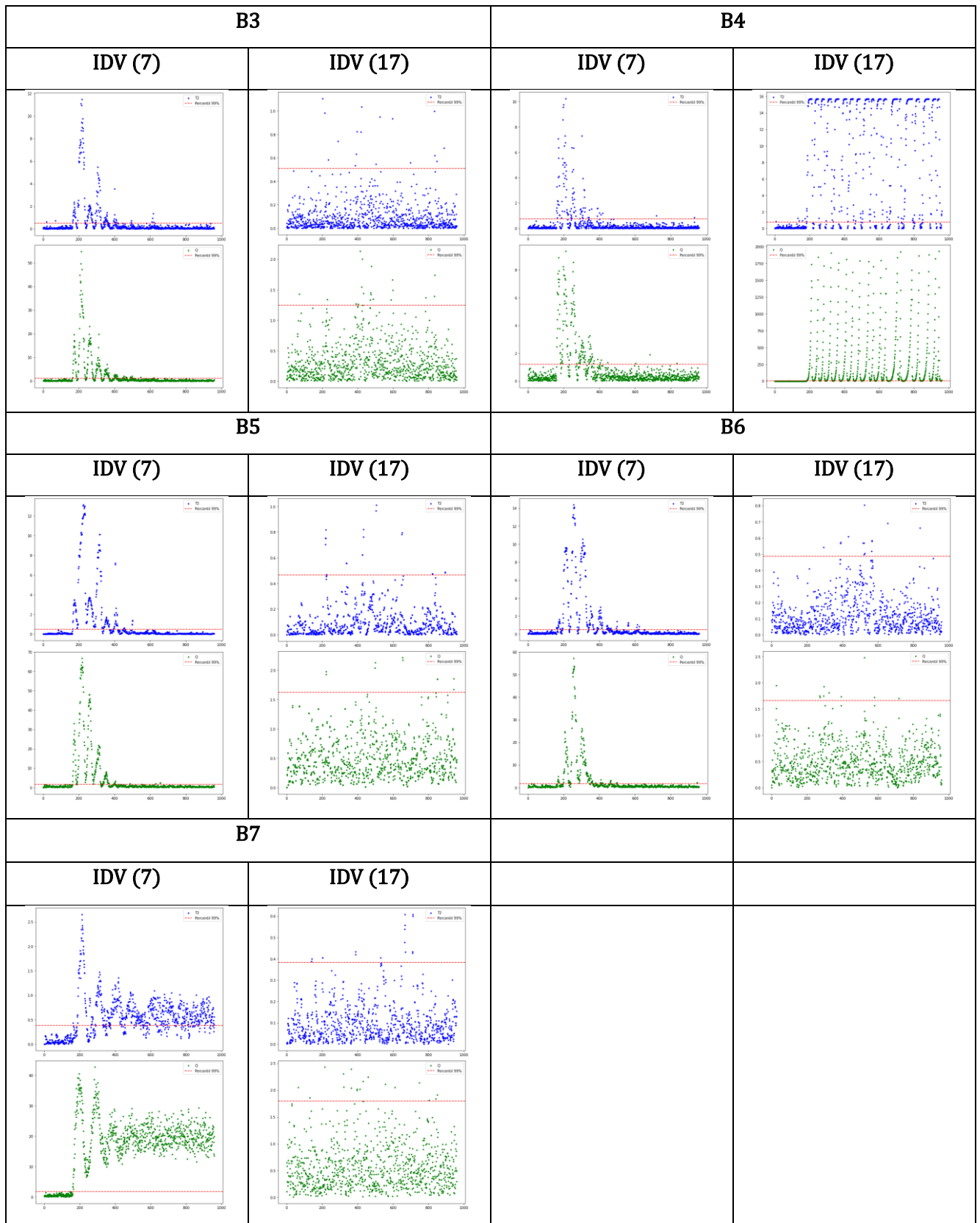


Figura 21: Representación de estadísticos para fallos IDV (7) y IDV (17) estudiados mediante cada red del Autoencoder Distribuido



Se puede observar en la Figura 21 que el comportamiento de los estadísticos para cada error en cada caso difiere mucho del comportamiento observado en el método de PCA y autoencoder general, que era similar entre sí.

Debido a la forma de parametrizar y entrenar cada una de las redes, la información está mucho más sesgada que en los casos donde se trataban todas las variables de manera general. Llevando a que cada una de las redes solo sea capaz de detectar alguno de los fallos.

Para observar de manera directa esta capacidad de cada red de detección de fallos, en este caso se va a hacer referencia en la siguiente tabla únicamente a los valores del número de alarmas detectadas para el estadístico Q. Ya que se trata del valor más representativo en el reconocimiento del error. Se refleja en una misma tabla los distintos valores de dicho elemento para cada fichero de fallo y cada red perteneciente al autoencoder distribuido (Tabla 6).

Sabiendo que para poder considerar un error como detectado el valor porcentual del número de alarmas tiene que ser del 100% o cerca de este. Analizando la Tabla 6, se puede observar que la detección de los fallos en cada bloque es distinta y por lo general reducida.

Esta información se contrasta con el comportamiento de los valores del estadístico Q, reflejado en cada caso en la Figura 21. Concretamente para los errores de estudio, IDV (7) e IDV (17), se puede observar que en ambos casos es únicamente una de las redes la que detecta el fallo. En comparación con el tratamiento general de datos tanto en PCA como en autoencoder general, donde se detectaban ambos fallos de manera relativamente sencilla.

En el caso de IDV (7) la red que detecta el error es la correspondiente al bloque 7, que tiene 7 variables. Sin embargo, el error IDV (17), se detecta en el bloque 4 que tiene un tamaño muy reducido, únicamente contiene 4 variables.

Esto permitirá focalizarse en un número pequeño de variables (7) para el fallo 7 y 4 para el fallo 17, que nos dirán que variable es la responsable del fallo. Por lo tanto, este método sigue detectado los fallos, pero además es mejor porque permite focalizarse en la zona de la planta (bloque o conjunto de variables) donde se ha producido el fallo, para diagnosticarle y en su caso poner los medios para corregir el fallo, y volver al comportamiento normal.

Un aspecto a destacar es el hecho de que el bloque 2, que en proporción contiene la mayoría de las variables, es el que mejor detecta los errores. Aunque concretamente no haya detectado los errores IDV (7) e IDV (17), se puede observar que su valor medio del número de alarmas para todos los errores es el más elevado.



Número de Fallo	Número de alarmas: Estadístico Q						
	B1	B2	B3	B4	B5	B6	B7
IDV (1)	98.75	99.63	20.50	12.38	40.50	22.50	43.75
IDV (2)	1.88	98.38	86.13	2.38	94.63	52.75	20.75
IDV (3)	0.50	1.13	2.00	1.38	1.00	1.13	1.00
IDV (4)	0.88	1.63	1.00	100.00	1.00	1.00	0.88
IDV (5)	9.50	22.75	98.38	9.88	15.38	14.13	3.63
IDV (6)	100.00	99.50	98.00	94.88	99.25	98.75	88.38
IDV (7)	20.25	32.88	23.88	20.13	25.75	23.38	100.00
IDV (8)	54.38	97.63	51.13	35.38	77.50	83.25	48.38
IDV (9)	0.75	1.88	1.38	1.63	0.63	0.75	1.13
IDV (10)	6.13	58.88	11.88	3.13	6.13	9.25	1.25
IDV (11)	1.13	1.75	0.63	84.63	0.50	0.13	1.13
IDV (12)	41.13	98.88	73.75	65.25	69.25	62.50	29.38
IDV (13)	42.88	95.13	72.75	52.00	90.63	79.50	53.25
IDV (14)	1.00	0.63	0.63	100.00	1.00	0.25	1.13
IDV (15)	1.50	2.50	0.63	1.88	1.75	0.25	1.38
IDV (16)	1.88	58.50	5.88	1.50	2.50	4.00	1.75
IDV (17)	1.00	7.13	2.25	96.38	1.25	1.13	2.00
IDV (18)	82.63	90.13	87.25	85.50	87.25	85.63	84.25
IDV (19)	0.75	2.63	24.63	6.25	2.13	1.00	1.25
IDV (20)	2.38	43.25	43.50	2.75	29.50	1.00	2.00
IDV (21)	0.63	44.25	5.38	1.13	3.63	0.75	0.25
Media Aritmética	22.38	45.67	33.88	37.07	31.01	25.86	23.18

*Tabla 6: Número de alarmas para el estadístico Q, analizando los 21 errores en cada uno de los bloques del autoencoder distribuido cuando el umbral límite de relevancia es igual a la media*

### Índice de inferencia Bayesiana (BIC)

Una vez conocido el comportamiento específico de cada red entrenada para cada bloque del autoencoder distribuido, es necesario establecer una relación y obtener parámetros estadísticos generales para compararlos con los modelos previamente analizados (PCA y autoencoder general). Para ello, se utilizará el método BIC.

Este método requiere los datos de los estadísticos  $T^2$  y  $Q$ , y sus respectivos umbrales calculados para cada error en cada bloque.

Una vez se cargan en el método estos valores, habrá que definir un nivel de confianza  $\alpha$  bajo el cual se realizan los cálculos basados en probabilidades definidos en el apartado 2.3.5. de este documento. En este caso se definirá un nivel de confianza  $\alpha = 0.9$

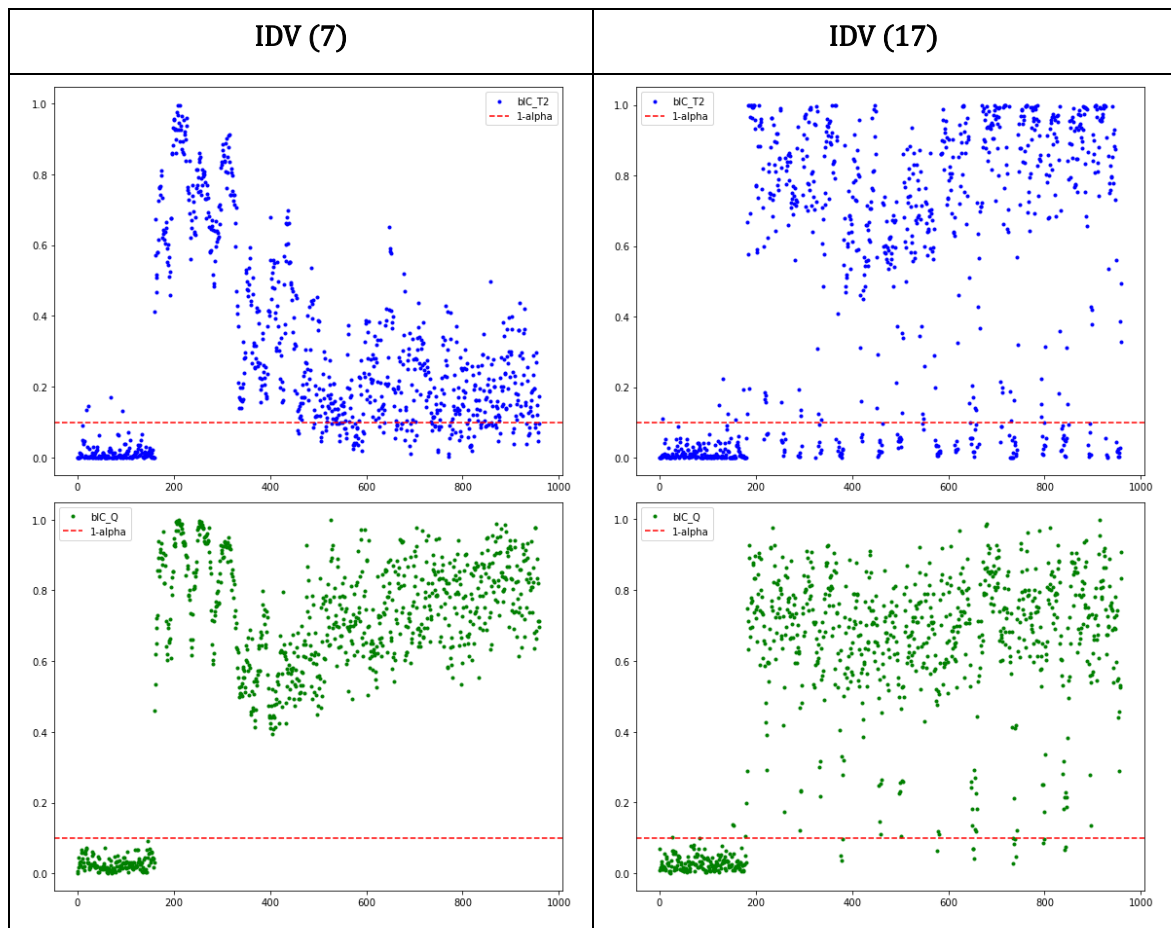


Figura 22: Representación de estadísticos BIC para fallos IDV (7) e IDV (17) estudiados mediante Autoencoder distribuido cuando el umbral límite de relevancia es igual a la media

Número de Fallo	Estadístico T2-BIC			Estadístico Q-BIC		
	Tiempo de detección	Número de alarmas	Falsas alarmas	Tiempo de detección	Número de alarmas	Falsas alarmas
IDV (1)	168	94.25	3.13	164	99.63	3.75
IDV (2)	95	98.25	8.13	174	98.63	1.88
IDV (3)	50	5.00	8.75	960	3.38	2.50
IDV (4)	166	52.50	4.38	160	100.00	2.50
IDV (5)	175	23.00	4.38	160	100.00	2.50
IDV (6)	170	98.75	3.75	160	100.00	1.88
IDV (7)	160	86.50	2.50	160	100.00	0.00
IDV (8)	185	96.88	1.88	180	97.50	0.63
IDV (9)	575	4.88	5.00	960	3.88	1.25
IDV (10)	244	33.75	3.75	207	55.38	2.50
IDV (11)	170	44.13	4.38	165	80.88	1.88
IDV (12)	181	95.13	3.75	167	98.63	0.63
IDV (13)	208	94.13	1.25	202	95.00	0.00
IDV (14)	163	87.00	1.25	160	100.00	0.00
IDV (15)	836	4.75	0.63	960	4.00	1.25
IDV (16)	353	21.00	13.13	175	51.50	1.25
IDV (17)	182	79.38	3.13	181	95.50	2.50
IDV (18)	252	89.00	4.38	243	90.25	2.50
IDV (19)	344	15.25	0.63	960	21.75	0.63
IDV (20)	249	33.75	3.13	243	46.50	3.13
IDV (21)	668	39.75	4.38	644	42.63	3.13
Media Aritmética		57.00	4.08		70.71	1.73
Media Aritmética sin fallos IDV (3) (9) (15)		65.69	3.96		81.88	1.74

Tabla 7: Parámetros estadísticos obtenidos para los 21 errores utilizando el método de Autoencoder distribuido cuando el umbral límite de relevancia es igual a la media

*Juan Domínguez González (2024) - EII - Universidad de Valladolid*

Como resultado de este método estadístico se obtienen unos vectores de comportamiento de ambos estadísticos, en los que se interrelaciona la información proveniente de la aplicación de los autoencoders por bloques.

Se muestran en la Figura 22, las representaciones de los estadísticos BIC, para los errores IDV (7) e IDV (17).

De manera análoga al análisis de los estadísticos según los otros métodos se obtienen las características principales de la detección de cada fallo, tiempo de detección, número de alarmas y porcentaje de falsas alarmas. Incluyéndolas en la tabla siguiente (Tabla 7).

La detección del error IDV (7) mediante el estadístico  $T^2$  empeora considerablemente con el uso de un autoencoder distribuido en comparación con el autoencoder general. En cambio, la exactitud en la detección del fallo mediante el estadístico  $Q$  se mantiene. Para el fallo IDV (17), los comportamientos son similares para ambos estadísticos al comparar el autoencoder distribuido con el general, reflejando la tendencia general en la detección de fallos. Aunque el tiempo de detección se reduce y el número de alarmas aumenta, mejorando la calidad de la detección, el incremento en el porcentaje de falsas alarmas tiene un impacto negativo en este proceso.

#### 4.4.2. Umbral igual a la media de las relevancias ampliada un 50%

Aplicando este valor del umbral se obtienen 8 bloques de variables, encontrándose en cada uno de los bloques las variables siguientes:

- **B1:** 1, 14, 15, 40, 44, 45, 49
- **B2:** 7, 8, 10, 11, 13, 16, 18, 19, 22, 25, 31, 35, 43, 47, 50
- **B3:** 5, 17, 46, 52
- **B4:** 2, 9, 21, 42, 51
- **B5:** 20, 27, 28, 33, 36
- **B6:** 4, 6, 23, 24, 38, 39, 41
- **B7:** 3, 29, 30, 34, 37
- **B8:** 12, 26, 32, 48

*Juan Domínguez González (2024) - EII - Universidad de Valladolid*

Definidos cada uno de los 8 bloques como un conjunto de variables, se entrena un autoencoder para cada uno de ellos, estudiando sus estadísticos y guardando los parámetros necesarios para la detección de fallos.

Tras sucesivos ensayos variando la posible estructura de cada uno de los autoencoders se define el número de neuronas por capa para la capa de entrada, las tres capas intermedias y la capa de salida de cada bloque de la siguiente manera:

- **B1:** 7-5-3-5-7
- **B2:** 15-10-5-10-15
- **B3:** 4-3-2-3-4
- **B4:** 5-4-2-4-5
- **B5:** 5-4-2-4-5
- **B6:** 7-5-3-5-7
- **B7:** 5-4-2-4-5
- **B8:** 4-3-2-3-4

### Detección de fallos

Se realiza una detección de fallos iterativa, para analizar cada bloque en función de su propio autoencoder entrenado. Se obtienen los valores de sus estadísticos y los parámetros que los caracterizan. Centrándose de nuevo en los valores del número de alarmas detectadas para el estadístico Q. Ya que se trata del valor más representativo en el reconocimiento del error.

Se refleja en una misma tabla (Tabla 8) los distintos valores de dicho elemento para cada fichero de fallo y cada bloque perteneciente al autoencoder distribuido.

Analizando la Tabla 8, de nuevo se puede observar que en este caso la detección de los fallos en cada bloque también es distinta y por lo general reducida. El fallo IDV (7) solo se detecta en el primer bloque y el IDV (17) solo se detecta en el cuarto, que coincide con el caso anterior, lo que nos permite centrarnos solo en las variables de dichos bloques para diagnosticar donde está el fallo.

De nuevo el bloque 2, que en proporción contiene la mayoría de las variables, es el que mejor detecta los errores. Sin embargo, el bloque 8, no es capaz de detectar correctamente ninguno de los errores.

Número de Fallo	Número de alarmas: Estadístico Q							
	B1	B2	B3	B4	B5	B6	B7	B8
IDV (1)	98.75	90.38	19.88	11.50	32.25	98.00	42.13	0.5
IDV (2)	9.88	98.50	52.38	1.63	94.00	72.38	98.63	1
IDV (3)	0.50	1.75	1.88	1.38	0.38	1.50	1.63	2
IDV (4)	0.75	1.13	1.13	100.00	1.00	1.00	1.38	0.75
IDV (5)	8.50	23.50	98.13	10.50	15.63	17.13	14.88	0.75
IDV (6)	100.00	99.50	97.50	95.25	99.38	98.50	98.75	4.75
IDV (7)	100.00	33.63	24.38	19.38	25.63	27.63	26.88	1.625
IDV (8)	50.88	97.75	51.25	32.50	72.00	87.13	89.75	1
IDV (9)	0.38	1.38	0.75	1.38	1.00	1.25	1.00	0.875
IDV (10)	1.38	63.75	11.38	2.88	6.50	9.75	8.63	1.5
IDV (11)	0.88	2.13	1.00	83.63	0.25	0.50	1.38	1.5
IDV (12)	44.25	98.88	71.50	67.75	68.75	70.13	58.88	3
IDV (13)	49.13	95.13	66.25	55.00	86.88	77.38	70.38	11.25
IDV (14)	1.63	0.75	0.63	100.00	0.75	1.00	0.63	0.875
IDV (15)	0.75	1.88	0.63	1.63	1.50	0.88	0.88	0.5
IDV (16)	0.75	60.88	7.00	1.50	1.63	4.88	2.75	1.25
IDV (17)	1.88	6.88	3.13	96.25	2.13	0.63	1.50	2.125
IDV (18)	86.00	90.25	87.00	85.75	87.25	87.38	86.25	68.5
IDV (19)	1.00	2.50	26.00	7.13	1.75	1.25	1.13	0.75
IDV (20)	1.13	43.13	44.13	3.13	30.38	1.13	2.50	0.375
IDV (21)	0.13	45.50	5.25	1.13	3.25	0.38	1.13	0
Media Aritmética	26.60	45.67	31.96	37.11	30.11	31.42	29.10	4.99

Tabla 8: Número de alarmas para el estadístico Q, analizando los 21 errores en cada uno de los bloques del autoencoder distribuido cuando el umbral es igual a la media de las relevancias ampliada un 50%

### Índice de inferencia Bayesiana (BIC)

Conocida la capacidad de detección de fallos de cada red, se establece una relación entre todas ellas con el método BIC. Se define de nuevo un nivel de confianza  $\alpha = 0.9$ .

Obteniéndose los vectores de comportamiento de los estadísticos BIC y sus representaciones (Figura 23). Como siempre, este estudio toma como ejemplo los fallos IDV (7) e IDV (17).

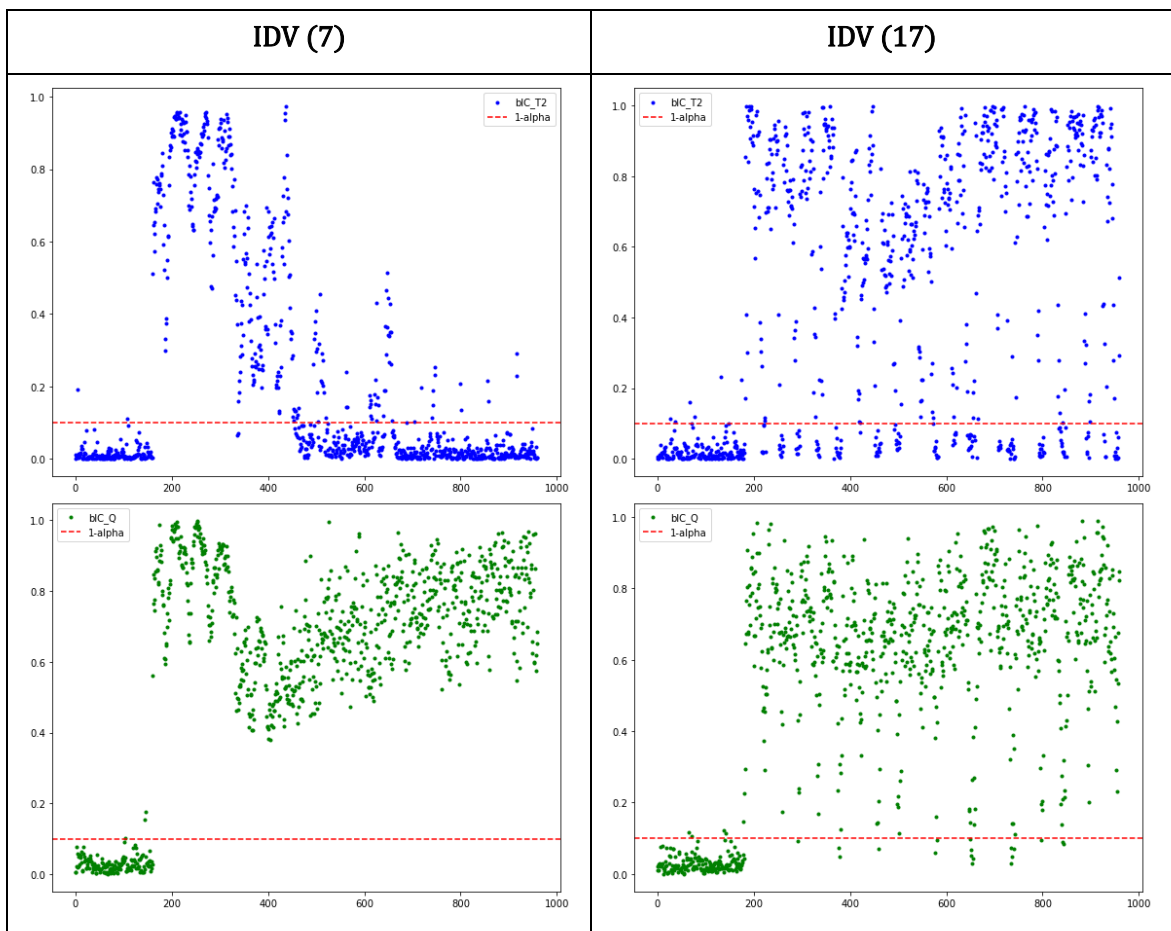


Figura 23: Representación de estadísticos BIC para fallos IDV (7) e IDV (17) estudiados mediante Autoencoder distribuido cuando el umbral es igual a la media de las relevancias ampliada un 50%

Número de Fallo	Estadístico T2-BIC			Estadístico Q-BIC		
	Tiempo de detección	Número de alarmas	Falsas alarmas	Tiempo de detección	Número de alarmas	Falsas alarmas
IDV (1)	166	99.25	4.38	164	99.50	1.88
IDV (2)	95	98.13	5.63	174	98.75	2.50
IDV (3)	200	6.88	2.50	960	4.13	1.88
IDV (4)	384	35.38	3.75	160	100.00	3.13
IDV (5)	160	24.38	3.75	160	100.00	3.13
IDV (6)	170	99.00	5.00	160	100.00	1.25
IDV (7)	160	45.38	1.25	160	100.00	1.88
IDV (8)	30	97.13	6.25	181	97.38	1.88
IDV (9)	960	5.00	9.38	960	2.88	1.25
IDV (10)	216	37.25	3.13	190	60.75	3.13
IDV (11)	206	33.63	8.75	165	80.63	1.25
IDV (12)	181	96.75	0.63	162	99.00	0.63
IDV (13)	208	94.13	1.25	202	95.00	0.00
IDV (14)	163	84.63	3.75	160	100.00	0.00
IDV (15)	755	8.00	1.25	960	3.25	2.50
IDV (16)	138	21.88	13.75	172	58.00	1.25
IDV (17)	182	77.13	3.75	181	95.25	2.50
IDV (18)	247	89.63	5.00	243	90.63	2.50
IDV (19)	960	21.00	1.25	960	23.88	0.00
IDV (20)	244	44.25	1.25	243	45.50	2.50
IDV (21)	676	36.13	1.25	644	42.50	1.25
Media Aritmética		54.99	4.14		71.29	1.73
Media Aritmética sin fallos IDV (3) (9) (15)		63.06	4.10		82.60	1.70

Tabla 9: Parámetros estadísticos obtenidos para los 21 errores utilizando el método de Autoencoder distribuido cuando el umbral es igual a la media de las relevancias ampliada un 50%



*Juan Domínguez González (2024) - EII - Universidad de Valladolid*

De manera análoga al análisis de los estadísticos según los otros métodos se obtienen las características principales de la detección de cada fallo, tiempo de detección, número de alarmas y falsas alarmas (Tabla 9).

El error IDV (7) deja de detectarse mediante el estadístico  $T^2$  frente a los casos anteriores. Mientras que el estadístico Q, mantiene la exactitud en el reconocimiento del fallo.

La detección general del estadístico  $T^2$  empeora frente al caso anterior, mientras que la del estadístico Q mejora ligeramente.

#### **4.4.3. Umbral igual a la media de las relevancias reducida un 50%**

Aplicando este valor del umbral se obtienen 5 bloques de variables, encontrándose en cada uno de los bloques las variables siguientes:

- **B1:** 1, 2, 9, 14, 39, 44, 51
- **B2:** 3, 4, 6, 7, 8, 10, 11, 13, 16, 18, 19, 20, 21, 22, 25, 29, 31, 33, 35, 37, 41, 43, 47, 50
- **B3:** 5, 17, 34, 42, 46, 52
- **B4:** 15, 23, 25, 26, 28, 30, 36, 38, 40, 45, 49
- **B5:** 12, 24, 27, 32, 48

Definidos cada uno de los 5 bloques como un conjunto de variables, se entrena un autoencoder para cada uno de ellos, estudiando sus estadísticos y guardando los parámetros necesarios para la detección de fallos.

Tras sucesivos ensayos variando la posible estructura de cada uno de los autoencoders se define el número de neuronas por capa para la capa de entrada, las tres capas intermedias y la capa de salida de cada bloque de la siguiente manera:

- **B1:** 7-4-2-4-7
- **B2:** 24-13-7-13-24
- **B3:** 6-4-2-4-6
- **B4:** 11-7-4-7-11
- **B5:** 5-4-2-4-5

### Detección de fallos

Se realiza de nuevo la detección de fallos iterativa, para analizar cada bloque en función de su propio autoencoder entrenado, obteniendo los valores de los estadísticos y sus parámetros. Se refleja en la Tabla 10 los valores del estadístico Q para cada fichero de fallo y cada bloque

Número de Fallo	Número de alarmas: Estadístico Q				
	B1	B2	B3	B4	B5
IDV (1)	98.75	99.88	46.50	61.50	7.63
IDV (2)	70.63	98.50	94.38	98.25	16.25
IDV (3)	1.25	1.13	1.88	0.88	0.88
IDV (4)	100.00	1.13	1.25	0.88	0.50
IDV (5)	8.25	24.00	98.13	14.88	6.63
IDV (6)	100.00	99.50	99.00	98.38	35.25
IDV (7)	20.63	33.38	26.75	100.00	14.63
IDV (8)	55.63	97.88	69.63	90.13	51.63
IDV (9)	0.50	1.25	1.50	1.13	1.25
IDV (10)	2.75	50.13	9.38	6.50	1.50
IDV (11)	78.63	1.13	0.75	0.88	1.25
IDV (12)	56.75	98.88	75.75	74.13	34.50
IDV (13)	53.25	95.25	83.50	91.13	58.63
IDV (14)	88.50	82.13	0.63	0.88	0.50
IDV (15)	1.75	2.50	1.25	0.88	1.63
IDV (16)	1.00	43.50	4.00	3.13	1.75
IDV (17)	68.25	83.00	1.13	1.50	0.63
IDV (18)	84.25	90.00	87.25	87.50	83.13
IDV (19)	2.38	1.25	26.50	0.88	2.38
IDV (20)	1.63	44.00	43.38	1.25	1.25
IDV (21)	0.38	49.00	2.75	2.50	1.38
Media Aritmética	42.63	52.26	36.92	35.10	15.39

Tabla 10: Número de alarmas para el estadístico Q, analizando los 21 errores en cada uno de los bloques del autoencoder distribuido cuando el umbral es igual a la media de las relevancias reducida un 50%

Analizando la Tabla 10, se puede observar que la detección del fallo IDV (7), de nuevo cambia de bloque, esta vez se detecta en el bloque 4. Sin embargo, en esta ocasión ninguno de los bloques realiza una buena detección del fallo IDV (17). Ya que el número de alarmas más elevado es 83%, correspondiente con el del bloque 2, valor que se considera suficiente como para decir que se ha detectado el fallo.

Al igual que en los casos anteriores, el bloque 2 es el que mejor detecta, sin embargo, aunque en este caso contenga prácticamente la mitad de las variables, no llega a realizar una detección más precisa que en otros casos.

### Índice de inferencia Bayesiana (BIC)

Conocida la capacidad de detección de fallos de cada red, se establece una relación entre todas ellas con el método BIC. Se define de nuevo un nivel de confianza  $\alpha = 0.9$ . Obteniéndose los vectores de comportamiento de los estadísticos BIC y sus representaciones (Figura 24). Como siempre, este estudio toma como ejemplo los fallos IDV (7) e IDV (17).

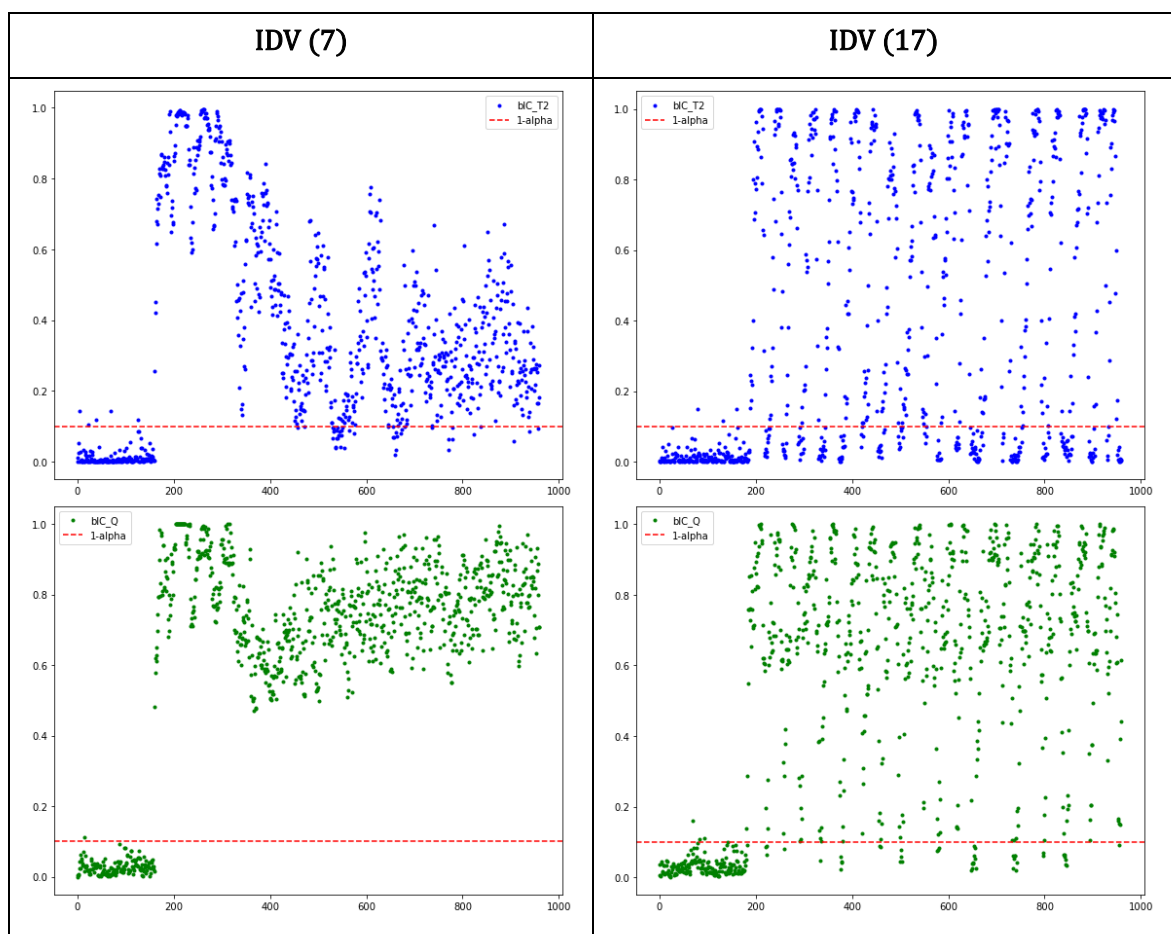


Figura 24: Representación de estadísticos BIC para fallos IDV (7) e IDV (17) estudiados mediante Autoencoder distribuido cuando el umbral es igual a la media de las relevancias reducida un 50%

Número de Fallo	Estadístico T2-BIC			Estadístico Q-BIC		
	Tiempo de detección	Número de alarmas	Falsas alarmas	Tiempo de detección	Número de alarmas	Falsas alarmas
IDV (1)	168	99.00	0.63	162	99.75	1.25
IDV (2)	174	98.38	0.63	172	98.50	1.88
IDV (3)	960	3.88	2.50	960	2.13	1.88
IDV (4)	166	50.50	3.75	160	100.00	0.00
IDV (5)	174	24.00	3.75	160	100.00	0.00
IDV (6)	170	99.13	2.50	160	100.00	0.63
IDV (7)	160	95.00	3.13	160	100.00	0.63
IDV (8)	188	96.00	0.63	178	97.75	0.00
IDV (9)	960	3.25	2.50	960	2.75	0.00
IDV (10)	248	32.38	1.25	206	42.25	1.88
IDV (11)	170	50.88	0.63	165	74.88	0.63
IDV (12)	181	97.38	1.88	162	98.75	1.25
IDV (13)	223	92.25	1.25	197	95.63	0.63
IDV (14)	165	91.88	1.88	160	100.00	1.25
IDV (15)	835	5.25	1.88	960	2.50	2.50
IDV (16)	145	16.63	10.63	279	37.25	0.00
IDV (17)	188	70.75	1.25	183	91.75	1.88
IDV (18)	254	89.13	3.13	243	89.75	0.00
IDV (19)	769	7.50	1.25	960	20.25	0.00
IDV (20)	244	44.25	1.25	243	45.38	1.88
IDV (21)	687	30.88	1.88	654	46.00	2.50
Media Aritmética		57.06	2.29		68.82	0.98
Media Aritmética sin fallos IDV (3) (9) (15)		65.88	2.29		79.88	0.90

Tabla 11: Parámetros estadísticos obtenidos para los 21 errores utilizando el método de Autoencoder distribuido cuando el umbral es igual a la media de las relevancias reducida un 50%

*Juan Domínguez González (2024) - EII - Universidad de Valladolid*

De manera análoga al análisis de los estadísticos según los otros métodos se obtienen las características principales de la detección de cada fallo, tiempo de detección, número de alarmas y falsas alarmas (Tabla 11).

La detección del fallo IDV (7) mediante el estadístico T2 se recupera frente al caso anterior. Mientras que el estadístico Q, mantiene la exactitud en el reconocimiento del fallo

En cuanto al fallo IDV (17), su detección en ninguno de los métodos ha sido perfecta, pero en este caso se reduce hasta un valor de 91,75% del número de alarmas para el estadístico Q. Valor que puede considerarse adecuado para la detección.

La detección general del estadístico T<sup>2</sup> mejora frente al caso anterior, mientras que la del estadístico Q empeora ligeramente.

# CAPÍTULO V: CONCLUSIONES Y TRABAJO FUTURO

## 5.1. CONCLUSIONES

### Definición del mejor método

Se han explorado en este documento 5 métodos distintos de análisis y detección de fallos, para la misma planta (Tennessee Eastman). Como se indicaba al comienzo, el objetivo principal es la observación de que método de los estudiados proporcionaba una detección más precisa.

Para esta labor es necesario un análisis más exhaustivo de los parámetros calculados en cada caso a partir de los valores de los estadísticos para los ficheros de fallo. De esta manera, en las tablas 4, 5, 7, 9 y 11, se han reflejado estos valores para su análisis.

Además, en cada caso se representaba un valor medio del número de alarmas y del porcentaje de falsas alarmas. Tanto para un estudio completo de los 21 ficheros de fallo, como para un caso más concreto en el que se excluían los fallos IDV (3), IDV (9) e IDV (15). Los cuales se sabe que no se detectan por ninguno de estos métodos.

Conocida esta información se construye la Tabla 12, que recoge los valores medios del número de alarmas y el porcentaje de falsas alarmas de cada método estudiado, para el caso en el que se excluyen los errores IDV (3), IDV (9) e IDV (15).

Método	Estadístico T2		Estadístico Q	
	Número de alarmas	Falsas alarmas	Número de alarmas	Falsas alarmas
PCA	67.58	1.18	71.24	0.31
Autoencoder General	60.29	0.38	79.82	0.66
Autoencoder Distribuido (umbral=media)	65.69	3.96	81.88	1.74
Autoencoder Distribuido (umbral=media*1.5)	63.06	4.10	82.60	1.70
Autoencoder Distribuido (umbral=media*0.5)	65.88	2.29	79.88	0.90

*Tabla 12: Valores medios excluyendo los fallos IDV (3), IDV (9) e IDV (15), para cada método estudiado*

El método más acertado será aquel que logre valores más cercanos al 100% en el número de alarmas y al 0% en falsas alarmas. En ocasiones, se ha observado que algunos métodos logran una detección “perfecta” de ciertos errores utilizando determinados estadísticos. Por ejemplo, el fallo IDV (7) fue detectado perfectamente mediante el método de PCA (Tabla 4) para ambos estadísticos, y el fallo IDV (14) fue detectado perfectamente mediante el Autoencoder Distribuido con una media aumentada un 50%, únicamente para el estadístico Q.

Esta detección perfecta implica que los valores del estadístico son todos menores que el umbral hasta la observación 160, momento en el cual teóricamente ocurre el fallo. A partir de ahí, todos los valores son superiores al umbral, como se muestra en la Figura 17 para el fallo IDV (7).

Sin embargo, este tipo de detección no es común en el estudio realizado. Por lo tanto, es necesario analizar los valores medios de detección de cada método, ya que la detección de errores puntuales no es representativa para el estudio completo de la planta.

Según la Tabla 12, la mejor detección con el estadístico  $T^2$  se logra mediante el método de PCA, ya que tiene el número de alarmas más alto. Aunque el método de Autoencoder General tiene un porcentaje de falsas alarmas mejor, su número de alarmas es considerablemente inferior.

Para los Autoencoders Distribuidos con el estadístico  $T^2$ , el análisis muestra un número de alarmas similar, siendo la detección mejor para los umbrales más pequeños. El caso con la media reducida en un 50% es el mejor, ya que presenta el menor porcentaje de falsas alarmas.

A pesar de estos resultados, ningún método muestra valores de detección confiables con el estadístico  $T^2$  debido al bajo número de alarmas.

El estadístico Q ofrece un mejor análisis para todos los métodos. En la Tabla 12, se observa que el número de alarmas es mayor para cada método cuando se analiza con Q, en comparación con  $T^2$ . Además, el porcentaje de falsas alarmas es menor con el estadístico Q. Por lo tanto, se considera el estadístico Q como más representativo para este estudio.

Al analizar el comportamiento del estadístico Q para cada método, se puede concluir que cuanto más complejo es el método de estudio, mejor es la detección. Aunque el método PCA tiene el porcentaje de falsas alarmas más bajo, también tiene el menor número de alarmas. El Autoencoder General detecta mejor que el PCA, pero es superado por el Autoencoder Distribuido, que realiza la mejor detección.

Entre las opciones de Autoencoder Distribuido estudiadas, los dos casos con umbral más grande presentan un porcentaje de falsas alarmas más elevado,





*Juan Domínguez González (2024) - EII - Universidad de Valladolid*

pero también un mayor número de alarmas, mejorando la detección en general.

Basándose en esta información y en los datos de la Tabla 12, se concluye que el método de Autoencoder Distribuido con una media incrementada en un 50%, mediante el estadístico Q, es el mejor método de detección del estudio completo.

### Justificación del método seleccionado

El método de Autoencoder Distribuido con media incrementada en un 50% presenta numerosos aspectos positivos en comparación con otros métodos, justificando su elección.

Como una opción de Autoencoder Distribuido, este método permite realizar un análisis más detallado, destacando la influencia individual de cada variable sobre el proceso completo.

El uso de un umbral más elevado implica que la relación entre las variables debe ser mayor para agruparse en el mismo bloque, lo que resulta en un mayor número de bloques (8 en este caso). Esto aumenta la especificidad de la influencia de cada variable.

Un posible problema con este método es que la interrelación entre los estadísticos obtenidos de las 8 redes entrenadas podría no ser óptima. Sin embargo, se ha demostrado que el método de inferencia Bayesiana cumple completamente con los requisitos del estudio.

La programación de este método no es significativamente más compleja que la del Autoencoder General. Ambos requieren una función para entrenar la red neuronal con datos de comportamiento normal y un archivo para detectar fallos mediante esa red. La principal diferencia es que, en el Autoencoder Distribuido, este proceso es iterativo para cada bloque de variables. El código es prácticamente el mismo, aunque requiere más espacio de memoria para almacenar los valores de cada iteración.

Aunque requiere entrenar 8 redes, el tiempo de entrenamiento de cada red es menor debido a que contienen menos variables. Por tanto, el tiempo total no es significativamente mayor que el de una única red con 52 variables.

En la práctica, el uso de un autoencoder distribuido implica que la detección de valores de variables se realiza mediante sensores diferenciados para cada bloque, lo que supone un mayor coste económico, pero también una mayor precisión en las mediciones.

Además, este enfoque ofrece mayor versatilidad ante fallos de los sensores. En un análisis general de las 52 variables, un fallo en el sensor requeriría detener el proceso de detección y reiniciarlo tras la reparación. Sin embargo, en un análisis por bloques, un fallo en un sensor no interrumpe todo el proceso; simplemente se continúa sin ese bloque de variables, proporcionando resultados razonablemente fiables.

## 5.2. TRABAJO FUTURO

El estudio mediante una red neuronal como es el caso del Autoencoder tiene muchas variables, lo positivo de que existen muchos procesos que pueden solaparse haciendo el estudio mucho más completo cada vez.

En este caso se ha estudiado y comprobado la mayor exactitud en la detección de fallos para un Autoencoder Distribuido. Sin embargo, se ha partido del concepto de una red autoencoder como una figura “simple” dentro de las posibilidades existentes.

Se encuentran variaciones del autoencoder a las cuales se les puede aplicar el concepto de autoencoder distribuido. Los ejemplos más relevantes son los siguientes:

- **Autoencoders variacionales (VAE)**, se componen de un codificador que mapea datos de entrada a una distribución latente, y un decodificador que reconstruye los datos originales a partir de muestras de esta distribución. Los VAE se entrenan maximizando la evidencia variacional inferior (ELBO), que incluye un término de reconstrucción y una regularización KL para mantener la distribución latente cercana a una normal estándar. Estos modelos permiten generar nuevos datos, explorar el espacio latente, comprimir datos y detectar anomalías. Sin embargo, pueden ser más complejos de entrenar y, en algunos casos, los datos generados no son tan realistas como los producidos por otros modelos como las GANs. [23]
- **Autoencoders recurrentes (RAE)**, son una variante de los autoencoders diseñados para procesar datos secuenciales, combinando las capacidades de las redes neuronales recurrentes (RNN) con las de los autoencoders tradicionales. Un RAE consta de un codificador recurrente que convierte una secuencia de entrada en un vector de estado oculto, y un decodificador recurrente que reconstruye la secuencia original a partir de este vector latente. Son útiles para tareas como la compresión de secuencias, el modelado de series temporales, el análisis de texto y la detección de anomalías en secuencias. Sin

*Juan Domínguez González (2024) - EII - Universidad de Valladolid*

embargo, pueden ser más difíciles de entrenar y requieren más datos y tiempo de cómputo comparado con los autoencoders tradicionales. [24]

- **Denoising autoencoders (DAE)**, son variantes de los autoencoders diseñadas para aprender representaciones robustas de los datos mediante la eliminación de ruido en las entradas. Consisten en un codificador que transforma entradas ruidosas en una representación latente y un decodificador que reconstruye la entrada original. Durante el entrenamiento, se añade ruido controlado a las entradas para que el modelo aprenda a eliminarlo, mejorando así la calidad de las reconstrucciones. [25]

El proceso de trabajo a futuro podría partir por la aplicación del concepto de autoencoder distribuido a cualquiera de estos tipos de autoencoders. Favoreciendo en cualquiera de los casos el estudio más exhaustivo de la planta Tennessee Eastman y la posibilidad de obtención de mejores resultados en la detección de fallos.

# BIBLIOGRAFÍA

[1] Flórez Ramírez, N. & Flórez Rendon, A. L. & Cogollo Flórez, J.M. (2019) “Notas de control estadístico de calidad” Editorial Universitaria.

[2] Arias Coello, A, “Gestión de la calidad” Universidad Complutense de Madrid, recuperado en: <https://webs.ucm.es/centros/cont/descargas/documento10123.pdf> [Último acceso: 19 Abril 2024].

[3] Cabezón Gutiérrez, S. (2014) “Control de calidad en la producción industrial” [Trabajo de fin de grado, Universidad de Valladolid].

[4] Besterfield, D.H. (2009) “Control de calidad” Octava Edición, Southern Illinois University, recuperado en: <https://cmapspublic3.ihmc.us/rid=1Y2G0F7VH-1RQJ94G-CVP/Control%20de%20Calidad%20H.%20Besterfield.pdf>

[5] Measure control, “La Historia del SPC”, recuperado en: <https://measurecontrol.com/la-historia-del-spc/> [Último acceso: 19 Abril 2024]

[6] Carro Paz, R. & González Gómez, D. “Control Estadístico de Procesos”, Universidad Nacional de Mar del Plata, recuperado en: [https://nulan.mdp.edu.ar/id/eprint/1617/1/12\\_control\\_estadistico.pdf](https://nulan.mdp.edu.ar/id/eprint/1617/1/12_control_estadistico.pdf)

[7] La librería del conocimiento, “Control Estadístico de los Procesos (SPC)”, recuperado en <https://lalibriadelconocimiento.blogspot.com/2016/10/control-estadistico-de-los-procesos.html> [Último acceso: 17 abril 2024]

[8] Curiel Alonso, J. (2022) “Mejora de la calidad de un proceso mediante la detección de anomalías basadas en inteligencia computacional” [Trabajo de fin de grado, Universidad de Valladolid].

[9] Fuente, M.J. & García-Alvarez, D. (2011), “Estudio comparativo de técnicas de detección de fallos basadas en el Análisis de Componentes Principales (PCA)”. Revista Iberoamericana de Automática e Informática industrial, Vol.8.

[10] Amat Rodrigo, J. (2017), “Análisis de Componentes Principales (Principal Component Analysis, PCA) y t-SNE” [https://cienciadedatos.net/documentos/35\\_principal\\_component\\_analysis](https://cienciadedatos.net/documentos/35_principal_component_analysis) [Último acceso: 15 abril 2024]

[11] González, A. “¿Que es Machine Learning?” Cleverdata, recuperado en: <https://cleverdata.io/que-es-machine-learning-big-data/> [Último acceso: 19 abril 2024]

[12] Bobadilla, J. (2020), “Machine Learning y Deep Learning: Usando Python, Scikit y Keras” Ediciones de la U. 621.39 ed., pp: 13-19.

[13] Izaurieta, F. & Saavedra, C. (2000) “Redes Neuronales Artificiales” Departamento de Física, Universidad de Concepción, Chile.

[14] Revista Ingeniería y tecnología, (2021), Unlr <https://www.unir.net/ingenieria/revista/redes-neuronales-artificiales/> [Último acceso: 22 abril 2024]

[15] Cash, S. & Yuste, R. “Linear Summation of Excitatory Inputs by CA1 Pyramidal Neurons,” Neuron, vol. 22, n<sup>a</sup> 2, pp: 383–394.

[16] Calvo, D. (2018), “Perceptrón Multicapa – Red Neuronal”, recuperado en: <https://www.diegocalvo.es/perceptron-multicapa/> [Último acceso: 20 abril 2024]

[17] Dertat, A. (2017), “Applied Deep Learning – Part 3: Autoencoders” recuperado en: <https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798#f686> [Último acceso: 07 mayo 2024]

*Juan Domínguez González (2024) - EII - Universidad de Valladolid*

- [18] Li, Z. & Tian, L. & Jiang, Q. & Yan, X. (2021), “Distributed ensemble stacked autoencoder model for non-linear process monitoring”, *Information Sciences*, nº 542, pp: 302-316.
- [19] Xu, C. & Zhao, S. & Liu, F. (2017) “Distributed plant-wide process monitoring based on PCA with minimal redundancy maximal relevance”. *Chemometrics and Intelligent Laboratory Systems*, nº 169, pp: 53-63.
- [20] Downs, J.J. & Vogel, E.F (1993), “A plant-wide industrial process control Problem”, *Computer Chemical Engineering*, nº 17, pp: 245-255.
- [21] Álvarez Toro, L.A. & Sotomayor, O. (2018) “Integración Optimización Control predictivo y aplicación a la planta Tennessee Eastman”, *Rev. Per. Quím. Ing. Quím.* vol. 11 nº 1.
- [22] Yin, S. & Ding, S.X. & Haghani, A. & Hao, H. & Zhang, P. (2012) “A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process,” *J. Process Control*, vol. 22, nº. 9, pp: 1567-1581.
- [23] Kingma, D. P., & Welling, M. (2014). “Auto-Encoding Variational Bayes” *arXiv*, nº1312.6114, vol. 11.
- [24] Olah, C. (2015). “Understanding LSTM Networks” Colah's Blog. recuperado en <https://colah.github.io/posts/2015-08-Understanding-LSTMs/> [Último acceso: 17 junio 2024]
- [25] Vincent, P. & Larochelle, H. & Bengio, Y. & Manzagol, P. A. (2008). “Extracting and Composing Robust Features with Denoising Autoencoders”, *Proceedings of the 25th International Conference on Machine Learning*, pp: 1096-1103.



## Monitorización distribuida de un proceso para mejora de su calidad mediante técnicas de inteligencia computacional/deep learning

---



*Juan Domínguez González (2024) - EII - Universidad de Valladolid*