



**Universidad de Valladolid**



UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

GRADO EN ORGANIZACIÓN INDUSTRIAL

# Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos

Autor:

González Velázquez, Miguel

Tutor:

De La Fuente Aparicio,  
María Jesús

Departamento de Ingeniería de  
Sistemas y Automática

Valladolid, noviembre 2020.



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos





## ÍNDICE DE CONTENIDOS

RESUMEN .....	6
CAPÍTULO I: INTRODUCCIÓN.....	9
1.1. INTRODUCCIÓN.....	10
1.2. OBJETIVOS.....	12
1.3. ORGANIZACIÓN.....	12
CAPÍTULO II: MARCO TEÓRICO .....	15
2.1. CONTROL DE CALIDAD .....	16
2.2. CONTROL ESTADÍSTICO DE PROCESOS .....	18
2.2.1. VARIABILIDAD DE UN PROCESO DE PRODUCCIÓN .....	19
2.2.2. CAUSAS DE LA VARIABILIDAD .....	21
2.2.3. GRÁFICOS DE CONTROL.....	21
2.2.4. ANÁLISIS DE COMPONENTES PRINCIPALES (PCA) .....	25
2.3. TÉCNICAS DE DIAGNOSTICO Y CLASIFICACIÓN DE FALLOS.....	31
2.3.1. INTELIGENCIA ARTIFICIAL, MACHINE LEARNING Y DEEP LEARNING. ....	31
2.3.2. REDES NEURONALES ARTIFICIALES (ANN). ....	32
2.3.3. REDES NEURONALES DE PROPAGACIÓN HACIA ADELANTE. ....	37
2.3.4. ENTRENAMIENTO DE LAS REDES NEURONALES ARTIFICIALES. ....	38
2.3.4.1. Sobreajuste (dropout) .....	39
2.3.5. REDES NEURONALES RECURRENTES .....	40
2.3.6. REDES NEURONALES LSTM.....	41
2.3.7. MATRIZ DE RESULTADOS. MATRIZ DE CONFUSIÓN. ....	45
CAPÍTULO III: PLANTA TENNESSEE EASTMAN .....	48
3. CASO DE ESTUDIO: PLANTA TENNESSEE EASTMAN.....	49



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos



3.1.	DESCRIPCIÓN DEL PROCESO.....	49
3.2.	DATOS DE LA PLANTA.....	51
CAPÍTULO IV: APLICACIÓN .....		57
4.1.	IMPLEMENTACIÓN PRÁCTICA .....	58
4.2.	DETECCIÓN DE FALLOS MEDIANTE TÉCNICAS DE ANÁLISIS MULTIVARIANTE. ANÁLISIS DE COMPONENTES PRINCIPALES.....	59
4.2.1.	FUERA DE LÍNEA.....	59
4.2.2.	EN LÍNEA.....	60
4.3.	DIAGNÓSTICO DE FALLOS MEDIANTE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO. 64	
4.3.1.	REDES NEURONALES DE PROPAGACIÓN HACIA ADELANTE .....	64
4.3.2.	REDES LONG SHORT-TERM MEMORY (LSTM). .....	83
CAPÍTULO V: CONCLUSIONES Y TRABAJO FUTURO.....		102
5.1.	CONCLUSIONES.....	103
5.2.	TRABAJO FUTURO.....	104
BIBLIOGRAFÍA.....		106



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos





## RESUMEN

En este trabajo se exponen diferentes técnicas relacionadas con la calidad y la monitorización de los procesos industriales. El uso de estas técnicas para la detección y diagnóstico de fallos (FDD) basadas en datos se nutren del avance de la industria y la tecnología que permiten una recogida de información de los procesos a gran escala. La nueva Industria 4.0, el big data, la utilización masiva de sensores y el control distribuido en planta permiten la aplicación de estas técnicas. Primero, para el control estadístico de procesos se aplica la técnica de Análisis de Componentes Principales (PCA) que nos permite detectar el estado de funcionamiento en un proceso industrial, y saber si su comportamiento es normal o existen fallos o anomalías mediante técnicas estadísticas. A continuación, se pretende diagnosticar el fallo mediante técnicas de aprendizaje automático aplicando redes neuronales. El creciente auge de la inteligencia artificial permite el entrenamiento de algoritmos que son capaces de identificar situaciones anómalas gracias a la experiencia adquirida. Las técnicas desarrolladas se aplican a la planta química que propone el *benchmark* Tennessee Eastman Process (TEP) obtenido de la literatura científica, donde se simula el funcionamiento de la planta en 21 tipos de fallos diferentes con los datos obtenidos para su funcionamiento normal y en estado de fallo. Se llevan a cabo simulaciones para las diferentes técnicas desarrolladas para los diferentes fallos, y se comparan los resultados obtenidos. Finalmente se realiza un breve estudio del trabajo futuro que se podría llevar a cabo para mejorar este trabajo.



## ABSTRACT

This work studies different techniques about quality and monitoring industrial process. The application of these data-based Fault Detection and Diagnosis (FDD) techniques are fueled by advancements in industry and technology that allow large-scale process information gathering. The new Industry 4.0, the big data, the massive use of sensors and distributed control in the plant allow the use of these techniques. First, for the statistical control of processes, the Principal Component Analysis (PCA) technique is applied, which allows to detect the state normal or failure in an industrial process using statistical techniques. Next, it's intended to diagnose the failures using machine learning techniques applying neural networks. The growing rise of artificial intelligence allows the train of algorithms able to identify anomalous situations thanks to the experience acquired. The techniques developed are applied to the chemical plant proposed by the Tennessee Eastman Process (TEP) benchmark obtained from the science literature, where the operation of the plant is simulated in 21 different types of faults with data from normal operation and in state of failure. Simulations are carried out for the different techniques developed for the different failures, and the results obtained are compared. Finally, a brief study is made about the future work that could be carried out to improve this work.



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos







# CAPÍTULO I: INTRODUCCIÓN



## 1.1. INTRODUCCIÓN

El control de calidad es uno de los aspectos fundamentales en la industria hoy en día, garantizar la calidad de los productos o servicios ofrecidos repercute directamente en el desarrollo, eficiencia y cuenta de resultados de una empresa. Es por ello, que desde hace unas décadas el control de los procesos productivos ha sido investigado y desarrollado por parte de la comunidad científica y las organizaciones en la búsqueda de una mejora continua.

Mediante el control de calidad se pretende asegurar que las especificaciones de un producto final se mantienen en un rango aceptable, asegurando así la calidad de este, además, se pretende monitorizar el proceso en su conjunto para su correcto funcionamiento, para la detección de fallos 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.

A lo largo de la historia los sistemas de producción han experimentado cambios vitales ocasionados por avances o desarrollos tecnológicos innovadores que han marcado un antes y un después. Como ya pasó en la primera y segunda revolución industrial, la aparición de la máquina de vapor en el siglo XVIII, o la producción en cadena en el siglo XIX los procesos de manufacturación alcanzaban una mayor calidad en sus productos, junto con una reducción en los tiempos de fabricación. Desde entonces los procesos productivos se han nutrido de las nuevas tecnologías para su mejora continua.

En el marco actual de la nueva era digital en la que nos encontramos, la “Cuarta Revolución Industrial”, también denominada Industria 4.0, marcada por la idea de la interconexión entre dispositivos, el flujo continuo de información entre máquinas y humanos [1], el IoT, el manejo de información a gran escala, la fabricación aditiva, la inteligencia artificial, etc., permiten a las empresas un enfoque basado en la automatización, la digitalización y el uso de tecnologías electrónicas y de la información para la gestión de los procesos [2].



La industria 4.0 se sustenta en la utilización de actuadores, sensores y software que trabajan interconectados entre sí, y permiten la monitorización de los procesos, y el flujo de información a través de toda la cadena de valor, y que sirve como apoyo para la gestión y toma de decisiones por parte de las organizaciones.

La seguridad en la producción es otro de los 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 estado pues presente desde hace décadas en cualquier proceso productivo, ha sido abordado desde diferentes puntos de vista y es una parte fundamental en la industria moderna y en el aseguramiento de la calidad de los procesos y productos. Entre los métodos desarrollados para abordar el problema se encuentran tres categorías, los métodos analíticos o basados en modelos, los métodos basados en datos y los métodos basados en el conocimiento, en el presente trabajo vamos a seguir un método basado en datos.

Entre las técnicas para la monitorización de los procesos, encontramos las que se basan en el control estadístico de los procesos, (SPC del inglés *statistical process control*). Técnicas como el Análisis de Componentes Principales, el Análisis de Componentes Independientes y Regresión de Mínimos Cuadrados Parciales pertenecen a la familia de técnicas para el control estadístico de procesos. En particular el Análisis de Componentes Principales (PCA) es una técnica estadística multivariante. Esta técnica 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. El PCA convierte un espacio original obtenido a partir de un histórico de datos en otro espacio de dimensionalidad reducida con casi la misma cantidad de información.

Por otro lado, entre los métodos de diagnóstico de fallos basados en datos, los métodos más usados son los de clasificación. Y dentro de estos hay muchas técnicas que pueden utilizarse, en este proyecto nos centraremos en una de ellas: las redes neuronales artificiales (ANN, Artificial Neural Networks en



inglés). Primero se realizará un estudio aplicado a las Redes Neuronales de propagación hacia adelante, el ejemplo más simple de una Red Neuronal Artificial, después, con el objetivo de comparar, y si es posible mejorar los resultados, se ha empleado un tipo especial de Red Neuronal Recurrente, una red LSTM, caracterizada por ser empleada con gran acierto mediante otros estudios en la predicción de secuencias a corto y largo plazo.

## 1.2. OBJETIVOS

Con la elaboración de este trabajo se pretende desarrollar técnicas de detección y diagnóstico basadas en datos (FDD, Fault Detection and Diagnosis en inglés). Para este cometido se ha utilizado para la detección de los fallos el análisis estadístico de procesos, y en concreto, la técnica de Análisis de Componentes Principales (PCA). Y para el diagnóstico (en este caso clasificación) de fallos se han usado algoritmos basados en la inteligencia computacional y el aprendizaje profundo, como son las Redes Neuronales Artificiales (ANN) y las Redes Neuronales LSTM (Long-short Term Memory), un tipo especial de redes neuronales recurrentes.

Con el análisis obtenido para los diferentes métodos, se pretende realizar comparaciones bajo las mismas circunstancias computacionales, comparaciones entre los diferentes métodos, y comparación de resultados.

Todas las metodologías desarrolladas en el presente trabajo han sido testadas en la planta química modelo Tennessee Eastman, *benchmark*, banco de pruebas obtenido de la literatura especializada en el campo de la FDD.

## 1.3. ORGANIZACIÓN

A continuación, se especifica brevemente la organización que seguirá la presente memoria, dividida en seis capítulos:



El primer capítulo recoge la presentación e introducción del trabajo, en el cual se especifica el contexto y metodologías seguidas, los objetivos que se persiguen y una breve explicación del esquema que sigue el documento.

En el segundo capítulo, se explicará desde un punto de vista teórico todas las metodologías desarrolladas. Profundizando en términos como la calidad de los procesos, análisis estadístico de los procesos, algoritmos de inteligencia computacional, Deep learning, redes neuronales artificiales y modos de representación de los resultados elegidos para el trabajo.

En el tercer capítulo se presenta en detalle la descripción del banco de pruebas elegido, en este caso la planta Tennessee Eastman, con una breve descripción del proceso, las variables implicadas, y por último los fallos que se utilizarán para la experimentación.

En el cuarto capítulo recogemos los datos obtenidos para los diferentes desarrollos, explicando con detalle la metodología seguida para su obtención, y el posterior análisis de los resultados arrojados. Comparación entre métodos durante el desarrollo del trabajo, mostrando diferentes opciones y soluciones propuestas para el problema de la detección y diagnóstico de fallos.

En el quinto capítulo del presente trabajo se presentarán las conclusiones, al realizar una visión global del trabajo, explicación de los resultados obtenidos desde un punto de vista analítico y con el objetivo de la mejora continua. Por último, sugerencias sobre futuros trabajos y una idea global en relación con el tema que puede conducir a la obtención de mejores resultados.

En último lugar, el sexto capítulo se especifica toda la bibliografía consultada para la realización de este trabajo.



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos





## CAPÍTULO II: MARCO TEÓRICO



## 2.1. CONTROL DE CALIDAD

La gestión de la producción industrial se ocupa de la toma de decisiones en relación con los procesos de producción a fin de que los productos o servicios resultantes cumplan con los estándares establecidos en términos de calidad, cantidad, tiempos de ejecución y con coste mínimo. La Calidad global y el término Excelencia en los procesos suponen el mejoramiento continuo basado en el ciclo PDCA (Plan, Do, Check, Act), y en el aseguramiento de la calidad a lo largo del proceso.

Las empresas se encuentran en un entorno actual cambiante en todos los ámbitos, tecnológico, económico, social, etc. Y eso conlleva un esfuerzo para la adaptación al entorno para seguir siendo competitivas [3]. Los modelos de gestión que siguen muchas empresas toman como puntos de referencia las normas ISO, documentos que especifican requerimientos que las organizaciones pueden implementar a fin de cumplir sus objetivos en términos de calidad, medio ambiente, prevención de riesgos laborales, seguridad y responsabilidad social. La norma ISO 9001 esta enfocada a homogeneizar los estándares de calidad de los productos y servicios de las organizaciones. La correcta implantación de estos sistemas de gestión en términos de calidad proporciona ventajas competitivas a las empresas, que suponen reducción de costes de producción, incremento en la satisfacción de los clientes, mejora de la imagen exterior de la organización, acceso a nuevos clientes potenciales y mayores niveles de productividad. [4]

La gestión de la calidad en los procesos engloba varias fases, la monitorización del proceso nos ofrece la información necesaria para controlar el funcionamiento del proceso. La detección de anomalías, errores o cambios en el funcionamiento, y su posterior actuación mediante medidas correctivas permite devolver al proceso a su punto de funcionamiento normal.

Se requieren herramientas de campo como sensores, actuadores, controladores lógicos programables (Programmable Logic Controller, PLC) que, hiperconectados entre sí, permiten un flujo continuo de información. Gracias a





los Sistemas de Control Distribuido (*Distributed Control System*, DCS), a los crecientes avances de la Industria 4.0, y la utilización de tecnologías como el Internet of Things (IoT) junto con el personal cualificado, permiten la monitorización detallada del proceso.

Una parte importante de la monitorización de los procesos, es la detección y diagnóstico de fallos (FDD) que consta de varias fases. Entendemos por fallo en un entorno industrial una condición anormal o defecto en un componente, equipamiento o subsistema que conlleva una situación de incumplimiento de las especificaciones requeridas, riesgo o daño material o personal [5]. Primero debemos detectar la presencia de un fallo, anomalía o variación en el proceso, gracias a los datos obtenidos del mismo. Después de la detección debemos identificar las variables o causas que pueden ser relevantes para poder clasificar el fallo y determinar su naturaleza, el tipo de fallo, su gravedad o repercusión en el proceso, y su localización. Por último, la recuperación tras fallo consiste en la erradicación de este, y vuelta al estado de funcionamiento normal del proceso.

Existen multitud de herramientas y técnicas aplicadas al problema de FDD, lejos de ser excluyentes, todas estas técnicas son complementarias y nos permiten abordar un tema complejo, que puede ser tratado desde diversas perspectivas [6].

Ya sea en términos cualitativos o cuantitativos, la calidad se puede medir como el porcentaje o la probabilidad de que las características o especificaciones se encuentren fuera de los límites establecidos de calidad. La variabilidad en un proceso es un concepto antagónico a la calidad de este. Si de manera abstracta se define un punto estándar u óptimo en el funcionamiento de un proceso, cualquier anomalía puede ser definida como una variación del punto de funcionamiento. Desde esta perspectiva, una de las técnicas más utilizadas para la gestión de la calidad es el Control Estadístico de Procesos.



## 2.2. CONTROL ESTADÍSTICO DE PROCESOS

El Control Estadístico de Procesos (*Statistical Process Control* en inglés, SPC) nace de la mano del Dr. Walter A. Shewhart a finales de los 20, quien en su libro “*Economic Control of Quality of Manufactured Products (1931)*” marcaría la pauta que han seguido diferentes autores en la evolución del control estadístico de los procesos. Un proceso podemos decir que está bajo control cuando, mediante la experiencia, somos capaces de predecir el comportamiento futuro del mismo dentro de unos límites establecidos [7]. Por lo tanto, el Control Estadístico de Procesos (SPC) tiene como objetivo la predicción del comportamiento de los procesos de estudio en el tiempo mediante la aplicación de técnicas estadísticas para la monitorización y control de un proceso. Se pretende determinar si el resultado concuerda con el diseño y entra dentro de un rango de tolerancia en términos de calidad.

El SPC se usa para alertar e informar a la gerencia u operarios de anomalías en los procesos con el propósito de aplicar las medidas correctivas pertinentes para devolver al proceso a su estado óptimo de funcionamiento. Una correcta implementación del SPC, conlleva beneficios productivos, como reducción del tiempo de fabricación, reducción de costes asociados a averías, reducción de desperdicios, reducción de tiempos ociosos, etc.

Los procesos están sujetos a una serie de factores de carácter aleatorio que hacen imposible el resultado de dos productos completamente iguales bajo el mismo proceso. Uno de los objetivos del control de procesos radica principalmente en reducir esas diferencias entre las características finales de los productos, dicho de otro modo, reducir la variabilidad lo máximo posible, o al menos mantenerla dentro de unos límites establecidos. [8]



### 2.2.1. VARIABILIDAD DE UN PROCESO DE PRODUCCIÓN

En los procesos de producción encontramos multitud de fuentes de variación. Aunque esta variabilidad sea provocada por la propia naturaleza aleatoria de los factores inherentes al proceso productivo, esto implica la imposibilidad de crear dos productos finales iguales.

Para determinar la variabilidad de un proceso primero debemos tomar una muestra aleatoria, que será ordenada de manera proporcional, como podemos observar en el eje vertical de la Figura 2.1. (a). Por otro lado, debemos indicar la frecuencia de cada grupo muestral, indicada en el eje horizontal que observamos en la gráfica de dicha figura. Una vez acabado este proceso obtenemos las gráficas indicadas en la Figura 2.1. (b), donde podemos observar la distribución que sigue la muestra seleccionada. Evidentemente las distribuciones varían en función de las muestras. Si solo encontrásemos causas de variaciones naturales o comunes, las distribuciones mostrarán una curva perteneciente a una distribución normal o campana de Gauss, como observamos en la Figura 2.1. (d), de modo contrario, si aparecen causas de variación asignables o especiales, causas no esperadas en el proceso, las muestras arrojaran distribuciones de la misma naturaleza de las que vemos en la Figura 2.1. (e).

La misión por lo tanto es intentar eliminar esas variaciones asignables o especiales y mantener el proceso dentro de los límites establecidos. En la Figura 2.2 podemos ver diferentes muestras de distribuciones de procesos diferentes. La Figura 2.2 (a) muestra un proceso fuera de control por causas de variación asignables. La Figura 2.2 (b) muestra un proceso bajo control, pues aparentemente solo influyen en él causas de variación naturales, pero sobrepasa los límites de especificación establecidos. Por último, la Figura 2.2 (c) nos muestra un proceso afectado únicamente por variaciones naturales y que además se desempeña dentro de los límites de especificación establecidos. Por definición, un proceso está bajo control estadístico cuando no aparecen causas asignables. Para este control es habitual el uso de Gráficos de

Control, y cálculo de límites de control para juzgar si es necesario el uso de medidas correctivas.

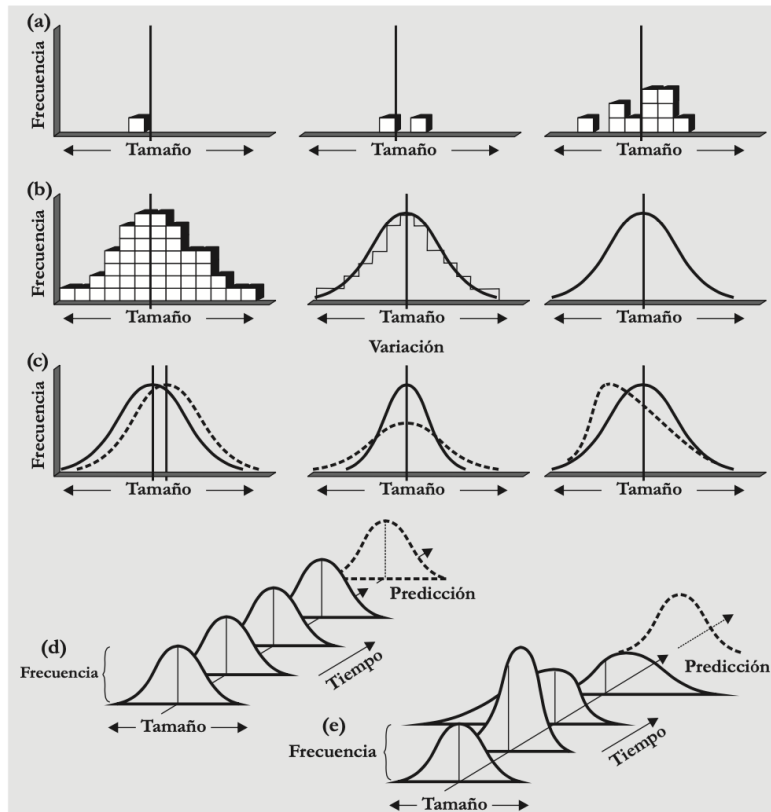


Figura 2.1. Variaciones naturales y asignables. (a) Toma de muestras. (b) Distribuciones. (c) Variación de las distribuciones. (d) Distribución de muestra con variabilidad natural. (e) Distribución de muestra con variabilidad especial. Fuente: [9]

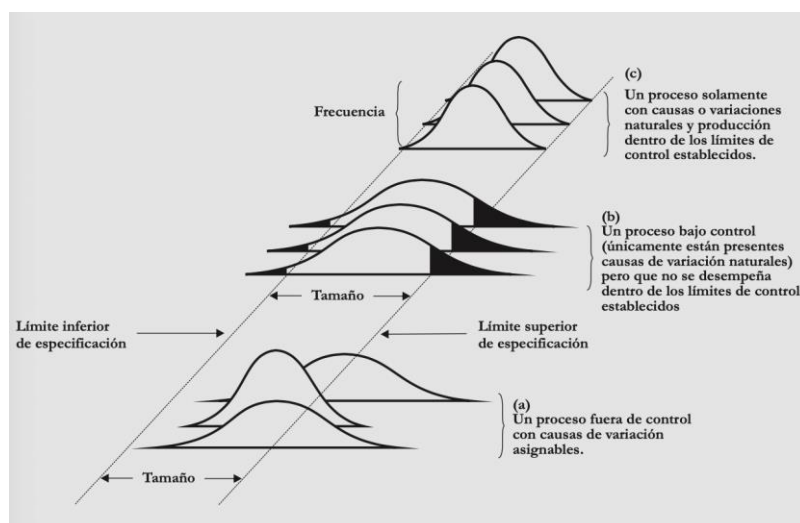


Figura 2.2. Tipos de salida del proceso. Fuente: [9]



## 2.2.2. CAUSAS DE LA VARIABILIDAD

Como ya hemos visto, existen dos categorías dentro de las causas de variación que pueden afectar a un proceso, las causas naturales o también denominadas causas comunes, y las causas asignables o especiales.

### Causas Comunes:

Son causas aleatorias, identificables, y por lo tanto inevitables. Su variabilidad es de carácter permanente, es intrínseca al proceso, y estable. Estas causas no provocan la pérdida de control del proceso.

### Causas Especiales:

Las causas especiales no son inherentes al proceso, provocan la pérdida de control del proceso, y por lo tanto deben ser identificadas y eliminadas. Dan lugar a un comportamiento impredecible y errático del proceso. La aparición de este tipo de causas debe ser informada a los responsables, que deberán aplicar las medidas correctivas necesarias para su erradicación y toma de control del proceso. Estas actuaciones suelen ser de carácter local, y habitualmente llevadas a cabo por los propios operarios encargados del proceso.

## 2.2.3. GRÁFICOS DE CONTROL

Los Gráficos de Control son la herramienta más habitual para monitorizar el estado de los procesos. A partir de los gráficos de control es posible trazar y medir la curva de una determinada característica en el espacio temporal. Propuestos a finales de los años 20 por el Dr. Walter A. Shewhart, los primeros gráficos de control, aplicados a variables cuantitativas, pretendían monitorizar el promedio y la variación de una característica de calidad. Los primeros gráficos, por lo tanto, son univariantes, ya que permiten la monitorización de una única variable sin tener en cuenta las variables restantes envueltas en el proceso.

En la gráfica de control encontramos un valor nominal situado como una línea central (CL) que corresponde con el valor objetivo del proceso, y dos límites de



control basados en la distribución de la muestra y la medida de calidad que delimitan el área de control. El valor más grande representa el límite superior de control (UCL, *upper control limit*) y el valor más pequeño viene dado por el límite inferior de control (LCL, *lower control limit*). En la Figura 2.3 observamos diferentes gráficas de control con el valor nominal y los límites de control establecidos a partir de la distribución. Observamos que en la Figura 2.3 (a) un proceso con comportamiento normal bajo control, sin embargo, en la Figura 2.3 (b) observamos un proceso con anomalías puntuales, que pueden ser signo de comportamiento errático o estar fuera de control estadístico.

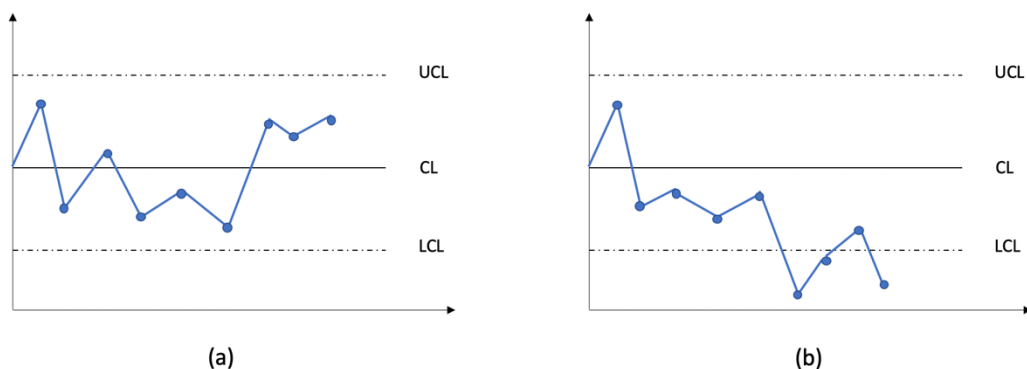


Figura 2.3. Gráficos de Control. (a) Proceso bajo control. (b) Proceso fuera de control

La disposición de la distribución y los límites de control nos permiten conocer la presencia de causas especiales, en caso de rebasamiento de los límites de control.

Los gráficos de control pueden presentar errores de dos tipos:

- Error “tipo I” (rechazar un lote de buena calidad) se presenta cuando se obtiene la conclusión de que el proceso se encuentra fuera de control debido a su estado fuera de los límites de control, pero este es debido su naturaleza intrínseca aleatoria.
- Error “tipo II” (aceptar un lote de mala calidad) se comete cuando se estima que el proceso está bajo control presentando únicamente



causas de variación naturales o comunes, cuando realmente el proceso se encuentra fuera de control.

Según W. A. Shewhart, la forma genérica de un gráfico de control corresponde con la siguiente Figura 2.4. Donde los límites de control y el valor nominal vienen dados por las siguientes ecuaciones (1), (2) y (3):

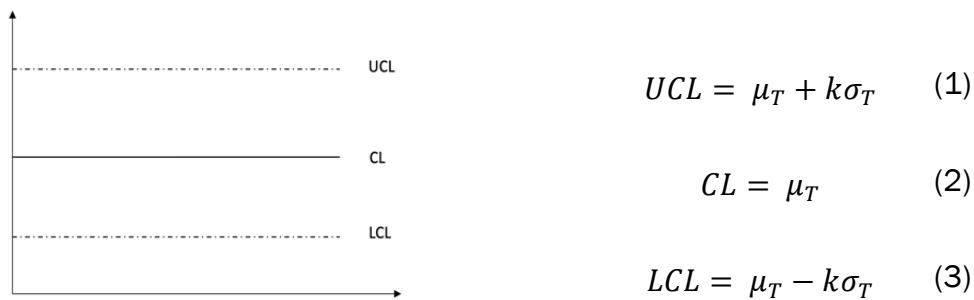


Figura 2.4. Gráfico de Control Shewhart

Donde:

$T$  es el estadístico muestral que mide la característica de calidad estudiada.

$\mu_T$  es la media del estimador  $T$ .

$\sigma_T$  es la desviación típica del estimador  $T$ .

$k$  es una constante que indica la distancia entre el valor central y los límites de control expresada en unidades de desviación típica  $\sigma_T$ . [10]

Por lo tanto, la distancia entre el valor central y los límites vienen dados por la desviación típica y la constante  $k$ , normalmente esta constante tiene valor igual a 3, en ese caso los límites de control son denominados tres-sigma.

Años más tarde de la aparición de los gráficos de Shewhart como técnica para el control de los procesos, fueron introducidos los gráficos de control para atributos, los gráficos tipo  $p$  y  $c$  permiten la monitorización del número de lotes



aptos y lotes defectuosos, por lo tanto, nacen los gráficos de control aplicados a variables cualitativas.

Como reinterpretación de los gráficos Shewhart, en la década de los 60, aparecen los gráficos de sumas acumuladas (CUSUM, por sus siglas en inglés) y los gráficos de promedios móviles exponencialmente ponderados (EWMA) [11]. Principalmente estos gráficos permiten una detección más rápida de pequeñas variaciones en el proceso, aunque siguen siendo extremadamente útiles en la actualidad, tienen como principal inconveniente su carácter univariante.

Para procesos con multitud de variables correlacionadas, el uso de estas técnicas univariantes puede resultar insuficiente. El control estadístico de procesos adaptado a la industria moderna requiere de técnicas que tengan en consideración la correlación existente entre los diferentes datos obtenidos, con el fin de predecir con mayor exactitud el comportamiento de la planta en su totalidad.

En un gráfico de control  $\bar{X}$  de Shewhart, la probabilidad de que un punto exceda de los límites de control estándar 3-sigma en un proceso bajo control (error tipo I) es de 0.0027. Para el control de dos variables completamente independientes, la probabilidad de que aparezca una señal fuera de los límites cuando el proceso está bajo control es  $1 - (1 - 0,0027) \cdot (1 - 0,0027)$  que resulta 0.0054. De aquí se deduce que la probabilidad de que se provoque un error tipo I es mayor cuando se utilizan dos gráficos de control sin modificación de los límites [12]. Para  $p$  variables independientes, la probabilidad  $\alpha'$  de error tipo I en su conjunto es:

$$\alpha' = 1 - (1 - \alpha)^p \quad (4)$$

donde la probabilidad de error tipo I en el gráfico es  $\alpha$ .

La primera aportación al estudio de técnicas de control estadístico de procesos de tipo multivariado fue realizada por Harold Hotelling (1947). El estadístico  $T^2$





Hotelling es una variable aleatoria, unidimensional construida con la combinación de información de posición y variación de las variables del proceso.

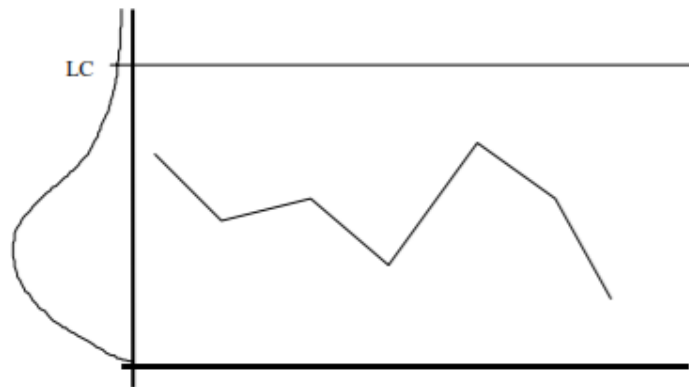


Figura 2.5. Gráfico  $T^2$  de Hotelling de un proceso bajo control. Fuente: [12]

De esta forma se sugiere la utilización de un gráfico similar al Shewhart, pero con único límite superior (LC), el proceso estará fuera de control cuando se supere el límite LC. Figura 2.5.

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

Dentro de los métodos basados en datos o señales, encontramos las técnicas estadísticas multivariantes, también denominado Control Estadístico de Procesos Multivariante (MSPC), son técnicas estadísticas para el control de variables correlacionadas, que permiten el análisis y monitorización del conjunto de variables y detección de fallos o anomalías cuando se superan unos límites establecidos para una nueva variable establecida del proceso de control. El Análisis de Componentes Principales (PCA en inglés), es una de las técnicas MSPC más utilizadas.

El método de Análisis de Componentes Principales tiene por objeto la reducción de un conjunto de variables a las que se puede denominar como originales, dando lugar a un conjunto reducido del original, al que llamaremos de “*componentes principales*”, que se caracterizan por estar incorrelacionadas



entre sí. Las variables  $a$ , denominadas componentes principales son el resultado de una transformación lineal ortogonal de las variables iniciales. Estas nuevas variables obtenidas se pueden colocar en orden decreciente de su varianza, estimando así un orden decreciente de información aportada que nos será útil en el proceso de selección del número de variables utilizadas para aportar la información del proceso [13].

Esta reducción del número de variables que ejecutamos mediante el PCA obedece a la intención de simplificar la dimensión del problema sin que esto suponga una pérdida sustancial de información.

El problema puede abordarse desde diversos enfoques, que se resumirán brevemente a continuación:

- Encontrar un subespacio de dimensión menor que el original  $p$ , tal que al proyectar sobre él los datos se produzca la menor distorsión posible. Desde una perspectiva geométrica, como podemos ver representado en la Figura 2.6, donde se representa un diagrama de dispersión en el que la nube de puntos se distribuye de manera intuitiva como una elipse. De esta manera se puede afirmar que la recta representada en la Figura 2.6 que corresponde con el eje mayor de la elipse es la recta que minimiza las distancias ortogonales entre los puntos. [14]

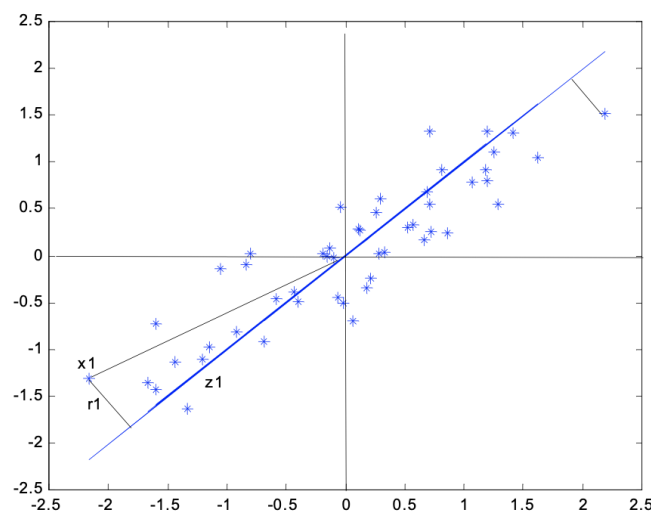


Figura 2.6. Nube de puntos. Fuente: [14]



- Se pretende encontrar el conjunto de variables con menor dimensión que representen con mayor exactitud el proceso, a fin de poder identificar los fallos.
- A partir de la matriz original de datos se crean dos subespacios, uno que capturará la tendencia del proceso, que será el formado por las  $a$  variables correspondientes con las componentes principales, y otro que contiene el ruido del proceso.

La extracción de las componentes principales se realiza sobre variables ya tipificadas, evitando así ponderaciones erróneas por diferentes escalas en los rangos de las variables.

Se considera una matriz de datos  $X \in \mathbb{R}^{n \times m}$  que representa las  $n$  observaciones de  $m$  variables, de la forma:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & \vdots \\ x_{31} & x_{32} & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \quad (5)$$

Como hemos comentado en la introducción, será necesaria la normalización de los datos a una distribución Normal  $N(0,1)$ .

A partir de esta matriz (5) podemos hallar la matriz de correlación  $R$  (6) y su descomposición en valores singulares (7): [6]

$$R = \frac{1}{n-1} X^T X \quad (6)$$



$$R = V \Lambda V^T \quad (7)$$

Donde  $V \in \mathfrak{R}^{n \times m}$  es la matriz de vectores propios de R, también llamados vectores de carga y  $\Lambda \in \mathfrak{R}^{n \times m}$  es la matriz diagonal de los valores propios reales no negativos en orden decreciente de R ( $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_m \geq 0$ ), donde  $\lambda_i = \sigma_i^2$ , es decir la varianza.

Eligiendo la variabilidad aceptada para nuestro método, escogemos los  $a$  vectores propios correspondientes a los valores propios más grandes y los almacenamos en la matriz P de dimensión reducida, tal que  $P \in \mathfrak{R}^{m \times a}$ , es la matriz de vectores conocidos como *loadings* que marca la dirección de nuevo espacio.

Para la elección del número de componentes, se puede optar por establecer una variabilidad mínima aceptada y seleccionar las  $a$  primeras variables que permiten cubrir con un porcentaje esperado de cota de varianza (valores de la diagonal de la matriz  $\Lambda$ ) y también denominados componentes principales.

Se calcula la proyección de los datos de partida en un espacio más reducido T:

$$T = X P \quad (8)$$

A cada uno de los componentes del nuevo espacio T se le denomina *scores*. A partir de P y T podemos volver al espacio de partida.

$$\hat{X} = \sum_{i=1}^a t_i p_i^T \quad (9)$$

y la diferencia entre X y  $\hat{X}$  es la matriz de ruido o residuos mencionada antes E

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

Finalmente podemos representar el conjunto de datos de partida como: [6]

$$X = \hat{X} + E \quad (11)$$



### 2.2.4.1. Estadísticas para la monitorización

En este método se definen dos estadísticos de monitorización calculados a partir del PCA desarrollado sobre los datos históricos del proceso. El objetivo de establecer estos estadísticos es determinar un límite de acción y alarma mediante un umbral. La estadística de Hotelling utiliza el espacio de dimensión  $a$ . La estadística  $Q$  se utiliza para la monitorización del resto del espacio correspondiente a los  $m - a$  valores singulares despreciados.

#### Estadística de Hotelling ( $T^2$ )

A partir de la matriz  $X$  normalizada con media 0 y desviación 1 y  $x \in \mathfrak{R}^{m \times 1}$  se puede definir la estadística  $T^2$  como: [6]

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

Donde  $\Lambda_a$  contiene las primeras  $a$  filas y columnas de la matriz  $\Lambda$  y  $a$  es el número de componentes principales elegidos.

Mediante este método obtendremos una matriz estadística  $T^2$ . Ahora debemos calcular el umbral que determinara si nuestro proceso está controlado o está fuera del control estadístico. El umbral se calculará fuera de línea y corresponde a la siguiente ecuación [15]:

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

Donde  $n$  es el número de muestras tomadas,  $a$  el número de componentes principales elegidas, y  $F_\alpha(a, n - a)$  el valor de la distribución F de Fisher-Snedecor con un nivel de significancia  $\alpha$ , que especifica el grado de compromiso de las falsas alarmas. Los valores por defecto de  $\alpha$  se sitúan en un rango entre 0.05 y 0.01.

#### Estadística $Q$ o SPE:

Al igual que calculábamos la estadística  $T^2$ , partiremos de nuestra matriz normalizada  $X$  y de  $x \in \mathfrak{R}^{m \times 1}$ , y definimos la estadística  $Q$  como: [6]



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

donde  $r$  es el vector de residuos:

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

Al igual que sucedía en el método de Hotellings, ahora podemos calcular un umbral  $Q_\alpha$  que nos determinará los límites del funcionamiento controlado. Este umbral  $Q_\alpha$  es:

$$Q_\alpha = g \chi^2(\alpha, h) \quad (16)$$

Con [15]:

$$g = \frac{\sigma_Q^2}{(2 * \bar{Q})} \quad (17)$$

$$h = \frac{(2 * \bar{Q}^2)}{\sigma_Q^2} \quad (18)$$

donde  $\bar{Q}$  es la media de los valores de  $Q$ ,  $\sigma_Q^2$  es su varianza, y  $\chi^2(\alpha, h)$  es la distribución estadística  $\chi^2$  con  $h$  grados de libertad y probabilidad  $\alpha$ .

Se estima que incurriremos en un fallo si las observaciones  $T^2$  y  $Q$  hayan sobrepasado cada uno de sus umbrales  $n$  veces consecutivas, en nuestro caso hemos decidido que ese  $n$  sea 10. Se debe elegir un número  $n$  no demasiado pequeño para caer en posibles falsas alarmas, ni tampoco lo suficientemente grande como para que pase por alto posibles fallos del sistema.



## 2.3. TÉCNICAS DE DIAGNOSTICO Y CLASIFICACIÓN DE FALLOS

### 2.3.1. INTELIGENCIA ARTIFICIAL, MACHINE LEARNING Y DEEP LEARNING.

Los primeros trabajos considerados el embrión de la Inteligencia Artificial moderna surgen en la década de los 40. Uno de los padres de la IA, Alan Turing publicaría en 1950 el artículo *“Computing Machinery and Intelligence”* donde se describe por primera vez la capacidad de que las maquinas puedan adquirir comportamientos que simulen el razonamiento humano. Según palabras del propio A. Turing, *“Existirá Inteligencia Artificial cuando no seamos capaces de distinguir entre un ser humano y un programa de ordenador en una conversación a ciegas”*.

El Machine Learning (ML) es uno de los enfoques principales de la inteligencia artificial, que consiste en algoritmos matemáticos que permiten al ordenador o máquina imitar la forma de proceder de los humanos a partir de patrones de datos. La forma de aprendizaje puede ser supervisado o no supervisado.

El Aprendizaje Profundo o Deep Learning (DL) es un conjunto de algoritmos matemáticos que mediante arquitecturas computacionales permiten a los ordenadores o máquinas aprender desde la experiencia y el entendimiento jerárquico [16]. Se trata de un subgrupo dentro del Machine Learning, de hecho, se podría afirmar que el Deep Learning es la nueva evolución del Machine Learning.

Mientras que la mayoría de los métodos de Deep Learning hacen uso de redes neuronales, el Machine Learning abarca muchas técnicas que incluyen las redes neuronales, los árboles de decisión, los algoritmos de agrupamiento, etc. Las redes neuronales también reciben el nombre de Redes Neuronales Profundas, o Deep Neural Networks en inglés, precisamente haciendo referencia a las capas que forman su arquitectura [17].



### 2.3.2. REDES NEURONALES ARTIFICIALES (ANN).

El cerebro humano ha sido motivo de estudio desde la concepción del ser humano. Son muchos los puntos de vista y campos de investigación que ha pretendido comprender y dar respuesta a los dilemas que da a lugar el funcionamiento del cerebro humano. Desde diversas disciplinas como la filosofía, la psicología, la fisiología y por supuesto la ingeniería se ha pretendido dar uso al potencial que esconde el cerebro humano. Desde la perspectiva ingenieril, se ha pretendido formar un sistema que emule las diferentes funciones básicas del cerebro, dando como resultado una tecnología llamada Computación Neuronal o también denominada Redes Neuronales Artificiales.

En 1943 el neurobiólogo Warren McCulloch publica con la colaboración del estadístico Walter Pitts el artículo “*A logical calculus of Ideas Imminent in Nervous Activity*” que sentará las bases para el desarrollo de campos como la Inteligencia Artificial, la Computación Digital y el funcionamiento de la red *Perceptron*. Una conferencia organizada por la Fundación Rochester, en el año 1956, se establece como la primera toma de contacto con el estudio y desarrollo de las Redes Neuronales Artificiales. Nathaural Rochester, del equipo de investigación de IBM presenta un modelo de ANN, que es considerado como el primer prototipo. Y un año más tarde, en 1957, es cuando Franz Rosenblatt presenta un trabajo de investigación sobre un sistema innovador capaz de identificar patrones ópticos, al que denomina Red *Perceptron*. Las limitaciones de este nuevo sistema, como la resolución del problema lógico OR, son puestas de manifiesto durante la década de los 60 por diferentes investigaciones (Minsky y Papert) que redujeron el interés, inversión e investigación en el campo de la computación neuronal. [18]

A pesar de la reducción en la investigación, diferentes autores presentan posteriores evoluciones en la investigación de la computación neuronal, pero no es hasta el año 1982 cuando John Hopfield, con la aparición del algoritmo *Backpropagation* devuelve la computación neuronal a la vanguardia de la investigación computacional. En la actualidad, diversos grupos de investigación





continúan realizando avances en el área de las redes neuronales artificiales, un campo con gran potencial en el campo computacional, y con un gran margen de evolución.

Las Redes Neuronales Artificiales (ANN del inglés, *Artificial Neural Networks*) tratan de imitar, en cierta medida, el comportamiento del propio cerebro humano. Su organización y estructura también es análoga a la distribución de un cerebro humano, formado por neuronas interrelacionadas entre sí. Las ANN aprenden de la experiencia, lo que nos conduce al entrenamiento de las redes a fin de establecer un modelo que generalizará lo aprendido y generará una respuesta/salida en función de las entradas.

El sistema nervioso humano es una estructura muy compleja, formada por neuronas interconectadas entre sí. Alrededor de  $10^{11}$  neuronas componen el sistema nervioso, y las interconexiones entre las neuronas se estima que alcanza el orden de  $10^{15}$ . Las neuronas, como unidad fundamental del sistema nervioso, se encargan de recibir, procesar y transmitir la información a través del sistema. En la Figura 2.7 observamos la estructura de una neurona biológica. Del cuerpo de la neurona, o soma, parten ramificaciones llamadas dendritas, que portan la información que se transmite a otras neuronas. El punto de interconexión entre las neuronas es lo que se denomina sinapsis. La información de entrada a las neuronas se conduce al núcleo donde se suman. La actividad de las neuronas depende de las entradas que reciben, pudiendo estas entradas activar la celda, o por el contrario, inhibirla.

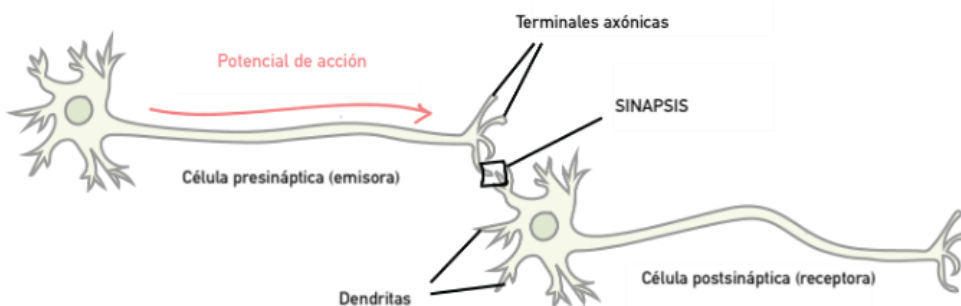


Figura 2.7. Componentes de una neurona.



En las Redes Neuronales Artificiales, ANN, las neuronas funcionan como unidades básicas de procesamiento. Los elementos procesadores (PE) procesan las entradas (input) recibidas, para ofrecer una salida (output). En este caso, las salidas de cada neurona se transforman en entradas de otras mediante conexiones ponderadas, correspondientes con las sinapsis de las neuronas. Generalmente las Redes Neuronales están compuestas por diversas capas de neuronas, o elementos procesadores, ordenadas secuencialmente.

La red *perceptron*, desarrollada por Rosenblatt en 1957 para el reconocimiento de imágenes como propósito, consistía en el modelo más simple de una red neuronal, con aprendizaje supervisado. El principal defecto de la red *perceptron* era la limitación a la hora de clasificar patrones, por el uso de clasificadores lineales.

En la Figura 2.8 se muestra la estructura de una única unidad de procesamiento, o *perceptron*. Las conexiones de entrada, desde donde se reciben los estímulos a partir de los cuales la neurona generará unos valores de salida a partir de una función de propagación y una función de activación. La función de propagación consiste comúnmente en un sumatorio ponderado de las entradas, que puede estar acompañado por un sesgo  $b$ .

$$Y = X \cdot W \quad (19)$$

$$Y = \sum x_i \cdot w_i = w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + \dots + w_n \cdot x_n \quad (20)$$

donde cada una de las entradas  $(x_1, x_2, x_3, \dots, x_n)$  viene ponderada por el vector de pesos  $W = (w_1, w_2, w_3, \dots, w_n)$  que son los parámetros que determinan nuestro modelo.

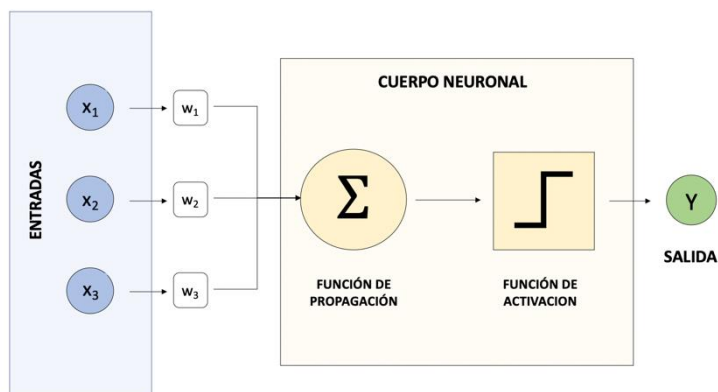


Figura 2.8. Diagrama de una Neurona Artificial. (Fuente: Elaboración propia)

Se puede realizar una analogía acertada entre la función descrita para una única neurona con un modelo de regresión lineal. En el modelo de regresión lineal contábamos con un término independiente, este término también lo encontramos dentro de nuestras entradas a la neurona, y es conocido como sesgo, o bias en inglés, en el cual la variable asignada siempre es 1.

$$Y = \sum x_i \cdot w_i (+b_i) = b + w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + \dots + w_n \cdot x_n \quad (21)$$

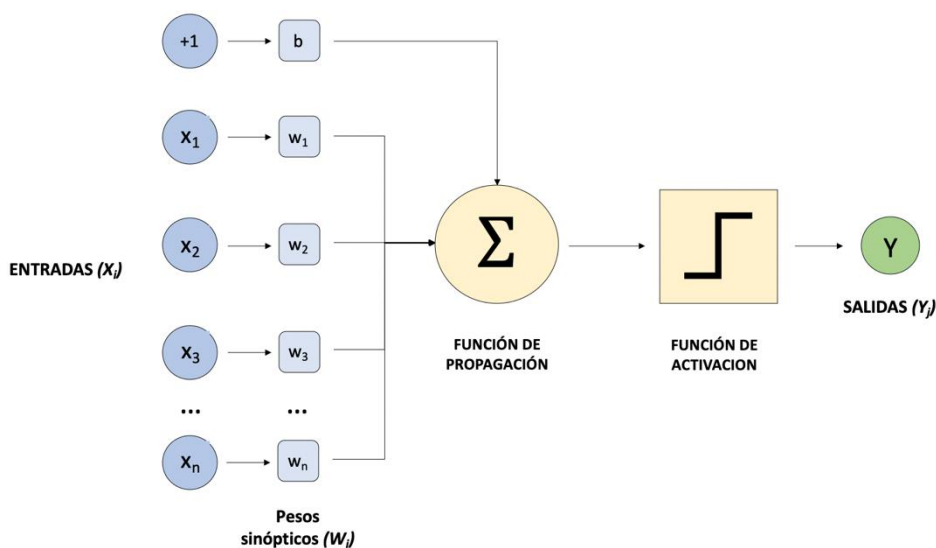


Figura 2.9. Diagrama de una Neurona Artificial con bias.



La función de transferencia o activación F, se aplica al resultado generado en la función de propagación para la producción de la salida de la neurona. Es uno de los elementos claves en el funcionamiento de las ANN, pues otorga la flexibilidad necesaria para estimar relaciones no lineales. Suelen ser funciones de salida acotada, y encontramos diferentes tipos de funciones de transferencia (Tabla 1). Las funciones F más utilizadas son la función Sigmoid y Tangente hiperbólica, aunque también encontramos otras menos comunes como la función lineal, base radial o la función de discriminación, también denominada escalón. [19]

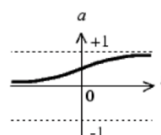
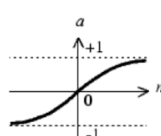
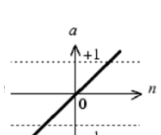
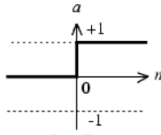
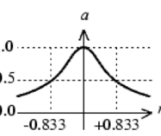
Función de Activación	Ecuación	Rango	Gráfica
Sigmoidea	$a = \frac{1}{1 + e^{-x}}$	[0, 1]	
	$a = tgh(x)$	[-1, 1]	
Lineal	$a = x$	$[-\infty, +\infty]$	
Escalón	$a = H(x)$	[0, 1]	 $a = hardlim(n)$
Base Radial	$a = A e^{-Bx^2}$	[0, 1]	 $a = radbas(n)$

Tabla 1. Funciones de Activación

### 2.3.3. REDES NEURONALES DE PROPAGACIÓN HACIA ADELANTE.

Normalmente las redes neuronales con mayores dimensiones arrojan mejores resultados cuando se requiere de un mayor cálculo computacional, en función de la complejidad del problema. Las redes *feedforward* o también denominadas *perceptrones* multicapa (*Multilayer Perceptrons*, MPLs) son modelos más complejos que el *perceptron*. Reciben la denominación de *feedforward* porque la información fluye siempre hacia adelante, por lo tanto, también reciben el nombre de Redes Neuronales de Propagación hacia adelante. Las redes neuronales con conexiones de realimentación reciben el nombre de Redes Neuronales Recurrentes.

Las ANN *feedforward* suelen presentar múltiples capas de neuronas en cascada de forma que las salidas de una capa N-1, pasa a ser las entradas de la capa N, y a su vez las salidas de ésta, pasarán a ser las entradas de la capa N+1, como podemos observar en la Figura 2.10 La estructura por tanto de una Red Neuronal de propagación hacia adelante constará de una capa de entrada, una capa de salida, y un número variante de capas ocultas.

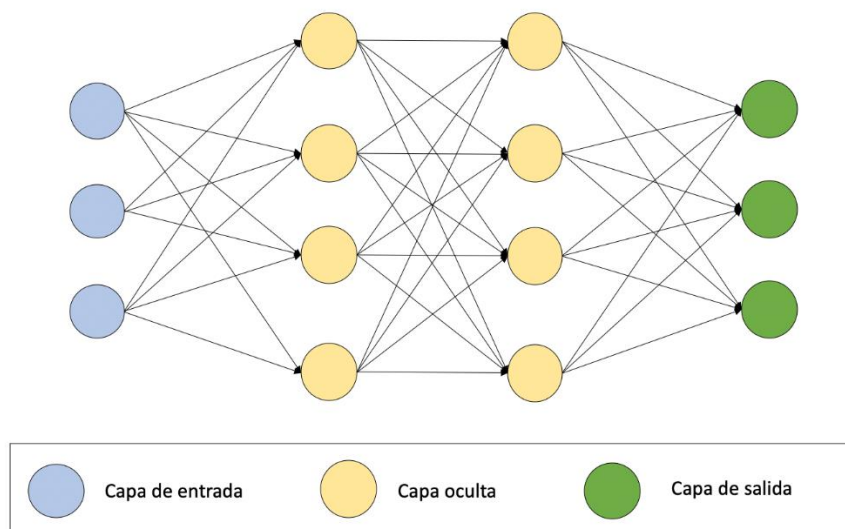


Figura 2.10. Estructura por capas de una RNA. Fuente: Adaptación de [20]



- Capa de entrada: Cada neurona de entrada corresponde con un elemento del vector de entrada a la red  $(x_1, x_2, x_3, \dots, x_n)$ .
- Capas ocultas: Tantas capas como disponga la red, está compuesta por neuronas con sus pesos específicos, y su sesgo o bias correspondiente. Son procesadores intermedios, y la base del cálculo de la respuesta de la red neuronal.
- Capa de salida: El número de neuronas de la capa de salida corresponde con el número de respuestas que ofrezca la red. En caso de ser una red de clasificación coincide con las clases, etc. [20]

#### 2.3.4. ENTRENAMIENTO DE LAS REDES NEURONALES ARTIFICIALES.

Una de las principales características de las ANN es su capacidad de aprendizaje. Las Redes Neuronales deben pasar por un procedimiento de aprendizaje o entrenamiento, en el cual se pretende modelar la red para que produzca una salida deseada a partir de unos datos de entrada. Durante este proceso de entrenamiento los pesos convergen gradualmente [18] hacia los valores que produzcan la salida deseada.

El entrenamiento de una ANN puede ser supervisado, o no supervisado. El entrenamiento supervisado consiste en el emparejamiento de la respuesta deseada a los datos de entrada. Este tipo de entrenamiento ha recibido críticas por su incoherente analogía biológica con el proceso de aprendizaje del ser humano. El algoritmo no supervisado fue desarrollado por primera vez por Hebb (1949). A diferencia del entrenamiento supervisado, estos sistemas de aprendizaje no requieren de la salida deseada para el entrenamiento. El modelado de la red y la variación de los pesos se producen hasta que la salida ofrecida por la red es consistente. El proceso de entrenamiento extrae las propiedades estadísticas del conjunto de vectores de entrenamiento u agrupa en clases los vectores similares [18].

### 2.3.4.1. Sobreajuste (dropout)

El entrenamiento de una ANN de grandes dimensiones con un conjunto de datos relativamente pequeño puede producir un rendimiento deficitario de la red. El sobreajuste u *overfitting* se produce como consecuencia del sobreentrenamiento de la red [21]. El mecanismo más simple para evitar el sobreajuste de nuestra red consiste simplemente en simplificar el proceso de aprendizaje, es decir, reducir las dimensiones de la red, eliminando capas ocultas, o reduciendo el número de neuronas por capa.

Como técnica destacada para evitar el sobreajuste está el *dropout*. El *dropout* consiste en transformar una red compleja en varias redes más simples ignorando aleatoriamente partes de la red. Para el uso de esta técnica, es necesario especificar la probabilidad o tasa de abandono que se empleará, por ejemplo, como vemos en la Figura 2.11 se ha empleado una tasa de abandono, *dropout*, con  $p = 0.5$ . Este valor variará en función de las necesidades del aprendizaje de la ANN.

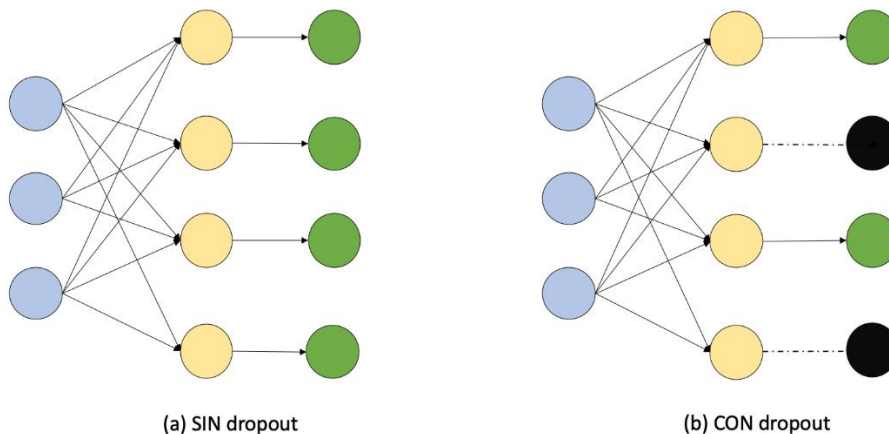


Figura 2.11. ANN sin Dropout (a) y con Dropout (b) (Fuente: Adaptación de [21])



### 2.3.5. REDES NEURONALES RECURRENTE

El razonamiento cognitivo, es la capacidad del ser humano para procesar información a partir del conocimiento adquirido mediante la experiencia, y la capacidad de valorar la información de manera subjetiva. Consiste por tanto en procesos que se basan en el aprendizaje, memoria, resolución de problemas y toma de decisiones [22].

Una de las principales limitaciones de las redes *feedforward* que hemos visto hasta ahora es la incapacidad de introducir la memoria en su arquitectura, la información viaja de la entrada hacia la salida de la red sin ningún tipo de realimentación.

Las redes recurrentes (*Recurrent Neural Networks*, RNN) son estructuras más complejas que las redes *feedforward*. La estructura no está organizada por capas, y su característica principal es la introducción de interconexiones arbitrarias entre las neuronas, lo que introduce la idea de temporalidad en la red, dotándola de algún modo de memoria.

Las RNN son arquitecturas ideales para la resolución de problemas con no-linealidades temporales significativas, el comportamiento arbitrario de las interconexiones de retroalimentación de este tipo de redes las hacen ideales para la resolución de problemas de identificación y clasificación de patrones a través del tiempo.

En la Figura 2.12 se observa la arquitectura básica de una RNN  $A$ , que recibe una entrada  $x$  y devuelve una salida  $h$ . El bucle de realimentación que observamos permite que la información viaje a través de la red de un ciclo al siguiente. Su comportamiento no difiere mucho del de las redes *feedforward*, la RNN pueden ser creadas como múltiples copias de la misma red, donde la información fluye de un nodo a otro. En la Figura 2.13 vemos un esquema de una RNN desenrollada con  $n$  nodos.



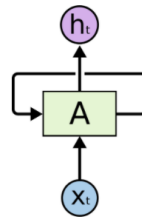


Figura 2.12. Esquema básico de una RNN (Fuente: Extraído de [23])

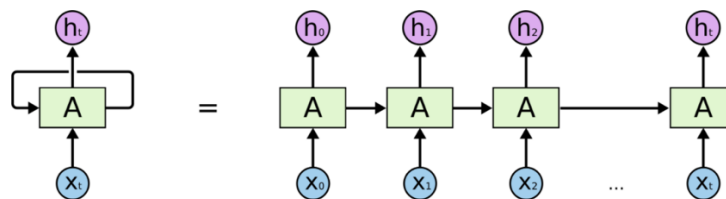


Figura 2.13. RNN desenrollada (Fuente: Extraído de [23])

En los últimos años ha habido multitud de avances en la aplicación de RNN a la resolución de problemas, tales como reconocimiento de voz, tratamiento de imágenes, modelado de lenguaje, traducción de textos, etc. El tipo de RNN más utilizado en la actualidad son las Redes Neuronales LSTM (*Long Short Term Memory*), un tipo especial de red neuronal recurrente que ofrece resultados sorprendentes en multitud de problemas.

### 2.3.6. REDES NEURONALES LSTM

Las RNN por tanto conectan información mediante realimentaciones a fin de procesar información aprendida o memorizada por la red. Estas realimentaciones funcionan correctamente siempre que la brecha temporal sea relativamente corta, es decir a corto plazo (Figura 2.14). El principal problema que surgía con la utilización de las ya nombradas RNN es su dificultad para conectar información con brechas temporales muy amplias, es decir, a largo plazo (Figura 2.15), llegando incluso a ser incapaces si la brecha era muy

grande, y por tanto no completar el proceso de aprendizaje de una manera correcta.

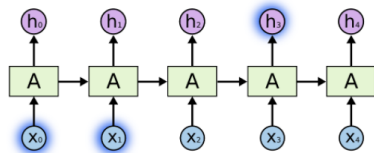


Figura 2.14. RNN con dependencia a corto plazo (Fuente: [23])

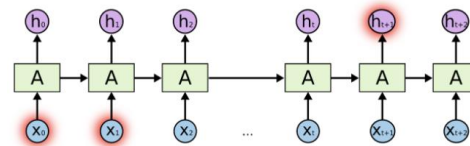


Figura 2.15. RNN con dependencia a largo plazo (Fuente: [23])

Para solucionar el problema de las dependencias a largo plazo, Hochreiter y Schmidhuber (1997) desarrollan un tipo de RNN especial al que denominan las redes LSTM, que utilizan unas celdas de procesamiento de información algo más compleja que las RNN.

Al igual que las RNN las redes LSTM están dispuestas en forma de cadena, la diferencia reside en su módulo repetidor. Mientras como vemos en la Figura 2.16, las RNN estándar tienen neuronas con una única función de transferencia, las redes LSTM, en la Figura 2.17, constan de cuatro, que interactúan entre si a fin de proporcionar esa memoria a largo plazo objetivo.

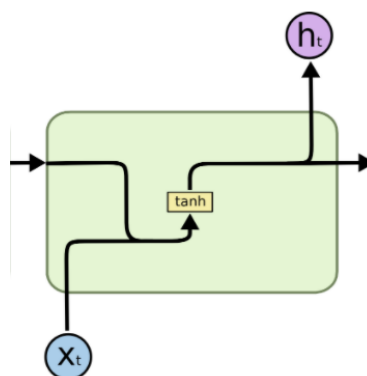


Figura 2.16. Repetidor estándar de una Red Neuronal Recurrente (RNN). Fuente: [23]

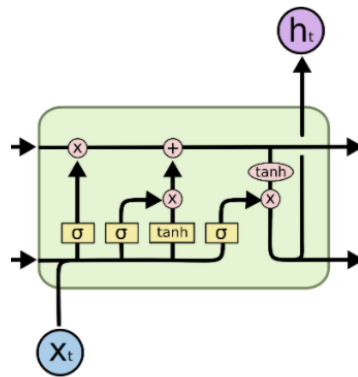


Figura 2.17. Repetidor estándar de una Red LSTM. Fuente: [23]

En el esquema de la figura 2.18 se observa la estructura de una celda de memoria de una red LSTM. Los datos de entrada a la celda  $x_k$  y la información que se envía a las capas ocultas se denomina  $h_k$ . La celda de memoria LSTM consta de cuatro compuertas que definen el estado actual  $s_k$  de la celda.

La primera es la compuerta del olvido  $f_k$ , que determina la información que será borrada de la memoria. La compuerta de candidatura  $g_k$  que define la información añadida a la celda, y a su vez la compuerta de entrada  $i_k$  controla el nivel de actualización de estado en la celda. Por último, la compuerta de salida  $o_k$  conduce la información de la memoria a la siguiente capa oculta.

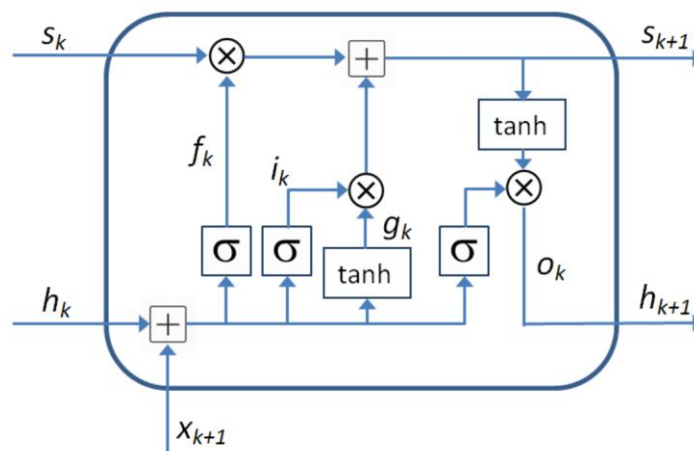


Figura 2.18. Esquema general celda de memoria LSTM (Fuente: [24])



El siguiente estado de la celda viene dado por:

$$s_{k+1} = f_{k+1} \odot s_k + i_{k+1} \odot g_{k+1} \quad (22)$$

donde  $\odot$  representa el producto elemento por elemento (producto de Hadamard)

El valor que se entrega a la siguiente capa oculta viene definido por:

$$h_{k+1} = o_{k+1} \odot \tanh(s_{k+1}) \quad (23)$$

Las siguientes ecuaciones determinan la actualización de las compuertas:

$$i_{k+1} = \sigma (R_i h_k + W_i x_{k+1} + b_i) \quad (24)$$

$$f_{k+1} = \sigma (R_f h_k + W_f x_{k+1} + b_f) \quad (25)$$

$$g_{k+1} = \tanh (R_g h_k + W_g x_{k+1} + b_g) \quad (26)$$

$$o_{k+1} = \sigma (R_o h_k + W_o x_{k+1} + b_o) \quad (27)$$

donde  $\tanh(\cdot)$  es la tangente hiperbólica,  $\sigma(\cdot)$  la función de activación sigmoidea,  $\sigma(x) = (1 + e^{-x})^{-1}$ ,  $W$  los pesos sinápticos,  $R$  los pesos recurrentes y  $b$  los sesgos para cada compuerta.

La arquitectura de las capas RNN y las LSTM es semejante. Constan de una capa de entrada, una capa de salida y un número determinado de capas intermedias, solo que las neuronas de las redes LSTM son las explicadas en esta sección (Figuras 2.19 y 2.20 respectivamente).

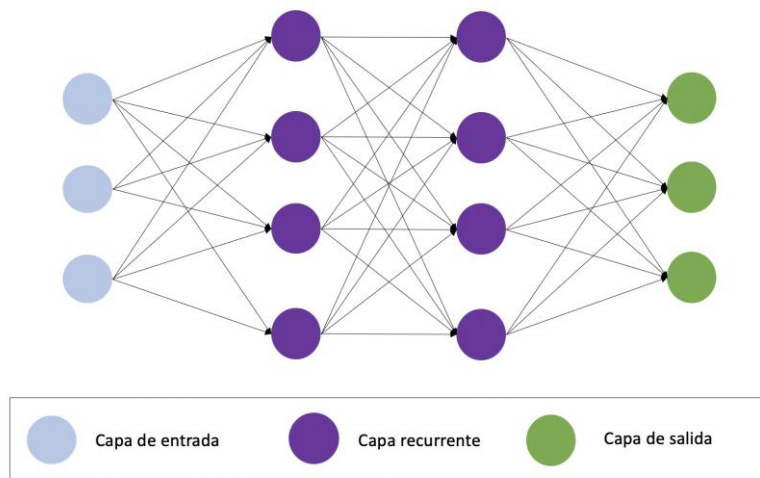


Figura 2.19. Estructura de una RNN con dos capas intermedias

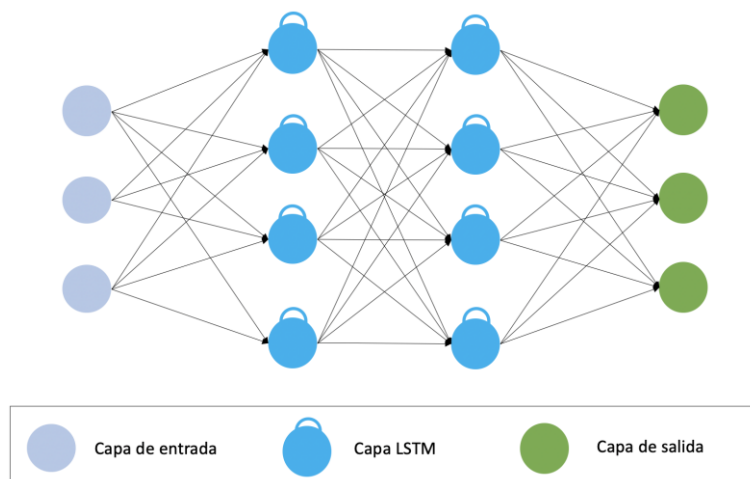


Figura 2.20. Estructura de una red LSTM con dos capas intermedias

### 2.3.7. MATRIZ DE RESULTADOS. MATRIZ DE CONFUSIÓN.

La matriz de confusión es una herramienta que proporciona el software MATLAB™ para visualizar los datos de clasificación. Después de un proceso de aprendizaje automático supervisado se deben comprobar los resultados obtenidos. En el caso de la clasificación la aplicación de la matriz de confusión



nos facilita la medición de aciertos y fallos a la hora de clasificar un gran conjunto de datos en las diferentes categorías.

La arquitectura de la matriz siempre es igual (Figura 2.21), donde cada una de las filas de la matriz representan las clases predichas (*Output classes*), mientras tanto las columnas representan el conjunto de clases deseadas (*Target classes*). La diagonal de la matriz compuesta por los valores [(1,1); (2,2); ...; (n, n)] donde n es el número total de clases, corresponden con las celdas de acierto en la clasificación, mientras que todos los datos que se sitúen fuera de la diagonal de la matriz serán datos incorrectamente clasificados.

La última fila muestra el total de observaciones para las clases deseadas, con un porcentaje que hace referencia a la tasa de verdaderos positivos (sensibilidad) y de falsos negativos. La columna de la derecha muestra el total de observaciones de cada clase predicha y cuantifica con un porcentaje la tasa de aciertos detectados (precisión) y su tasa de falsos descubrimientos. [5]

La celda más importante de la matriz es la situada en la parte inferior derecha, donde se muestra el porcentaje de predicciones correctas e incorrectas. Este valor será el utilizado como referencia a lo largo del presente trabajo.

1	150 20.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	150 20.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	150 20.0%	1 0.1%	0 0.0%	99.3% 0.7%
4	0 0.0%	0 0.0%	0 0.0%	149 19.9%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	150 20.0%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	99.3% 0.7%	100% 0.0%	99.9% 0.1%
	1	2	3	4	5	

Figura 2.21. Matriz de confusión con una tasa de 99.9% de clasificaciones correctas



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos





## CAPÍTULO III: PLANTA TENNESSEE EASTMAN





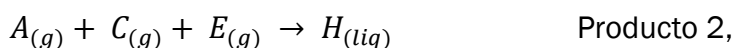
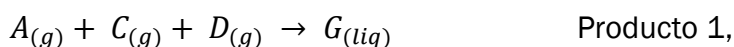
### 3. CASO DE ESTUDIO: PLANTA TENNESSEE EASTMAN.

Durante un gran periodo de tiempo la comunidad académica había expresado el interés en contar con un problema real para poder testar los diferentes métodos de control de procesos emergentes. A finales del siglo XX aparece el Proceso Tennessee Eastman (TEP), publicado por J. J. Downs y E. F. Vogel (1993), en el cual la Universidad de Tennessee junto con Eastman Chemical Company presentan un modelo basado en una planta real de la compañía.

El proceso TE es ampliamente utilizado por la comunidad científica como banco de pruebas debido a que se trata de un proceso altamente no lineal, que involucra procesos complejos y un elevado número de variables que ofrecen múltiples posibilidades desde un punto de vista de la ingeniería de control de procesos.

#### 3.1. DESCRIPCIÓN DEL PROCESO.

En la Figura 3.1. se presenta el diagrama de flujo del proceso Tennessee Eastman. El resultado del proceso son dos productos (G, H) a partir de los reactivos, (A, C, D, E), además de un inerte (B), y un subproducto (F), dando a lugar a un total de ocho componentes. Las reacciones que tienen lugar durante el proceso son las siguientes: [25]



Todas las reacciones que suceden son irreversibles y exotérmicas, y su equilibrio depende de la temperatura. La reacción que da lugar al producto G

tiene una energía de activación mayor por lo que su sensibilidad frente a la temperatura será mayor. Las reacciones son de primer orden respecto a la concentración de los reactivos.

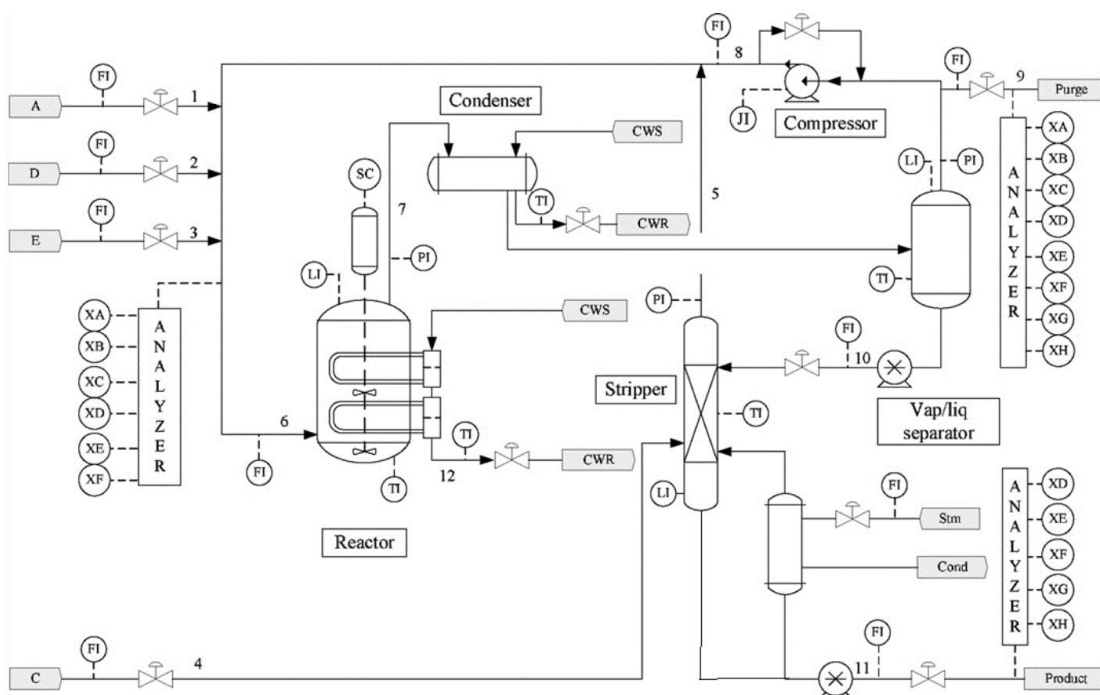


Figura 3.1. Diagrama de Flujo del Proceso Tennessee Eastman. Fuente: Adaptación [25]

En el proceso contamos con cinco operaciones clave producidas en el reactor, el condensador, el separador de líquido y gas, el compresor y en el destilador. En el proceso también encontramos diferentes elementos como bombas, además de elementos de control como válvulas, cromatógrafos de gases (analizadores), indicadores, etc.

Los reactivos gaseosos A, D y E entran en el reactor, las reacciones en fase gaseosa son catalizadas por la introducción de un catalizador no volátil no disuelto, que permanecerá en el reactor. En el reactor también tiene lugar el enfriamiento a fin agilizar la reacción mediante la eliminación del calor de reacción. Los productos salen del reactor como vapor junto con los productos no reaccionados y se conducen hacia un condensador y posteriormente a un separador líquido-vapor. Los no condensables son recirculados mediante un



compresor centrífugo, para volver junto con los reactivos A, D y E que alimentan el reactor. [25]

Los condensables son redirigidos a una columna de destilación donde se eliminan los reactivos remanentes provenientes del reactor mediante stripping con la adición del reactivo C. La salida de base de la columna de destilación son los productos G y H los cuales son separados aguas abajo mediante una sección de refinamiento, la cual no ha sido objeto de estudio en el proceso. El inerte B producido y el subproducto F son purgados en la corriente de vapor del separador líquido-vapor.

### **3.2. DATOS DE LA PLANTA.**

El proceso cuenta con un total de 12 variables manipuladas y 41 variables medidas, tablas 1 y 2 respectivamente. De las 41 variables medidas, 22 se monitorizan de manera continua (XMEAS (1-22)). Las otras 19 se obtiene de los analizadores que encontramos distribuidos en la planta que dan valores solo cada cierto intervalo de tiempo (XMEAS (23-42)). Las 12 variables manipuladas presentes en el proceso TE ofrecen al diseñador 12 grados de libertad correspondientes a las 9 válvulas de flujo, 2 válvulas de control de temperatura y una velocidad de agitación.

En esta planta se han provocado distintos tipos de fallos como se muestra en la Tabla 3. Los fallos están distribuidos a lo largo de todo el proceso, encontramos fallos en las entradas al reactor, en el condensador, etc. En la Tabla 3 también se indica el tipo de fallo producido.



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos



<b>Variable</b>	<b>Variable</b>	<b>Unidades</b>
XMV (1)	Flujo de alimentación D	$\text{Kgh}^{-1}$
XMV (2)	Flujo de alimentación E	$\text{Kgh}^{-1}$
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	$\text{m}^3 \text{h}^{-1}$
XMV (8)	Flujo de líquido de la columna de stripping	$\text{m}^3 \text{h}^{-1}$
XMV (9)	válvula de vapor de la columna de stripping	%
XMV (10)	Flujo de agua de refrigeración del reactor	$\text{m}^3 \text{h}^{-1}$
XMV (11)	Flujo de agua en el condensador	$\text{m}^3 \text{h}^{-1}$
XMV (12)	Velocidad del agitador del reactor	rpm

**Tabla 1.** Variables manipuladas del proceso. Fuente: Adaptación [25]



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos



Variable	Variable	Unidades
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 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)	Kgh <sup>-1</sup>
XMEAS (20)	Potencia de compresor	kW
XMEAS (21)	Temperatura de la salida del agua de refrigeración del reactor	°C
XMEAS (22)	Temperatura de la salida del agua de refrigeración del separador	°C
XMEAS (23-28)	Concentración de la alimentación del reactor (A-F)	% mol
XMEAS (29-36)	Concentración de la purga (A-H)	% mol
XMEAS (37-41)	Concentración aguas abajo del destilador (A-H)	% mol

Tabla 2. Variables medidas del proceso. Fuente: Adaptación [25]



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos



Fallo	VARIABLES DE FALLO	Tipo de Fallo
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 del agua del refrigerante del reactor	Escalón
IDV (5)	Temperatura de entrada del agua refrigerante del 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 del agua refrigerante al reactor	Variación aleatoria
IDV (12)	Temperatura de entrada del agua de refrigerante del condensador	Variación aleatoria
IDV (13)	Cinética de las reacciones	Variación lenta
IDV (14)	Válvula del agua de refrigerante del reactor	Bloqueo
IDV (15)	Válvula del agua de 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 del proceso. Fuente: Adaptación [25]



En estudios anteriores se ha probado que encontramos fallos más difíciles de detectar que otros debido a su naturaleza imperceptible en las variables de salida. Estos fallos son también llamados *incipientes*. Los fallos IDV3, IDV9 e IDV15 resultaron ser fallos incipientes de muy difícil detección y clasificación.

Para el estudio de la planta en este trabajo hemos trabajado con dos tipos de datos diferentes, los primeros, un conjunto de datos para el entrenamiento y el test con una extensión relativamente pequeña, perteneciente a una única simulación, los segundos, son matrices de metadatos para el entrenamiento y test de la planta, con 500 simulaciones, siendo así quinientas veces los datos usados inicialmente.

Los datos de la Planta Tennessee Eastman son públicos y pueden ser descargados desde varias fuentes en la documentación académica aplicada al problema. Los datos de entrenamiento y test se han descargado de (<http://web.mit.edu/braatzgroup/links.html>) y los metadatos de (<https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/6C3JR1>)



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos







## CAPÍTULO IV: APLICACIÓN



#### 4.1. IMPLEMENTACIÓN PRÁCTICA

Para la simulación del modelo propuesto en el Capítulo III de este trabajo se han descargado ficheros de datos de fallos del Proceso Tennessee Eastman (TEP). Estos ficheros ofrecen veintidós simulaciones que recogen datos para el funcionamiento normal de la planta, y para cada uno de los 21 fallos recogidos en la documentación relativa al proceso.

Encontramos dos bloques de ficheros para cada uno de los fallos. El primero corresponde con datos de entrenamiento que utilizaremos para probar las diferentes metodologías de aprendizaje automático. Estos datos de entrenamiento cuentan con 960 observaciones para cada una de las 52 variables analizadas, las primeras 160 observaciones registran datos del comportamiento normal, es decir, previo al fallo. De las 800 observaciones de fallo restantes, desde la fila 161 hasta la 960, las primeras 650 serán utilizadas para el entrenamiento y aprendizaje, y las 150 observaciones restantes serán utilizadas para comprobar la solidez de la respuesta, lo que denominaremos validación.

El segundo fichero son datos online de la planta, que nos sirven para testear el funcionamiento del algoritmo clasificador. Estos ficheros de datos cuentan con 480 observaciones para cada una de las 21 situaciones de fallo de las 52 variables medidas en el proceso.

Es importante destacar que los datos de los siguientes fallos:

- IDV (3)            Temperatura de alimentación D
- IDV (9)            Temperatura de alimentación D
- IDV (15)          Válvula del agua de refrigerante del condensador.

Nombrados *incipientes* en el capítulo anterior debido a su naturaleza imperceptible, han sido excluidos de estudio para conseguir un resultado satisfactorio en la clasificación.

El fallo número 6, IDV (6) – Pérdida de alimentación de A – ha sido omitido en el estudio debido a falta de datos de validación.



Todas las técnicas aplicadas a lo largo del presente trabajo, los datos extraídos, así como las graficas y matrices de confusión obtenidos han sido desarrollados en el software MATLAB™.

## 4.2. DETECCIÓN DE FALLOS MEDIANTE TÉCNICAS DE ANÁLISIS MULTIVARIANTE. ANÁLISIS DE COMPONENTES PRINCIPALES.

Para la aplicación del Análisis de Componentes Principales (PCA), primero vamos a realizar los cálculos iniciales fuera de línea, tanto el cálculo de las componentes principales que reduzcan el espacio muestral de trabajo, como los estadísticos indicados en el Capítulo II, que serán utilizados como umbrales o límites estadísticos. Posteriormente se realiza la validación del método con los diferentes ficheros de datos de fallo online de la planta.

### 4.2.1. FUERA DE LÍNEA.

Partimos de los datos fuera de línea, aplicaremos la metodología descrita en el marco teórico partiendo del fichero de datos de comportamiento normal de la planta fuera de línea. Partimos de la matriz  $X$ , de dimensiones  $500 \times 52$ . El espacio  $\mathbb{R}^{52 \times 500}$  pretende ser reducido a un espacio  $\mathbb{R}^{a \times 500}$ , donde  $a$  sean las componentes principales con una variabilidad y porcentaje de falsas alarmas aceptados.

Se normaliza la matriz  $X$  a media cero y varianza uno. Se obtienen la matriz de correlaciones  $R$  a partir de  $X$  (Ecuación 6) y se extraen los valores singulares en orden decreciente de  $R$  que se almacenan en la matriz  $\Lambda$ , y la matriz de vectores propios  $V$  de  $R$ , también llamada de vectores de carga (Ecuación 7). Se ha estimado una variabilidad del 90%, a partir del cual calcularemos la matriz  $P \in \mathbb{R}^{a \times 52}$ , con los vectores de carga correspondientes a los valores singulares con una variabilidad acumulada igual o mayor a la aceptada. Con la variabilidad indicada, se han obtenido un total de 31 componentes principales.



A partir de  $P$  podemos calcular la proyección de los datos de partida  $X$  en un espacio reducido al que denominamos  $T$  (Ecuación 8), pasando de un espacio inicial en  $\mathfrak{R}^{500 \times 52}$  a su proyección reducida en  $\mathfrak{R}^{500 \times 31}$ .

La matriz de ruido o residuos se calcula a partir del espacio de partida y el espacio actual (Ecuación 10).

Para establecer los límites de actuación y umbrales debemos calcular los estadísticos  $T^2$  y  $Q$ , que nos permiten la monitorización del estado del proceso. La estadística de Hotellings  $T^2$  se calcula a partir de los vectores de observación  $x$  (Ecuación 12) y el umbral  $T_{\lambda}^2$  con los datos fuera de línea (Ecuación 13). La estadística  $Q$  se calcula a partir del vector de observación  $x$  (Ecuación 14) previo cálculo de los vectores de residuos  $r$  (Ecuación 15). El umbral de  $Q_{\alpha}$  se calcula mediante la distribución estadística  $X^2$  (Ecuación 16).

#### 4.2.2. EN LÍNEA.

Una vez que han sido calculados los datos del PCA, se puede monitorizar el proceso, y realizar la detección de fallos. Para ello con los datos online de la planta, es decir, datos reales de cada fallo, podemos calcular los estadísticos  $T^2$  y  $Q$  para compararlos con los umbrales calculados fuera de línea. Se considera que se incurre en fallo cuando el umbral es sobrepasado, no obstante, para evitar falsas alarmas, consideraremos que entramos en situación de fallo cuando el umbral haya sido sobrepasado  $n$  veces consecutivas, en esta ocasión se ha estimado que sean un total de 10 datos consecutivos por encima del límite para declarar estado de fallo.

Los resultados ofrecidos por el PCA para la detección de los diferentes fallos se pueden apreciar en los siguientes gráficos. Los diferentes valores de la estadística  $T^2$  y  $Q$  vienen definidos por puntos unidos entre sí (línea azul), por otro lado, el valor constante del umbral obtenido  $T_{\lambda}^2$  y  $Q_{\alpha}$  viene definido por la línea continua roja.

En la Figura 4.1 podemos observar el grafo de control del PCA online para el Fallo 2 “Composición de B con relación A/C constante”. Las primeras 160



observaciones corresponden con el funcionamiento normal de la planta previa al fallo, y podemos observar como no supera el umbral establecido, ni  $T_\lambda^2$  ni  $Q_\alpha$  en dichas observaciones. Las siguientes 800 observaciones apreciamos claramente que superan el límite establecido de control, además vemos que la detección del fallo se produce en un corto periodo de tiempo.

En la Figura 4.2 se muestran los resultados para la detección del Fallo 14 “Válvula del agua de refrigerante del reactor”. Como sucedía con el Fallo 2, las primeras 160 observaciones de comportamiento normal no indican presencia de fallo, mientras que a partir de la observación 161 los valores de los estadísticos sobrepasan los umbrales establecidos.

En la Figura 4.3 se muestra la detección del Fallo 9 “Temperatura del reactor”. Como se puede apreciar, el umbral no es superado en la mayoría de las observaciones, por lo que el Fallo 9 no es detectado correctamente. En ningún momento se producen 10 puntos consecutivos por encima del umbral. El Fallo 9 había sido denominado anteriormente como fallo incipiente por estudios realizados anteriormente.

Los fallos detectados mostrados hasta ahora han ofrecido unos tiempos de detección relativamente cortos, tanto el Fallo 2 (Figura 4.1) como el Fallo 14 (Figura 4.2) eran detectados prácticamente tras las 160 observaciones de comportamiento normal. En la observación 170 ya ofrecían las 10 alertas necesarias para determinar el estado de fallo.

En la Figura 4.4 se muestra el Fallo 21 “Desconocido”, el cual tiene un tiempo de detección considerablemente mayor que los fallos anteriormente mostrados.



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos

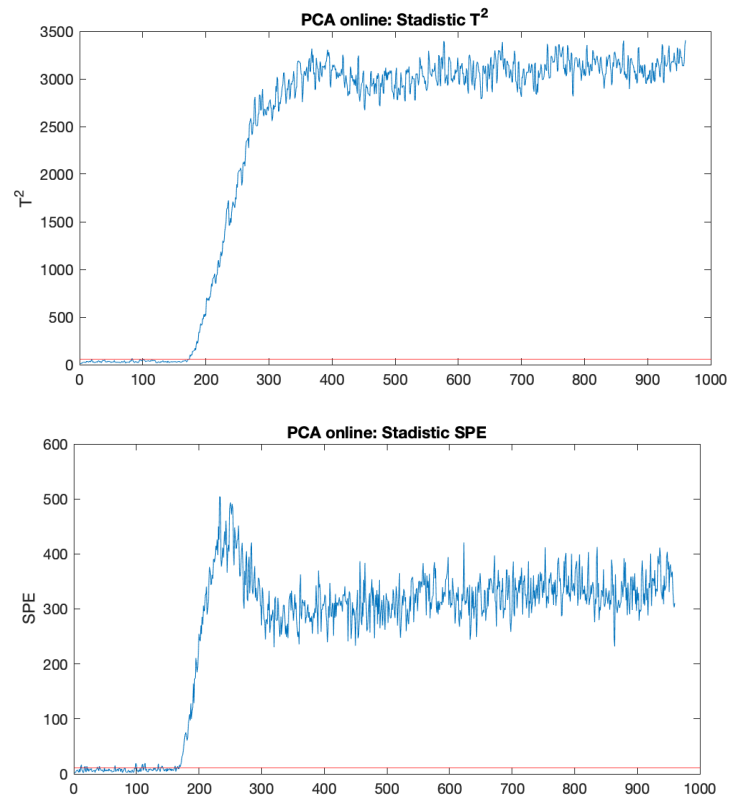


Figura 4.1. (a) Arriba. Gráfico de control  $T^2$  para el Fallo 2. (b) Abajo. Gráfico de control Q-SPE para el Fallo 2.

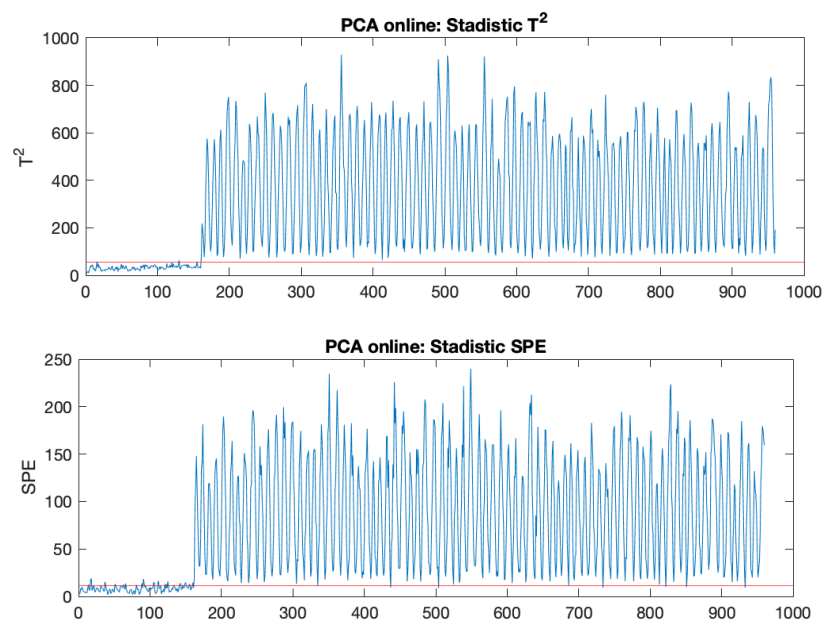


Figura 4.2. (a) Arriba. Gráfico de control  $T^2$  para el Fallo 14. (b) Abajo. Gráfico de control Q-SPE para el Fallo 14.



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos

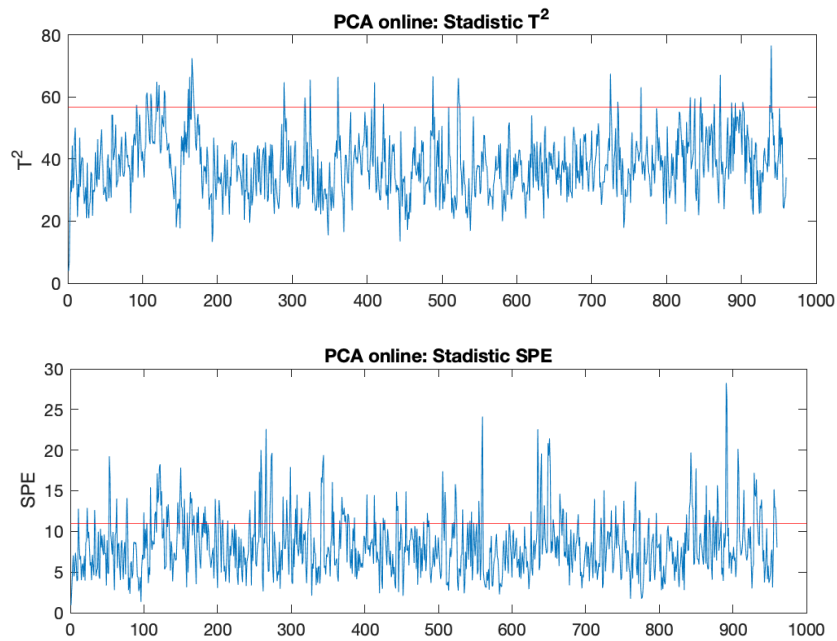


Figura 4.3. (a) Arriba. Gráfico de control  $T^2$  para el Fallo 9. (b) Abajo. Gráfico de control Q-SPE para el Fallo 9.

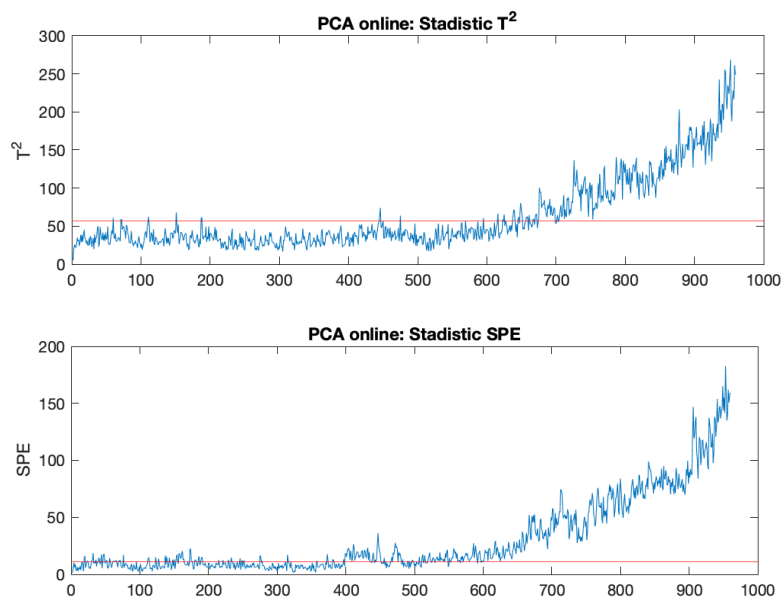


Figura 4.4. (a) Arriba. Gráfico de control  $T^2$  para el Fallo 21. (b) Abajo. Gráfico de control Q-SPE para el Fallo 21.



### 4.3. DIAGNÓSTICO DE FALLOS MEDIANTE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO.

Para el problema de Diagnóstico y Clasificación de Fallos vamos a utilizar métodos de clasificación basados en el aprendizaje automático. La metodología que se va a seguir para todos los procesos de aprendizaje ha sido ya especificada en el Capítulo II. Se experimentará la clasificación de los fallos mediante redes neuronales de propagación hacia adelante y redes recurrentes, en particular redes LSTM. Modificando las arquitecturas de las redes, número de capas ocultas, número de neuronas por capa, etc. Se pretende comparar el acierto en la clasificación para diferentes situaciones, también en función del número de fallos introducidos en la red.

Los resultados obtenidos serán plasmados en matrices de confusión, capaces de ofrecernos resultados intuitivos sobre la eficacia en la clasificación de las clases introducidas en la red.

#### 4.3.1. REDES NEURONALES DE PROPAGACIÓN HACIA ADELANTE

En este primer apartado se describe el uso de ANN más simples, las redes neuronales de propagación hacia adelante. Se ha experimentado en diferentes situaciones en función de la arquitectura de la red, y del número de clases, fallos introducidos. Con respecto a los fallos se han generado cinco situaciones con diferentes fallos (Tabla 1):

5	IDV(1), IDV(2), IDV(4), IDV(5), IDV(7).
7	IDV(1), IDV(2), IDV(4), IDV(5), IDV(7), IDV(8), IDV(10).
10	IDV(1), IDV(2), IDV(4), IDV(5), IDV(7), IDV(8), IDV(10), IDV(11), IDV(12), IDV(13).
13	IDV(1), IDV(2), IDV(4), IDV(5), IDV(7), IDV(8), IDV(10), IDV(11), IDV(12), IDV(13), IDV(14), IDV(16), IDV(17).
17	IDV(1), IDV(2), IDV(4), IDV(5), IDV(7), IDV(8), IDV(10), IDV(11), IDV(12), IDV(13), IDV(14), IDV(16), IDV(17), IDV(18), IDV(19), IDV(20), IDV(21).

Tabla 1.- Fallos de las diferentes situaciones propuestas.





#### 4.3.1.1. SALIDAS DE LA ANN

Primero se realizará un estudio en función del tipo de salida  $Y$  propuesta a la red. A fin de obtener mejores resultados se han propuesto dos tipos de salida para las redes neuronales.

##### Salida tipo I: Única escalonada

La salida tipo escalonada, en función del número de clases, fallos, consiste en una única salida de la red, que adquiere el valor nominal del fallo al que representa. Por lo tanto  $Y$  será un vector con dimensión igual a  $1 \times \alpha$ , donde  $\alpha$  es el resultado del producto de los  $N$  fallos introducidos en la red y las  $m$  observaciones por cada fallo. Por tanto, para el entrenamiento de la red, donde son utilizadas 650 observaciones por fallo, y  $N$  fallos, el vector salida escalonada tendrá dimensiones  $(1 \times 650N)$ .

Por ejemplo, en la figura 4.5 se muestra la salida escalonada deseada del entrenamiento para  $N = 15$  Fallos. Así los primeros 650 valores para el entrenamiento pertenecientes al primer fallo introducido, tienen un valor de 1, los siguientes 650 valores, pertenecientes al fallo introducido en segundo lugar, adquiere valor 2, y así sucesivamente.

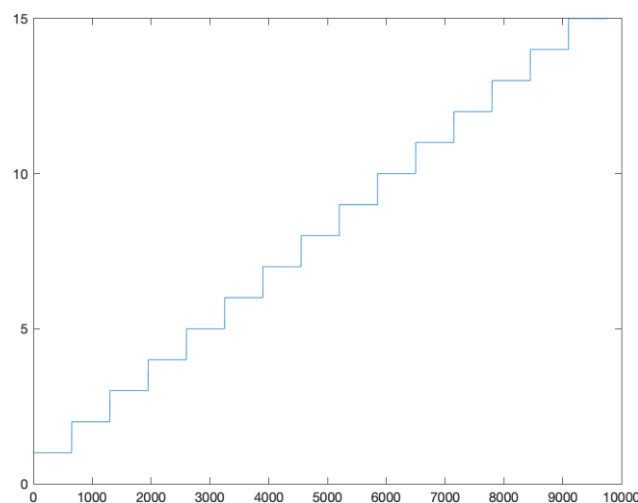


Figura 4.5. Salida Escalonada para 15 Fallos



### Salida tipo II: Múltiple binaria.

El segundo tipo de salida será una matriz binaria  $Y$  compuesta por tantas salidas como clases, o fallos, se pretendan introducir en la red. De este modo,  $Y$  tendrá unas dimensiones de  $(N \times m)$  donde  $N$  son el número de fallos y  $m$  las observaciones por fallo. Así, la capa de salida de la red neuronal tendrá un total de  $N$  neuronas, que adquirirán un valor binario en función del fallo a clasificar.

En la Figura 4.6 se puede observar los vectores de la salida deseada  $y \in \mathbb{R}^{4 \times m}$  para los primeros cuatro fallos.

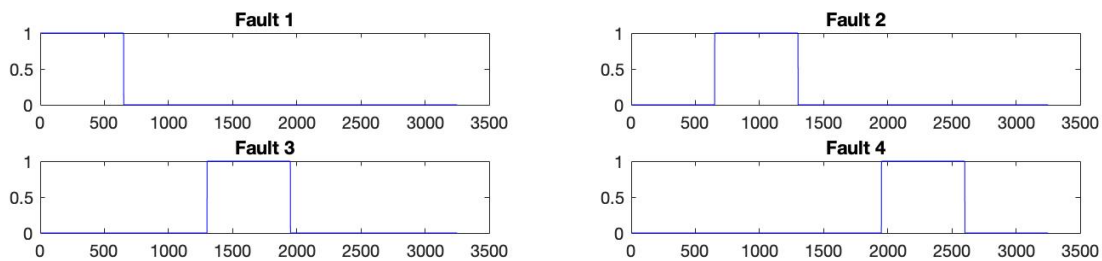


Figura 4.6. Salida múltiple binaria para los primeros 4 Fallos

Para el primer vector  $y_{(1)}$  los componentes del 1 al 650 toman valor uno, el restante de los valores tiene valor igual a cero. Para el vector  $y_{(2)}$  los componentes desde 651 hasta el 1300 toman valor uno, el resto cero, y así sucesivamente.

La matriz de datos de entrada se forma en función de los datos de fallo introducidos. Por ejemplo, para la clasificación de cinco fallos, la matriz de entrada a la red  $X_E$  con los fallos (1, 2, 3, 4 y 5) con 650 observaciones para el entrenamiento (observaciones desde 161 a 810) tendrá unas dimensiones de  $52 \times 650 \cdot 5$ . Por tanto, la matriz de entrada a la red  $X_E \in \mathbb{R}^{52 \times 3250}$ . Esta matriz será utilizada para el entrenamiento de la red.

A continuación, se realiza una validación con las 150 observaciones restantes (observaciones desde 811 a 960). Lo que provoca que la matriz de entrada en la validación  $X_V$  tenga unas dimensiones de  $52 \times 150 \cdot 5$ .

Y finalmente, se pretende probar la red con los datos de test de la planta, con un total de 480 observaciones. Las dimensiones de la matriz de entrada para el test serán  $52 \times 480 \cdot 5$ .

Las matrices resultantes de los tres procesos, entrenamiento, validación y test, para el ejemplo de cinco fallos son las siguientes:

- Matriz para el entrenamiento:  $X_E \in \mathfrak{R}^{52 \times 3250}$
- Matriz para la validación:  $X_V \in \mathfrak{R}^{52 \times 750}$
- Matriz para el test:  $X_T \in \mathfrak{R}^{52 \times 2400}$

Las matrices de entrada a la red deben ser normalizadas, en el primer proceso de entrenamiento, la matriz de entrada  $X_E$  es normalizada entre -1 y 1. La normalización posterior de las matrices  $X_V$  y  $X_T$  se realiza con los valores máximos y mínimos obtenidos con la normalización de  $X_E$ . Este procedimiento se repite para todos los procesos posteriores.

Una vez hemos creado las matrices de entrada y salida, se debe determinar la arquitectura de la red neuronal que utilizaremos para el estudio, como ya se ha adelantado, la red neuronal utilizada para este estudio será una red neuronal con varias capas. Se pretende comparar el acierto de la red para los dos tipos de salidas, en función de los fallos introducidos en la red. Es lógico esperar resultados óptimos para un número de fallos reducidos, y una pérdida de eficacia en la clasificación a medida que se incrementen el número de clases, fallos para la clasificación.

En la Figura 4.7 se observa la arquitectura de la ANN para una salida única y escalonada tipo I, y en la Figura 4.8 la de la red neuronal con salida tipo II.

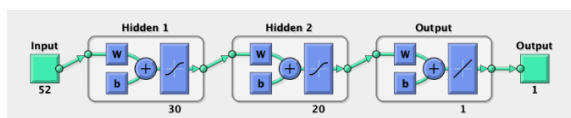


Figura 4.7. Red Neuronal de propagación hacia adelante con salida Tipo I para 17 Fallos

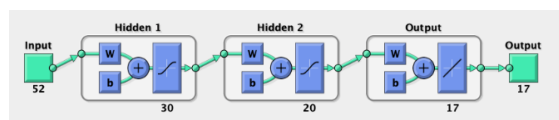


Figura 4.8. Red Neuronal de propagación hacia adelante con salida Tipo II para 17 Fallos



En la Tabla 2 se indica el acierto o eficacia en la clasificación de una red neuronal con dos capas ocultas, con 30 y 20 neuronas respectivamente. El acierto corresponde con el acierto global en la clasificación obtenido mediante las matrices de confusión, como se observa en la Figura 4.9 donde se representa la matriz de confusión para el test de 10 fallos, y salida tipo II.

**Test: Plot Confusion**

Output Class	1	2	3	4	5	6	7	8	9	10	Accuracy	Loss
1	467 9.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 0.1%	6 0.1%	0 0.0%	0 0.0%	0 0.0%	97.3%	2.7%
2	0 0.0%	459 9.6%	0 0.0%	0 0.0%	0 0.0%	8 0.2%	13 0.3%	0 0.0%	0 0.0%	0 0.0%	95.6%	4.4%
3	0 0.0%	0 0.0%	478 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	99.6%	0.4%
4	0 0.0%	0 0.0%	0 0.0%	455 9.5%	0 0.0%	0 0.0%	17 0.4%	0 0.0%	8 0.2%	0 0.0%	94.8%	5.2%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	479 10.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	99.8%	0.2%
6	40 0.8%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	372 7.8%	41 0.9%	1 0.0%	7 0.1%	18 0.4%	77.5%	22.5%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 0.1%	467 9.7%	5 0.1%	2 0.0%	3 0.1%	97.3%	2.7%
8	0 0.0%	0 0.0%	49 1.0%	0 0.0%	0 0.0%	0 0.0%	58 1.2%	373 7.8%	0 0.0%	0 0.0%	77.7%	22.3%
9	0 0.0%	0 0.0%	1 0.0%	7 0.1%	0 0.0%	14 0.3%	59 1.2%	9 0.2%	390 8.1%	0 0.0%	81.2%	18.8%
10	8 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	134 2.8%	73 1.5%	12 0.2%	12 0.2%	241 5.0%	50.2%	49.8%
	90.7% 9.3%	100% 0.0%	90.5% 9.5%	98.5% 1.5%	99.8% 0.2%	69.1% 30.9%	63.6% 36.4%	92.8% 7.2%	92.9% 7.1%	92.0% 8.0%	87.1% 12.9%	
	1	2	3	4	5	6	7	8	9	10		
	<b>Target Class</b>											

Figura 4.9. Matriz de confusión para el test. (10 fallos) con salida tipo II.

Nº Fallos	Salida Tipo I			Salida Tipo II		
	Train	Validation	Test	Train	Validation	Test
5	99,8%	100%	97,8%	100%	100%	99,7%
7	91,1%	82,2%	80,3%	98,8%	97,8%	95,7%
10	61,0%	58,7%	50,2%	94,6%	91,3%	87,1%
13	46,4%	40,4%	35,6%	86,0%	81,0%	78,8%
17	24,0%	23,7%	17,9%	69,3%	70,7%	61,1%

Tabla 2.- Eficacia de clasificación de la ANN con dos capas ocultas (30, 20 neuronas)



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos

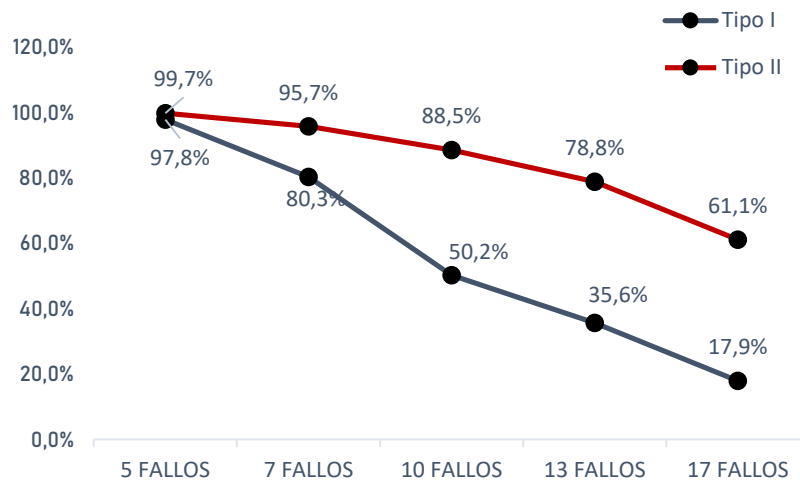


Figura 4.10. Comparativa de la eficacia en la respuesta de la ANN con salida Tipo I vs. Tipo II para el test.

Los resultados reflejan una gran pérdida de eficacia en la clasificación para un mayor número de fallos cuanto utilizamos la salida tipo I. Uno de los grandes inconvenientes es la necesidad de aproximar la respuesta al entero más próximo, lo que provoca la aparición de clases ficticias que provocan error en la clasificación. La Figura 4.11 muestra la respuesta de la red para la clasificación de 10 fallos, se puede observar como la respuesta ofrece valores negativos, o mayores a 10 que no pertenecen a ninguna clase. Por otro lado, el mejor comportamiento del segundo tipo de salida se puede ver en la Figura 4.12, donde se muestra la salida de la red con 10 fallos.

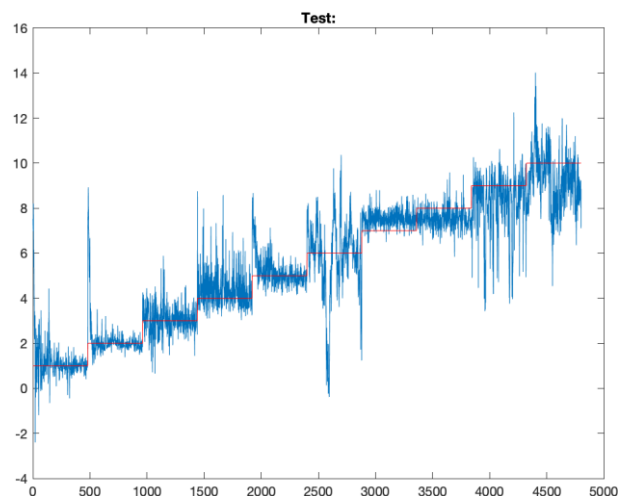


Figura 4.11. Respuesta de la ANN con salida única Tipo I para el test (10 fallos). Eficacia de 50,2%

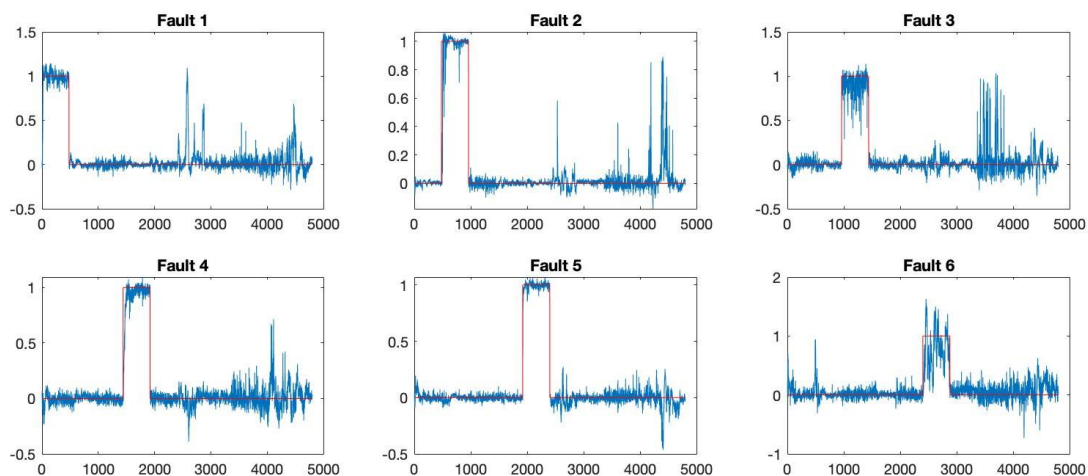


Figura 4.12. Respuesta de la ANN con salida Tipo II para el test (10 fallos).

#### 4.3.1.2. ARQUITECTURAS DE LAS ANN.

En este apartado, y a la vista de los resultados anteriores, se experimenta con la salida múltiple denominada Tipo II. Se pretende realizar una comparación en el comportamiento de la red de propagación hacia adelante en función de su arquitectura, tanto número capas como de neuronas por capa, y del número de fallos a clasificar.

##### Una capa.

La primera experimentación se realiza para una ANN con una única capa. Se han registrado los porcentajes de eficacia en la clasificación en situaciones diferentes, variando el número de neuronas por capa. Se han propuesto cuatro escenarios, con 30, 50, 100, y por último, 150 neuronas en la capa única de la ANN. En la Tabla 3 se presentan los resultados en la eficacia tanto en el entrenamiento, validación y test en la clasificación para las diferentes situaciones estimadas en el apartado anterior (5, 7, 10, 13 y 17 fallos).

Los resultados ofrecidos muestran un gran comportamiento en la clasificación cuando se trata de menos de 7 fallos. Cuando aumentamos el número de fallos, la pérdida de eficacia es significativa.

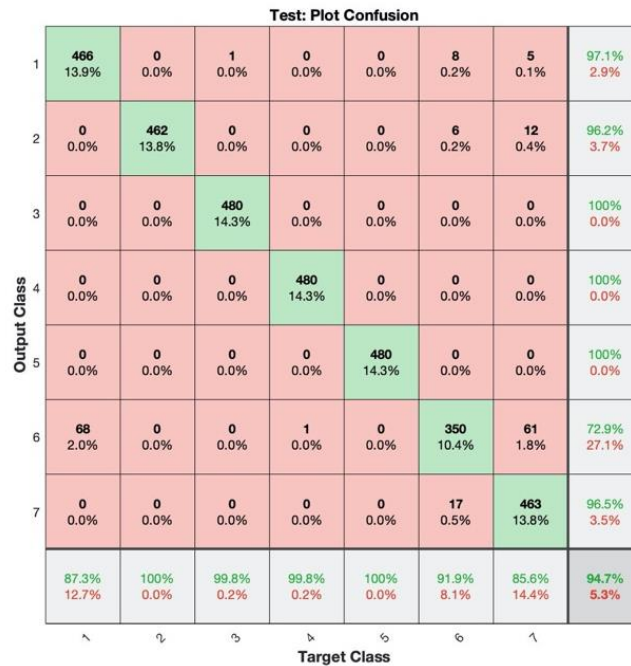


## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos



Nº Neuronas	[30]			[50]		
Nº de Fallos	Train	Validation	Test	Train	Validation	Test
5	99,90%	100,00%	99,50%	100%	100,00%	99,50%
7	98,40%	98,20%	95,10%	98,50%	97,10%	94,70%
10	92,40%	89,10%	87,00%	93,40%	92,40%	86,80%
13	81,30%	76,90%	76,70%	84,40%	79,30%	78,00%
17	68,30%	70,10%	62,40%	72,60%	73,60%	64,70%
Nº Neuronas	[100]			[150]		
Nº de Fallos	Train	Validation	Test	Train	Validation	Test
5	100%	100,00%	99,80%	100%	100,00%	99,60%
7	98,90%	97,70%	96,80%	98,90%	97,20%	95,50%
10	94,20%	91,90%	88,70%	94,50%	92,50%	88,00%
13	85,60%	80,60%	77,90%	86,50%	81,60%	79,40%
17	76,20%	75,30%	67,30%	76,50%	75,00%	65,70%

**Tabla 3.** Eficacia de la ANN con una capa en función del número de neuronas y fallos



**Figura 4.13.** Matriz de confusión para el test. (7 fallos).



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos



Test: Plot Confusion

Output Class	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	Accuracy		
0	455 5.6%	0	0	1	15	0	0	0	0	0	0	0	0	0	0	0	0	8	1	94.8%	
1	0	463 5.7%	0	0	1	2	2	0	0	0	0	0	0	0	0	0	0	0	8	4	96.5%
2	0	0	479 5.9%	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	99.8%	
3	0	0	0	440 5.4%	0	7	0	0	22	3	0	0	0	0	2	3	3	3	3	91.7%	
4	0	0	0	0	477 5.8%	0	0	0	0	0	0	0	0	0	0	0	0	3	3	99.4%	
5	82 1.0%	1 0.0%	2 0.0%	1 0.0%	4 3.7%	6 0.1%	2 0.0%	3 0.0%	32 0.4%	1 0.0%	5 0.1%	0 0.0%	4 0.0%	22 0.3%	12 0.1%	12 0.1%	12 0.1%	22 0.3%	12 0.1%	63.1%	
6	1 0.0%	0 0.0%	16 0.2%	19 0.2%	16 0.2%	5 0.1%	46 0.6%	28 0.3%	13 0.2%	8 0.1%	0 0.0%	120 1.5%	0 0.0%	0 0.0%	0 0.0%	86 1.1%	62 0.8%	60 0.7%	60 0.7%	9.6%	
7	0	0	135 1.7%	10 0.1%	6 0.1%	0	21 0.3%	9 0.1%	2	0	190 2.3%	17 0.2%	1	0	35 0.4%	12 0.1%	42 0.5%	42 0.5%	42 0.5%	1.9%	
8	1 0.0%	0	8 0.1%	33 0.4%	14 0.2%	28 0.3%	18 0.2%	5 0.1%	250 3.1%	22 0.3%	1	29 0.4%	0	7 0.1%	21 0.3%	25 0.3%	18 0.2%	18 0.2%	18 0.2%	52.1%	
9	10 0.1%	0	1 0.0%	3 0.0%	8 0.1%	56 0.7%	1	4	2	303 3.7%	0	44 0.5%	0	0	24 0.3%	17 0.2%	7 0.1%	7 0.1%	7 0.1%	63.1%	
10	0	0	21 0.3%	1 0.0%	0	0	0	0	8	0	0	376 4.6%	3 0.0%	61 0.7%	0	6 0.1%	1 0.0%	3 0.0%	3 0.0%	78.3%	
11	0	0	7 0.0%	31 0.4%	9 0.1%	2 0.0%	20 0.2%	20 0.2%	2	1	0	261 3.2%	0	0	46 0.6%	19 0.2%	62 0.8%	62 0.8%	62 0.8%	54.4%	
12	0	0	58 0.7%	2 0.0%	5 0.1%	0	18 0.2%	4	0	3	11 0.1%	3	328 4.0%	0	22 0.3%	9 0.1%	17 0.2%	17 0.2%	17 0.2%	68.3%	
13	1 0.0%	0	4 0.0%	32 0.4%	4 0.0%	0	7 0.1%	7	25	0	0	7	0	341 4.2%	19 0.2%	8 0.1%	25 0.3%	25 0.3%	25 0.3%	71.0%	
14	1 0.0%	0	2 0.0%	9 0.1%	9 0.1%	0	14 0.2%	6	5	0	0	16 0.2%	0	0	302 3.7%	73 0.9%	43 0.5%	43 0.5%	43 0.5%	62.9%	
15	0	0	4 0.0%	9 0.1%	4 0.0%	2	15 0.2%	11	2	1	0	9	0	0	22 0.3%	360 4.4%	41 0.5%	41 0.5%	41 0.5%	75.0%	
16	0	0	8 0.0%	17 0.2%	0	2	31 0.4%	42 0.5%	16 0.2%	7	1	120 1.5%	0	0	95 1.2%	57 0.7%	84 1.0%	84 1.0%	84 1.0%	17.5%	
17	82.6%	99.8%	64.3%	72.5%	85.5%	71.8%	23.1%	6.2%	73.1%	79.7%	64.7%	41.2%	84.1%	98.0%	43.6%	53.3%	19.8%	64.7%	17.4%	0.2%	
	17.4%	0.2%	35.7%	27.5%	14.5%	28.2%	78.9%	93.8%	26.9%	20.3%	35.3%	58.8%	15.9%	2.0%	56.4%	46.7%	80.2%	85.3%			

Figura 4.14. Matriz de confusión para el test. (17 fallos).

### Multicapa.

Para intentar mejorar la respuesta de la ANN se ha aumentado el número de capas ocultas de la red. Se han hecho pruebas con dos y con tres capas ocultas, variando el número de neuronas por capa. Los resultados se presentan en la Tabla 4 y Tabla 5.

Nº Neuronas	[30 20]			[50 30]		
	Train	Test	Validation	Train	Test	Validation
5	100%	100,00%	99,50%	100%	100,00%	99,70%
7	99,00%	97,00%	96,20%	99,30%	98,20%	95,70%
10	95,40%	92,50%	88,60%	95,50%	93,50%	88,40%
13	85,0%	80,40%	77,20%	86,30%	79,90%	78,70%
17	65,70%	65,70%	66,10%	72,50%	72,80%	63,30%

Tabla 4-1. Eficacia de la ANN con dos capas en función del número de neuronas y fallos





## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos



Nº Neuronas	[100 50]			[200 100]		
Nº de Fallos	Train	Test	Validation	Train	Test	Validation
5	99,90%	100,00%	99,50%	99,90%	100,00%	98,50%
7	99,30%	97,80%	96,40%	99,30%	94,40%	92,70%
10	96,80%	91,00%	87,60%	96,00%	86,10%	84,10%
13	90,20%	83,10%	82,10%	90,40%	82,20%	80,00%
17	80,20%	80,20%	72,30%	74,70%	76,00%	62,70%

**Tabla 4-2.** Eficacia de la ANN con dos capas en función del número de neuronas y fallos

Nº Neuronas	[30 50 20]			[50 60 25]		
Nº de Fallos	Train	Test	Validation	Train	Test	Validation
5	100%	100,00%	99,50%	100%	100,00%	99,40%
7	99,10%	98,50%	96,20%	99,10%	97,70%	95,80%
10	96,90%	89,10%	87,60%	96,90%	91,40%	87,80%
13	89,30%	80,10%	79,80%	94,30%	89,40%	86,20%
17	67,80%	67,70%	58,20%	75,60%	77,50%	65,20%

Nº Neuronas	[100 120 50]			[200 240 100]		
Nº de Fallos	Train	Test	Validation	Train	Test	Validation
5	99,90%	100,00%	99,40%	99,70%	100,00%	98,60%
7	99,20%	97,80%	96,00%	99,20%	96,50%	95,10%
10	96,60%	89,60%	88,30%	96,40%	87,30%	82,00%
13	96,70%	85,90%	85,20%	79,60%	70,10%	70,20%
17	84,50%	81,10%	73,40%	61,00%	59,20%	49,90%

**Tabla 5.** Eficacia de la ANN con tres capas en función del número de neuronas y fallos

Los resultados obtenidos para una, dos y tres capas en una red neuronal de propagación hacia adelante (Tablas 3, 4 y 5) se resumen en la Tabla 6 para los datos únicamente de Test. Se muestra un aumento de la eficacia con tres capas comparándolo con el caso de una única capa. Además, parece evidente que los resultados con un número de neuronas de orden 100 (Columna 3) por lo general



ofrecen mejores resultados que las de orden 30 o 50 (Columna 1 y 2 respectivamente). También parece significativo que los mejores resultados los encontramos siempre en la Columna 3 y se observa una pérdida de eficacia cuando duplicamos la cantidad de neuronas (datos de la Columna 4). La causa de estos resultados pueden ser el sobreajuste u *overfitting* que se mencionaba en el Capítulo II. La cantidad de neuronas debe ir acorde a los datos introducidos en la red, si la dimensión de la red no es suficientemente grande, corremos el riesgo de no capacitar a la red con un aprendizaje lo suficientemente bueno con los datos que contamos, pero, por otro lado, si la arquitectura es demasiado grande para los datos que poseemos podemos caer en el sobreajuste de la red, esto podría explicar la caída de rendimiento a pesar del aumento de neuronas por capa.

Nº Neuronas	[30]	[50]	[100]	[150]
5	99,50%	99,50%	99,80%	99,60%
7	95,10%	94,90%	96,80%	95,50%
10	87,00%	86,80%	88,70%	88,00%
13	76,70%	78,00%	77,90%	79,40%
17	62,40%	65,30%	67,30%	65,70%
Nº Neuronas	[30 20]	[50 30]	[100 50]	[200 100]
5	99,50%	99,70%	99,50%	98,50%
7	96,20%	95,70%	96,40%	92,70%
10	88,60%	88,40%	87,60%	84,10%
13	77,20%	78,70%	82,10%	80,00%
17	66,10%	63,30%	72,30%	62,70%
Nº Neuronas	[30 50 20]	[50 60 25]	[100 120 50]	[200 240 100]
5	99,50%	99,40%	99,40%	98,60%
7	96,20%	95,80%	96,00%	95,10%
10	87,60%	87,80%	88,30%	82,00%
13	79,80%	86,20%	85,20%	70,20%
17	58,20%	65,20%	73,40%	49,90%

Tabla 6. Eficacia de la ANN para el test en función del nº de neuronas, fallos y capas.



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos



**Test: Plot Confusion**

Output Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		
1	462 5.7%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	11 0.1%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 0.0%	0 0.0%	96.2% 3.7%
2	0 0.0%	459 5.6%	0 0.0%	2 0.0%	0 0.0%	11 0.1%	1 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 0.1%	0 0.0%	95.6% 4.4%
3	0 0.0%	0 0.0%	479 5.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.8% 0.2%
4	0 0.0%	0 0.0%	0 0.0%	448 5.5%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	27 0.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.0%	0 0.0%	2 0.0%	0 0.0%	93.3% 6.7%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	480 5.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
6	40 0.5%	2 0.0%	0 0.0%	12 0.1%	1 0.0%	320 3.9%	3 0.0%	2 0.0%	58 0.7%	31 0.4%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	2 0.0%	5 0.1%	3 0.0%	0 0.0%	66.7% 33.3%
7	1 0.0%	0 0.0%	2 0.0%	52 0.6%	0 0.0%	21 0.3%	184 2.3%	22 0.3%	35 0.4%	9 0.1%	1 0.0%	11 0.1%	0 0.0%	0 0.0%	42 0.5%	56 0.7%	44 0.5%	0 0.0%	38.3% 61.7%
8	0 0.0%	0 0.0%	74 0.9%	3 0.0%	0 0.0%	13 0.2%	0 0.0%	343 4.2%	0 0.0%	0 0.0%	17 0.2%	0 0.0%	5 0.1%	0 0.0%	15 0.2%	4 0.0%	6 0.1%	0 0.0%	71.5% 28.5%
9	0 0.0%	0 0.0%	0 0.0%	10 0.1%	1 0.0%	10 0.1%	9 0.1%	2 0.0%	417 5.1%	4 0.0%	0 0.0%	1 0.0%	0 0.0%	2 0.0%	8 0.1%	11 0.1%	5 0.1%	0 0.0%	86.9% 13.1%
10	0 0.0%	8 0.1%	20 0.2%	8 0.1%	0 0.0%	101 1.2%	8 0.1%	4 0.0%	129 1.6%	146 1.8%	0 0.0%	18 0.2%	1 0.0%	1 0.0%	8 0.1%	4 0.0%	24 0.3%	0 0.0%	30.4% 69.6%
11	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 0.0%	0 0.0%	0 0.0%	434 5.3%	1 0.0%	42 0.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90.4% 9.6%
12	0 0.0%	0 0.0%	2 0.0%	11 0.1%	0 0.0%	1 0.0%	24 0.3%	8 0.1%	3 0.0%	0 0.0%	383 4.7%	1 0.0%	0 0.0%	18 0.2%	12 0.1%	17 0.2%	0 0.0%	0 0.0%	79.8% 20.2%
13	0 0.0%	0 0.0%	1 0.0%	4 0.0%	0 0.0%	22 0.3%	17 0.2%	1 0.0%	0 0.0%	1 0.0%	0 0.0%	412 5.0%	0 0.0%	10 0.1%	3 0.0%	9 0.1%	0 0.0%	0 0.0%	85.8% 14.2%
14	2 0.0%	0 0.0%	6 0.1%	9 0.1%	1 0.0%	3 0.0%	37 0.5%	7 0.1%	32 0.4%	1 0.0%	1 0.0%	0 0.0%	0 0.0%	332 4.1%	25 0.3%	13 0.2%	11 0.1%	0 0.0%	69.2% 30.8%
15	0 0.0%	0 0.0%	2 0.0%	7 0.1%	0 0.0%	0 0.0%	37 0.5%	24 0.3%	0 0.0%	0 0.0%	29 0.4%	0 0.0%	0 0.0%	308 3.8%	60 0.7%	11 0.1%	0 0.0%	0 0.0%	64.2% 35.8%
16	0 0.0%	0 0.0%	2 0.0%	34 0.4%	0 0.0%	1 0.0%	25 0.3%	6 0.1%	0 0.0%	0 0.0%	0 0.0%	3 0.0%	0 0.0%	0 0.0%	22 0.3%	359 4.4%	28 0.3%	0 0.0%	74.8% 25.2%
17	0 0.0%	0 0.0%	8 0.1%	80 1.0%	0 0.0%	3 0.0%	99 1.2%	30 0.4%	42 0.5%	0 0.0%	0 0.0%	9 0.1%	27 0.3%	0 0.0%	105 1.3%	56 0.7%	21 0.3%	0 0.0%	4.4% 95.6%
	91.5% 8.5%	97.9% 2.1%	80.4% 19.6%	65.8% 34.2%	99.4% 0.6%	66.3% 33.7%	39.7% 60.3%	73.0% 27.0%	56.0% 44.0%	76.4% 23.6%	95.6% 4.4%	84.2% 15.8%	83.9% 16.1%	99.1% 0.9%	54.0% 46.0%	61.2% 38.8%	11.5% 88.5%	73.4% 26.6%	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		

**Target Class**

Figura 4.15 Matriz de confusión para el test. (17 fallos). ANN multicapa

Con lo experimentado en ANN de propagación hacia adelante, no hemos conseguido unos resultados aceptables para el problema propuesto de clasificación de los 17 fallos de estudio. El mejor dato de eficacia obtenido, 73,40%, corresponde a una ANN con tres capas con 100, 120 y 50 neuronas respectivamente en cada una de sus capas, y no alcanza un 75% de acierto.

Por otro lado, logramos buenos datos de clasificación en algunos fallos puntuales como IDV (1), IDV (2), IDV (4), IDV (7) con porcentajes mayores al 95%, sin embargo, fallos como IDV (10), IDV (13), no superan el 50% (Tabla 7). Cabe destacar la incapacidad de la red para la clasificación del último fallo IDV (21) (4,4%).



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos



Fallo	Eficacia individual (Test) (%)
IDV (1)	96,2%
IDV (2)	95,6%
IDV (4)	99,8%
IDV (5)	93,3%
IDV (7)	100,0%
IDV (8)	66,7%
IDV (10)	38,3%
IDV (11)	71,5%
IDV (12)	86,9%
IDV (13)	30,4%
IDV (14)	90,4%
IDV (16)	79,8%
IDV (17)	85,8%
IDV (18)	69,2%
IDV (19)	64,2%
IDV (20)	74,8%
IDV (21)	4,4%

Tabla 7. Eficacia por fallo para una ANN con tres capas ocultas.

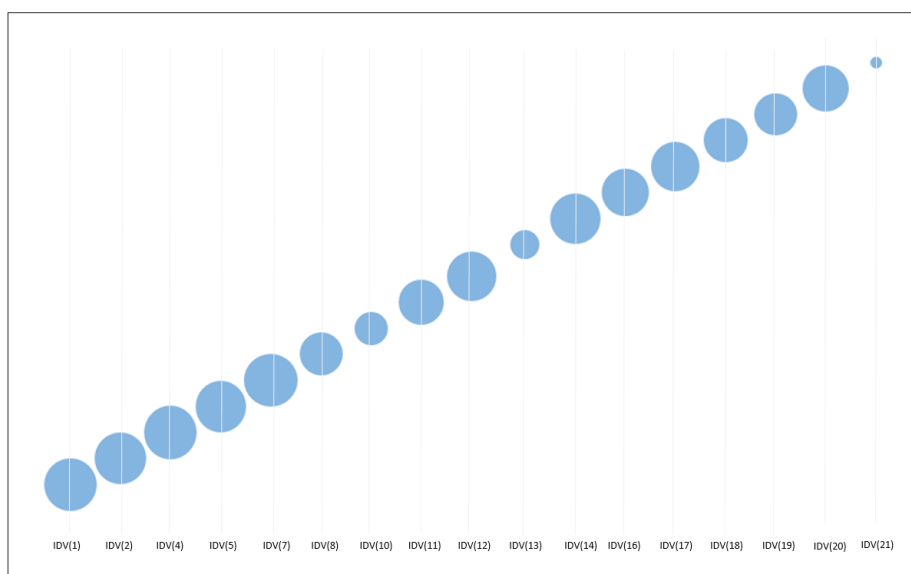


Figura 4.16. Eficacia por fallo para una ANN con tres capas ocultas



#### 4.3.1.3. APLICACIÓN DEL PCA A LAS ANN DE PROPAGACIÓN HACIA ADELANTE.

Hasta ahora se han aplicado los métodos de aprendizaje que nos proporcionan las redes neuronales de propagación hacia adelante con los datos de partida, que contienen una simulación para las 52 variables. Se ha pretendido introducir previamente un Análisis de Componentes Principales (PCA) a los datos en función de la variabilidad aceptada, para reducir las matrices de datos de entrada con las que se pretende entrenar las redes, y medir si la reducción de efectividad es significativa.

Como se ha hecho en apartados anteriores se han tomado datos de eficacia para las cinco situaciones descritas con 5, 7, 10, 13 y 17 fallos.

La red que ha sido utilizada para la toma de estos registros ha sido la red que ofrecía mejores resultados en el apartado anterior (Figura 4.17).

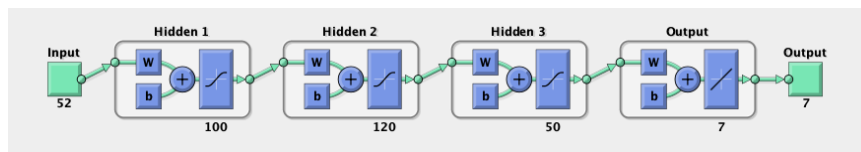


Figura 4.17. Arquitectura ANN con 100, 120 y 50 neuronas por capa respectivamente.

Se ha adoptado una variabilidad aceptada del 90%, para la aplicación del PCA, lo que reduce el espacio muestra inicial de 52 variables a 18, 19 o 20 componentes principales. Por lo tanto, la matriz de entrada a la red se reduce de  $\mathfrak{R}^{52}$  a  $\mathfrak{R}^a$ . Donde  $a$  obtiene valores comprendidos entre 18 y 20 en función del número de fallos introducidos en la matriz de entrada  $X$ .

Los resultados se ofrecen en la siguiente tabla (Tabla 8) donde se muestran los resultados obtenidos con la aplicación previa del PCA con una variabilidad del 90%, comparándolos con los resultados del apartado anterior.



Nº de Fallos	Con PCA			Sin PCA		
	Train	Validation	Test	Train	Validation	Test
5	99,70%	99,90%	98,70%	99,90%	100,00%	99,40%
7	90,90%	81,50%	85,40%	99,20%	97,80%	96,00%
10	86,70%	77,90%	74,40%	96,60%	89,60%	88,30%
13	80,50%	73,10%	69,40%	96,70%	85,90%	85,20%
17	63,70%	58,60%	50,90%	84,50%	81,10%	73,40%

Tabla 8. Acierto en la clasificación para el entrenamiento, validación y test con PCA vs. Sin PCA

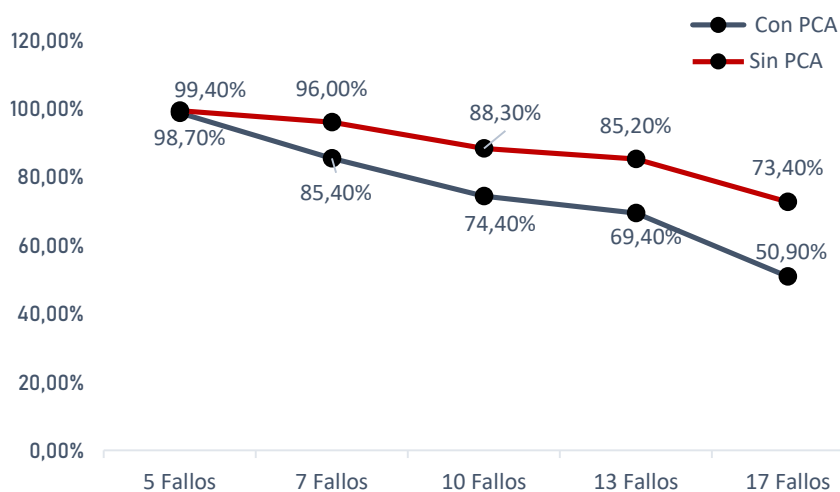


Figura 4.18. Comparativa evolutiva de la eficacia en la respuesta de la ANN con salida PCA vs. Sin PCA para el test

Parece evidente la pérdida de eficacia a partir de los 10 fallos. La simulación se ha repetido para una variabilidad aceptada mayor, del 99% y 99,9% obteniendo resultados similares. Concluyendo que la aplicación previa del PCA supone una pérdida significativa de acierto en la clasificación.



#### 4.3.1.4. METADATOS APLICADOS A LA RED DE PROPAGACIÓN HACIA ADELANTE.

A la vista de los resultados obtenidos, se han experimentado con los metadatos descargados de las matrices ‘*faultytraining*’ y ‘*faultytesting*’ para una red neuronal artificial de propagación hacia adelante. Las matrices de metadatos contienen 500 simulaciones (*SimulationRun*) con 500 observaciones (*sample*) para cada uno de los 20 fallos. Por lo tanto, contamos con matrices de 250.000 observaciones para cada una de las 52 variables para cada fallo.

Al igual que hemos hecho en todas las experimentaciones realizadas a lo largo de este trabajo se han obviado los fallos IDV(3), IDV(9) e IDV(15). Por otro lado, por falta de datos se ha obviado el IDV(21). Con intención de agilizar el proceso, no se han utilizado las 250.000 observaciones de cada fallo. Para el entrenamiento de la ANN se han utilizado el 20% de los datos de la matriz ‘*faultytraining*’, lo que supone que para cada uno de los 17 fallos se han utilizado matrices de  $50.000 \times 52$ .

La salida deseada será de la ya denominada tipo II realizada en el apartado 4.3.1.1 del presente trabajo. Para el proceso de validación se ha utilizado también observaciones obtenidas de la matriz ‘*faultytraining*’. En este caso se han utilizado matrices con una dimensión de  $10.000 \times 52$  por fallo. La red elegida contiene la misma arquitectura que nos arrojaba mejores resultados con los datos iniciales, con tres capas ocultas y 100, 120 y 50 neuronas respectivamente.

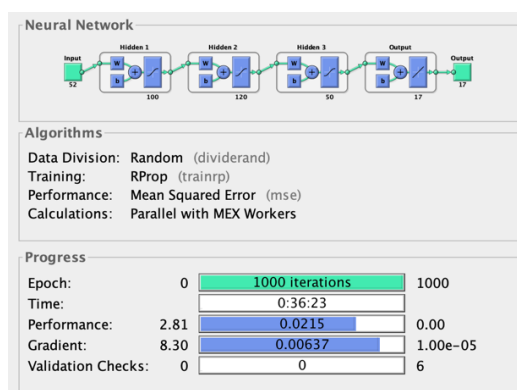


Figura 4.19. Captura del proceso de entrenamiento de la ANN con metadatos



La fase de entrenamiento con una matriz de entrada en la red  $X_{Train}$  con dimensiones  $(850.000 \times 17)$  se ha prolongado durante 36min 23s (Figura 4.19), lo que supone un aumento significativo en el tiempo computacional con respecto a los entrenamientos hasta el momento.

Finalmente, para el proceso de test, obtenemos los datos de la matriz descargada 'faultytesting' que contiene 500 simulaciones con 960 observaciones lo que supone matrices de observaciones de dimensiones  $48.000 \times 52$  para cada fallo. Al igual que en los procesos anteriores no se han utilizado la totalidad de los datos, en este caso se han introducido en la red matrices con el 10% de los datos, con 48.000 observaciones por fallo. Nótese que estos datos suponen un aumento de las dimensiones del 9.900% con respecto a los datos originales testados.

En la Figura 4.20 se muestra la salida de la red para el proceso de test para los fallos IDV(1), IDV(2), IDV(4), IDV(5), IDV(6) e IDV(7).

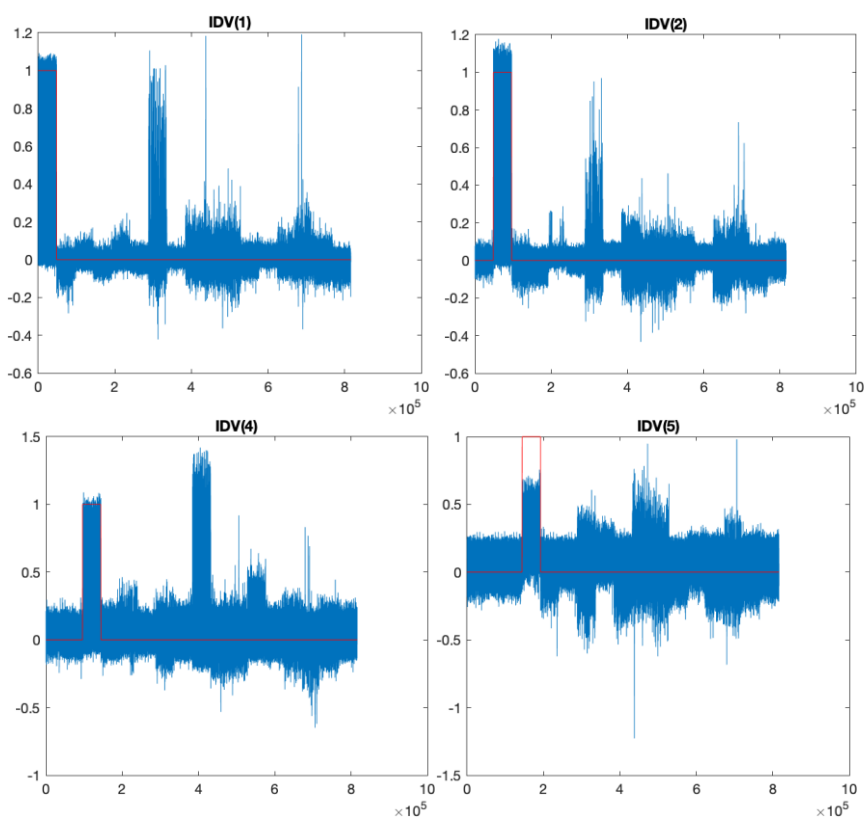


Figura 4.20-1. Salida de la ANN con metadatos para IDV(1), IDV(2), IDV(4), IDV(5) para el proceso de test.





## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos

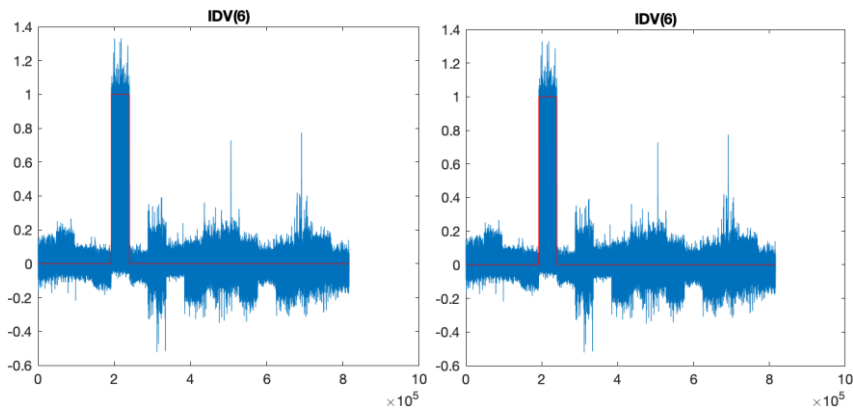


Figura 4.20-2. Salida de la ANN con metadatos para IDV(6) e IDV(7) para el proceso de test.

Los resultados obtenidos para las fases de entrenamiento, validación y test se representan en las siguientes matrices de confusión (Figura 4.21 a 4.23).

		TRAIN: Plot Confusion																																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17																			
Output Class	1	46391 5.5%	0	0	0	0	0	311	0	0	2	1	0	0	0	0	0	0	90.3%	0.7%																	
	2	0	46038 5.4%	0	0	0	0	160	0	0	0	0	0	0	0	0	0	0	99.7%	0.3%																	
	3	21	25	47657 5.6%	33	18	18	75	730	9184	86	108	875	532	422	90	543	161	78.7%	21.3%																	
	4	186	254	182	44485 5.2%	182	182	1105	10603	1011	2251	1727	186	5550	1260	1412	3253	1801	58.8%	41.2%																	
	5	0	0	0	0	48000 5.6%	0	1	0	0	2	5	0	0	0	0	0	0	100.0%	0.0%																	
	6	0	0	0	0	0	47999 5.6%	0	60	0	0	2	0	2	0	0	0	0	99.9%	0.1%																	
	7	974	930	0	48	0	0	42666 5.0%	679	23	322	597	20	331	17	53	95	88	91.1%	8.9%																	
	8	624	1095	601	609	601	601	1406	13904	1875	1067	2250	670	10044	2241	2413	5291	3490	28.5%	71.5%																	
	9	147	103	366	196	70	70	259	1628	30995 3.6%	323	353	2779	1282	2476	295	2787	728	69.1%	30.9%																	
	10	36	11	8	2593 0.3%	8	9	956	809	50	41927 4.9%	1016	21	603	69	2870	122	414	81.4%	18.6%																	
	11	2	2	2	14	2	2	303	398	5	254	40209 4.7%	4	273	4	33	3	17	96.8%	3.2%																	
	12	0	0	38	0	0	0	2	0	2849 0.3%	4	5	39566 4.7%	0	9	1	8	0	93.1%	6.9%																	
	13	917	1133	823	836	823	823	1510	15063	2257	1529	2464	876	23481 2.8%	2711	2718	7002	4588	33.9%	66.2%																	
	14	0	0	0	2	0	0	128	9	413	12	193	4397	28	39742 4.7%	2	42	0	88.4%	11.6%																	
	15	0	0	0	0	0	0	16	5	0	1424	54	0	14	0	39088 4.6%	6	2	96.3%	3.7%																	
	16	232	288	233	1083	206	206	489	4107	996	555	696	456	3975	767	737	28001 3.3%	1558	62.8%	37.2%																	
	17	470	121	90	101	90	90	613	2005	342	242	320	150	3885	282	290	2847	37153 4.4%	75.7%	24.3%																	
		92.8%	92.1%	95.3%	89.0%	96.0%	96.0%	85.3%	27.8%	62.0%	83.9%	80.4%	79.1%	47.0%	79.5%	78.2%	56.0%	74.3%	77.3%	7.2%	7.9%	4.7%	11.0%	4.0%	4.0%	14.7%	72.2%	38.0%	16.1%	19.6%	20.9%	53.0%	20.5%	21.8%	44.0%	25.7%	22.7%
		Target Class																																			

Figura 4.21. Matriz de confusión para el entrenamiento de ANN con metadatos



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos



**VALIDATION: Plot Confusion**

1	9286	0	2	41	0	0	196	140	25	2	0	0	180	0	0	42	86	92.9%
	5.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.1%	7.1%
2	0	9194	5	53	0	0	206	226	14	0	0	0	217	0	0	55	30	91.9%
	0.0%	5.4%	0.0%	0.0%	0.0%	0.0%	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	8.1%
3	0	0	9536	40	0	0	0	134	60	0	0	10	158	0	0	44	18	95.4%
	0.0%	0.0%	5.6%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	4.6%
4	0	0	8	8925	0	0	9	137	28	596	9	0	164	0	0	195	19	89.2%
	0.0%	0.0%	0.0%	5.2%	0.0%	0.0%	0.0%	0.1%	0.0%	0.3%	0.0%	0.0%	0.1%	0.0%	0.0%	0.1%	0.0%	10.8%
5	0	0	2	40	0	0	0	134	10	0	0	0	158	0	0	38	18	96.0%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	4.0%
6	0	0	2	40	0	0	0	134	10	0	0	0	158	0	0	38	18	96.0%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	4.0%
7	49	51	19	194	0	0	8594	288	41	177	64	2	286	33	2	98	102	85.9%
	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	5.1%	0.2%	0.0%	0.1%	0.0%	0.0%	0.2%	0.0%	0.0%	0.1%	0.1%	14.1%
8	0	0	134	2188	0	15	139	2898	314	151	61	0	2877	4	0	811	410	29.0%
	0.0%	0.0%	0.1%	1.3%	0.0%	0.1%	1.7%	0.2%	0.1%	0.0%	0.0%	0.0%	1.7%	0.0%	0.0%	0.5%	0.2%	71.0%
9	0	0	1768	236	0	0	6	448	6164	8	1	536	465	93	0	201	74	61.8%
	0.0%	0.0%	1.0%	0.1%	0.0%	0.0%	0.0%	0.3%	3.6%	0.0%	0.0%	0.3%	0.1%	0.0%	0.1%	0.0%	0.0%	38.4%
10	0	0	23	391	0	0	54	232	64	8471	28	0	293	0	255	125	64	84.7%
	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.1%	0.0%	5.0%	0.0%	0.0%	0.2%	0.0%	0.1%	0.1%	0.0%	15.3%
11	0	0	20	293	2	0	89	409	172	103	8111	0	475	43	61	141	81	81.1%
	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.1%	0.2%	0.1%	0.1%	4.8%	0.0%	0.3%	0.0%	0.1%	0.0%	0.0%	18.9%
12	0	0	156	41	0	0	4	143	553	0	0	7949	164	884	0	83	23	79.5%
	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.1%	0.3%	0.0%	0.0%	4.7%	0.1%	0.5%	0.0%	0.0%	0.0%	20.5%
13	0	0	94	1091	0	0	54	2063	274	119	65	0	4621	7	2	798	792	46.2%
	0.0%	0.0%	0.1%	0.8%	0.0%	0.0%	0.0%	1.2%	0.2%	0.1%	0.0%	0.0%	2.7%	0.0%	0.0%	0.5%	0.5%	53.8%
14	0	0	98	250	0	0	7	474	500	13	0	2	537	7886	0	168	65	78.9%
	0.0%	0.0%	0.1%	0.1%	0.0%	0.0%	0.0%	0.3%	0.3%	0.0%	0.0%	0.0%	0.3%	4.6%	0.0%	0.1%	0.0%	21.1%
15	0	0	23	276	0	0	15	498	35	627	21	0	500	1	7770	172	62	77.7%
	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.3%	0.0%	0.4%	0.0%	0.0%	0.3%	0.0%	4.6%	0.1%	0.0%	22.3%
16	0	0	83	640	0	0	23	1068	575	29	3	0	1439	13	0	5577	552	55.8%
	0.0%	0.0%	0.0%	0.4%	0.0%	0.0%	0.0%	0.6%	0.3%	0.0%	0.0%	0.0%	0.8%	0.0%	0.0%	3.3%	0.3%	44.2%
17	0	0	39	346	0	0	14	678	148	61	0	0	883	0	0	302	7529	75.3%
	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.4%	0.1%	0.0%	0.0%	0.0%	0.5%	0.0%	0.0%	0.2%	4.4%	24.7%
	99.5%	99.4%	79.4%	59.2%	100.0%	99.8%	91.3%	28.6%	68.0%	82.5%	97.0%	93.5%	34.0%	88.0%	96.0%	62.7%	75.7%	77.8%
	0.5%	0.6%	20.6%	40.8%	0.0%	0.2%	8.7%	71.4%	31.4%	17.5%	3.0%	6.5%	66.0%	12.0%	4.0%	37.3%	24.3%	22.8%

Figura 4.22. Matriz de confusión para la validación de ANN con metadatos

**TEST: Plot Confusion**

1	39203	0	77	1030	0	1	513	2634	296	46	2	0	2857	0	2	790	549	81.7%	
	4.8%	0.0%	0.0%	0.1%	0.0%	0.0%	0.1%	0.3%	0.0%	0.0%	0.0%	0.0%	0.4%	0.0%	0.0%	0.1%	0.1%	18.3%	
2	0	39027	82	1063	0	1	482	2845	281	36	2	0	2985	0	2	827	367	81.3%	
	0.0%	4.8%	0.0%	0.1%	0.0%	0.0%	0.1%	0.3%	0.0%	0.0%	0.0%	0.0%	0.4%	0.0%	0.0%	0.1%	0.0%	18.7%	
3	0	0	39822	1028	0	1	39	2616	477	34	2	17	2813	0	2	803	346	83.0%	
	0.0%	0.0%	4.9%	0.1%	0.0%	0.0%	0.0%	0.3%	0.1%	0.0%	0.0%	0.0%	0.3%	0.0%	0.0%	0.1%	0.0%	17.0%	
4	0	0	92	38743	0	1	66	2628	387	1349	9	0	2822	1	2	1544	356	80.7%	
	0.0%	0.0%	0.0%	4.7%	0.0%	0.0%	0.0%	0.3%	0.0%	0.2%	0.0%	0.0%	0.3%	0.0%	0.0%	0.2%	0.0%	19.3%	
5	0	0	76	1028	40000	1	39	2616	283	34	2	0	2813	0	2	780	346	83.3%	
	0.0%	0.0%	0.0%	0.1%	4.9%	0.0%	0.0%	0.3%	0.0%	0.0%	0.0%	0.0%	0.3%	0.0%	0.0%	0.1%	0.0%	16.7%	
6	0	0	76	1028	0	0	40001	39	2616	263	34	2	0	2813	0	2	780	346	83.3%
	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	4.9%	0.0%	0.3%	0.0%	0.0%	0.0%	0.3%	0.0%	0.0%	0.1%	0.0%	16.7%	
7	310	108	118	1869	1	2	35701	3091	384	1204	245	5	3232	78	12	967	673	74.4%	
	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	4.4%	0.4%	0.0%	0.1%	0.0%	0.0%	0.4%	0.0%	0.0%	0.1%	0.1%	25.6%	
8	0	0	645	10191	0	90	592	13718	1551	638	277	0	14413	15	3	3955	1912	28.6%	
	0.0%	0.0%	0.1%	1.2%	0.0%	0.0%	1.7%	0.2%	0.1%	0.0%	0.0%	0.0%	1.8%	0.0%	0.0%	0.5%	0.2%	71.4%	
9	0	0	7439	1698	0	1	55	3710	26516	67	3	2325	3927	358	2	1355	544	55.2%	
	0.0%	0.0%	0.9%	0.2%	0.0%	0.0%	0.0%	0.5%	3.2%	0.0%	0.0%	0.3%	0.5%	0.0%	0.0%	0.2%	0.1%	44.8%	
10	6	0	109	2136	0	1	307	2811	540	30172	221	279	3202	15	6781	984	436	62.9%	
	0.0%	0.0%	0.0%	0.3%	0.0%	0.0%	0.0%	0.3%	0.1%	3.7%	0.0%	0.4%	0.0%	0.0%	0.8%	0.1%	0.1%	37.1%	
11	0	0	274	1825	0	5	623	3395	529	950	34552	3	3733	168	432	1050	461	72.0%	
	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.1%	0.4%	0.1%	0.1%	4.2%	0.0%	0.5%	0.0%	0.1%	0.1%	0.1%	28.0%	
12	0	0	768	1033	0	1	55	2657	2435	40	4	33099	2852	3655	2	1014	385	69.0%	
	0.0%	0.0%	0.1%	0.1%	0.0%	0.0%	0.0%	0.3%	0.3%	0.0%	0.0%	4.1%	0.3%	0.4%	0.0%	0.1%	0.0%	31.0%	
13	0	0	474	5469	0	3	320	10326	1306	496	258	0	21871	32	11	3888	3526	45.6%	
	0.0%	0.0%	0.1%	0.7%	0.0%	0.0%	1.3%	0.2%	0.1%	0.0%	0.0%	0.0%	2.7%	0.0%	0.0%	0.5%	0.4%	54.4%	
14	0	0	405	1746	0	1	49	3543	2338	102	3	9	4042	34137	2	1145	478	71.1%	
	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.4%	0.3%	0.0%	0.0%	0.0%	0.5%	4.2%	0.0%	0.1%	0.1%	28.9%	
15	79	0	109	1552	0	1	73	3408	370	2108	58	0	3709	1	35058	1035	439	73.0%	
	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.4%	0.0%	0.3%	0.0%	0.0%	0.5%	0.0%	4.3%	0.1%	0.1%	27.0%	
16	0	0	453	3651	0	1	122	6552	2622	132	6	3	7899	47	4	23949	2559	49.9%	
	0.0%	0.0%	0.1%	0.4%	0.0%	0.0%	0.0%	0.8%	0.3%	0.0%	0.0%	0.0%	1.0%	0.0%	0.0%	2.9%	0.3%	50.1%	
17	0	0	195	2235	0	1	98	4510	838	375	9	0	5723	2	6	1728	32280	67.2%	
	0.0%	0.0%	0.0%	0.3%	0.0%	0.0%	0.0%	0.6%	0.1%	0.0%	0.0%	0.0%	0.7%	0.0%	0.0%	0.2%	4.0%	32.8%	
	99.0%	99.7%	77.8%	50.1%	100.0%	99.7%	91.1%	18.6%	64.1%	79.8%	96.9%	92.6%	23.8%	88.6%	82.8%	51.4%	70.2%	68.4%	
	1.0%	0.3%	22.2%	49.9%	0.0%	0.3%	8.9%	81.4%	35.9%	20.2%	3.1%	7.4%	76.2%	11.4%	17.2%	48.6%	29.8%	31.6%	

Figura 4.23. Matriz de confusión para el test de ANN con metadatos



Los porcentajes de eficacia obtenidos para la red con los metadatos con dimensiones reducidas en los procesos de entrenamiento, validación y test se muestran en la Tabla 9. Con los resultados obtenidos se puede afirmar que no hay un aumento significativo en la eficacia de clasificación, y por otro lado, el coste computacional del entrenamiento es mayor.

Nº de Fallos	Train	Validation	Test
17	77,30%	77,50%	68,40%

Tabla 9. Resultados de la red ANN de propagación hacia adelante con metadatos

#### 4.3.2. REDES LONG SHORT-TERM MEMORY (LSTM).

La experimentación con redes de memoria a corto plazo (LSTM) se puede subdividir en dos grandes apartados en función de la arquitectura diseñada de la red. A continuación, se explica dos métodos de trabajo con datos de series de tiempo y secuencia para tareas de regresión y clasificación usando redes LSTM.

Los componentes de la arquitectura de una red LSTM que comparte con las ya vistas ANN es su capa de entrada (*Sequence Input*) que introduce los datos a la red. Además, tanto en los métodos de clasificación como de regresión contamos con un cuerpo de capas LSTM, que pueden ser alternadas con capas de abandono o *dropout*, y una capa completamente conectada, FC, Fully Connected, en la cual todas las neuronas están conectadas a las neuronas de la siguiente capa.

Como se pueden ver en las Figura 4.24 y 4.25, donde se representan los diagramas de las arquitecturas para la regresión y clasificación con una red LSTM, la principal diferencia entre métodos surge en la estructura de la salida de la red.

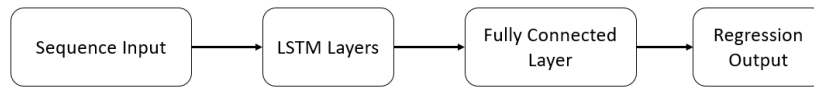


Figura 4.24. Diagrama de la Arquitectura de una red LSTM para la regresión



Figura 4.25. Diagrama de la Arquitectura de una red LSTM para la clasificación

#### 4.3.2.1. Metodología con redes LSTM aplicando Regresión.

Las redes LSTM para la regresión cuentan con una capa de salida, denominada Regression Output, que requiere de una capa FC (completamente conectada) previa. La capa de regresión calcula la pérdida de error medio cuadrático medio,  $MSE$ , para problemas de regresión. Donde el error cuadrático viene dado por:

$$MSE = \sum_{i=0}^R \frac{(t_i - y_i)^2}{R} \quad (1)$$

Donde  $R$  es el número de respuestas, fallos en nuestro caso,  $t_i$  es la salida deseada e  $y_i$  es la predicción de salida para la respuesta  $i$ .

La estructura del estudio con la red LSTM es análoga al estudio realizado para ANN, primero se ha experimentado con dos tipos de salidas, al igual que en las ANN se ha generado una salida tipo I y tipo II.

##### Tipo I.

La salida tipo I, vuelve a ser una salida única y escalonada, compuesta por los valores nominales de los  $N$  fallos introducidos en la red de la forma que el vector salida corresponde con un vector de dimensiones  $1 \times N$ .

$$Y = 1, 2, 3, 4, \dots, N \quad (2)$$

Las matrices de entrada a la red son iguales que las que introducíamos en los estudios anteriores, se introducen unos datos de entrenamiento a la red con las



observaciones 161 a la 810 para los 17 fallos. Con estos datos se entrena la red, y se valida con los datos de las observaciones 811 a 960 del mismo fichero de datos. Por último, se realiza el test con las 480 observaciones de test.

Debemos definir la arquitectura de la red, la red LSTM en este caso consta de dos capas ocultas LSTM de 150 y 100 neuronas respectivamente, consultando estudios anteriores al respecto [24], se ha considerado oportuno definir una tasa de abandono o *dropout* posterior a cada una de las capas LSTM del 50%, y una capa FC con el número total de respuestas, en este caso única. Finalmente, se utiliza una capa Regression.

Otros parámetros utilizados como el MaxEpochs, MiniBatchSize, o InitialLearningRate son especificados a continuación (Tabla 10).

MaxEpochs	MiniBatchSize	InitialLearningRate	GradientThreshold
150	20	0.001	1

Tabla 10. Parámetros utilizados para la red LSTM

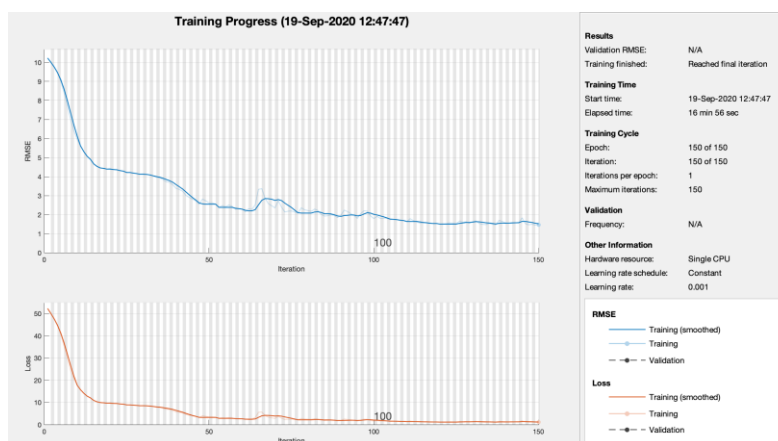


Figura 4.26. Proceso de entrenamiento para la red LSTM con salida tipo I

Los resultados obtenidos para este tipo de salida se pueden observar en la Figura 4.27 y 4.28. Se aprecia un acierto en la respuesta para el proceso de entrenamiento (Figura 4.27 derecha), pero no así para la validación (Figura 4.27 izquierda), y el test (Figura 4.28). La respuesta para estos dos últimos procedimientos parece no conseguir replicar la salida escalonada en absoluto,



por lo que los resultados obtenidos mediante este método no resultan satisfactorios.

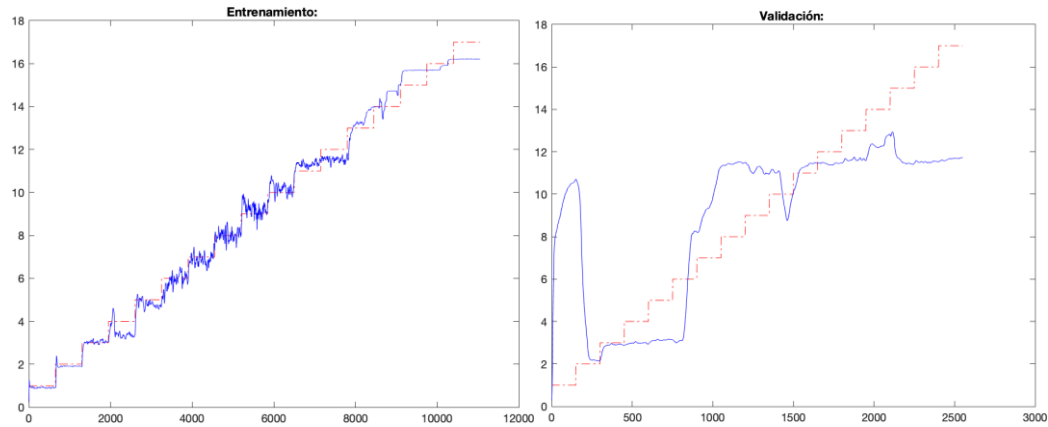


Figura 4.27. Respuesta de la red LSTM con salida tipo I para el entrenamiento (izquierda) y validación (derecha)

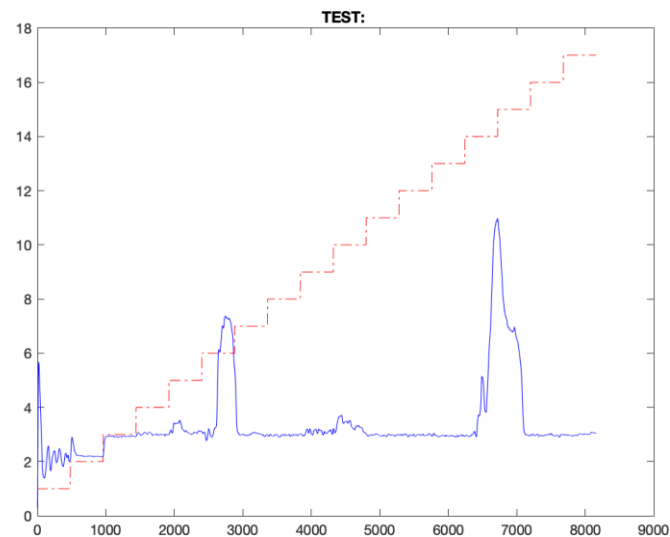


Figura 4.28. Respuesta de la red LSTM con salida tipo I para el test.

## Tipo II.

La salida tipo II corresponde con una salida múltiple binaria, al igual que en el apartado 4.3.1.1. Los datos para el entrenamiento son iguales que en el apartado anterior, se experimenta con los 17 fallos.

La estructura de la red es similar a la utilizada anteriormente, consta de una capa de entrada donde se introduce la matriz de datos de los 17 fallos, tres capas LSTM, una capa FC y una capa de Regression. Únicamente ha sido



incrementado el número de unidades en las capas ocultas, con 200, 130 y 100 unidades en las respectivas capas LSTM.

Por otro lado, se han modificado ligeramente los parámetros de la red, se ha incrementado el número de épocas, y se ha reducido el parámetro MiniBatchSize, los parámetros utilizados se detallan en la Tabla 11.

MaxEpochs	MiniBatchSize	InitialLearningRate	GradientThreshold
250	10	0.001	1

Tabla 11. Parámetros utilizados para la red LSTM

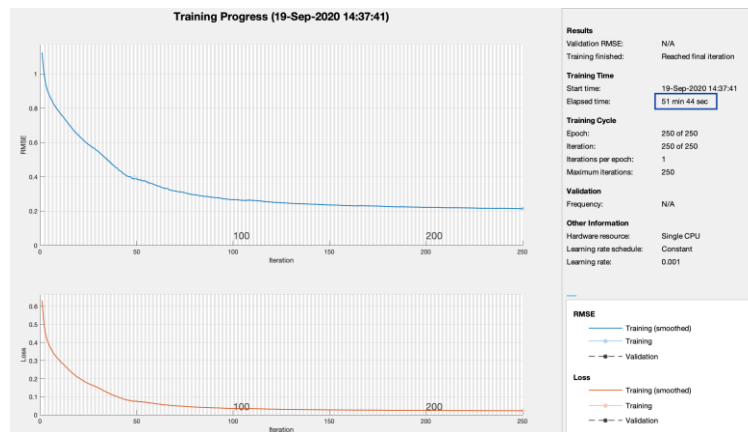


Figura 4.29. Proceso de entrenamiento para la red LSTM con salida tipo II

La respuesta de la red para el proceso de entrenamiento y test para los 4 primeros fallos se puede observar en las Figuras 4.30 y 4.31. Las matrices de confusión para respectivos procesos se observan en las Figura 4.32 y 4.33 con una tasa global de acierto en la clasificación del 99,9% y 6,9% respectivamente.

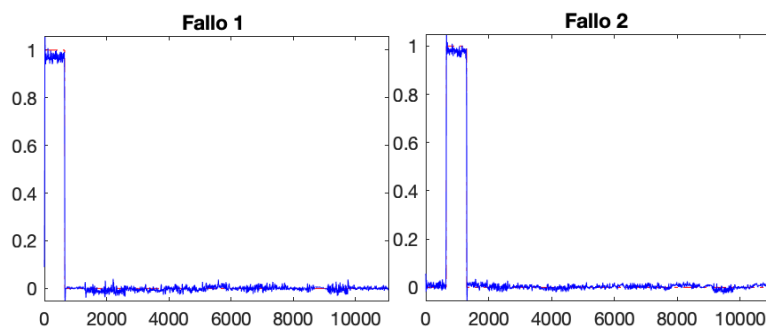


Figura 4.30-1. Respuesta de la red LSTM para los fallos IDV(1), IDV(2) con salida tipo II para el entrenamiento.

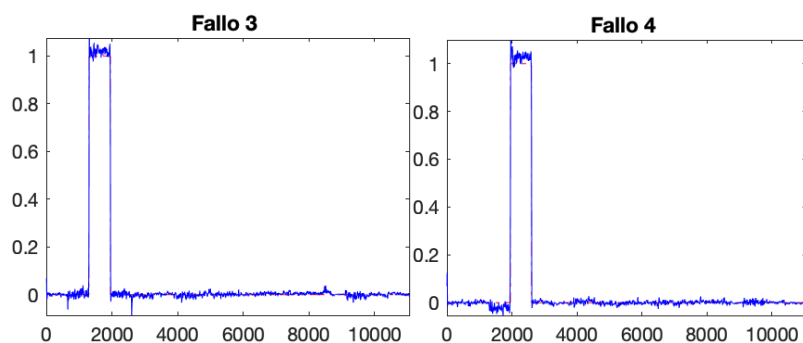


Figura 4.30-2. Respuesta de la red LSTM para los fallos IDV(4), IDV(6) con salida tipo II para el entrenamiento.

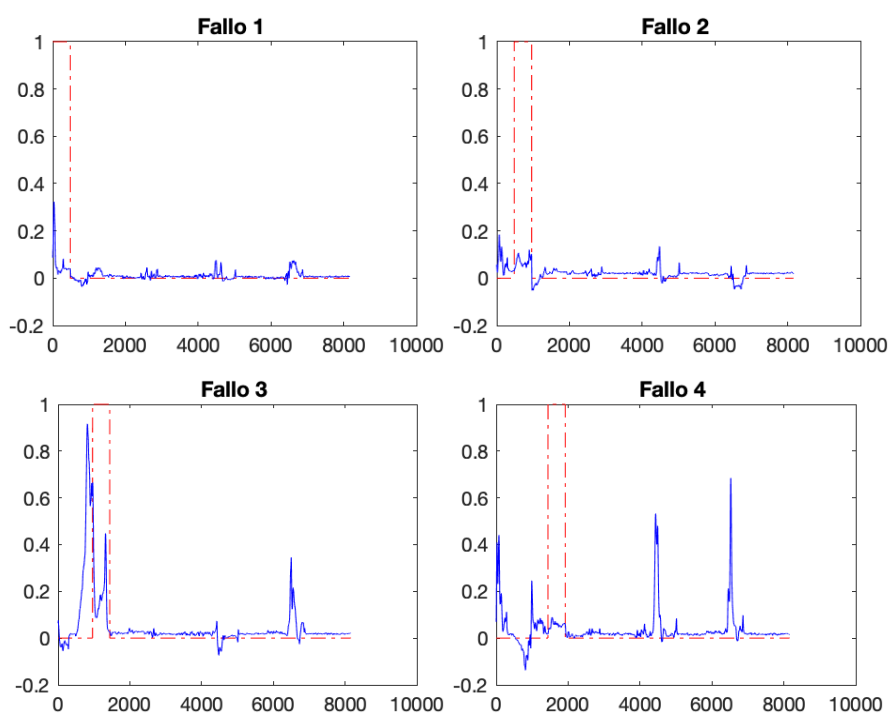


Figura 4.31. Respuesta de la red LSTM para los fallos IDV(1), IDV(2), IDV(4), IDV(6) con salida tipo II para el test.

Los resultados para la red LSTM con capa de regresión obtienen resultados muy satisfactorios en la fase de entrenamiento de la red, pero su rendimiento es deficiente en las fases tanto de validación como de test, siendo estas redes incapaces de clasificar adecuadamente 17 fallos simultáneos.





# Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos

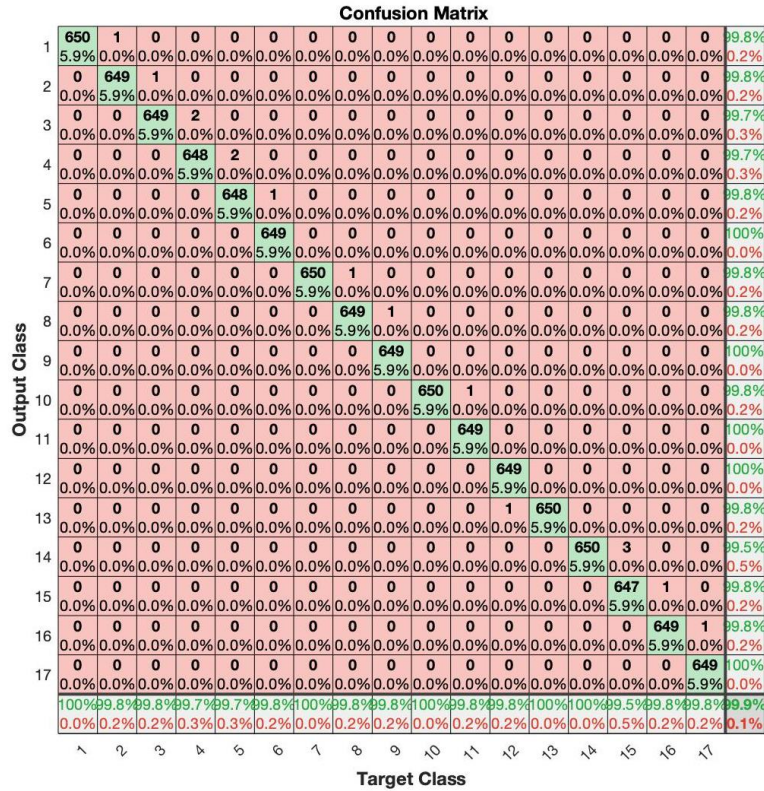


Figura 4.32. Matriz de confusión para el entrenamiento de la red LSTM con salida tipo II

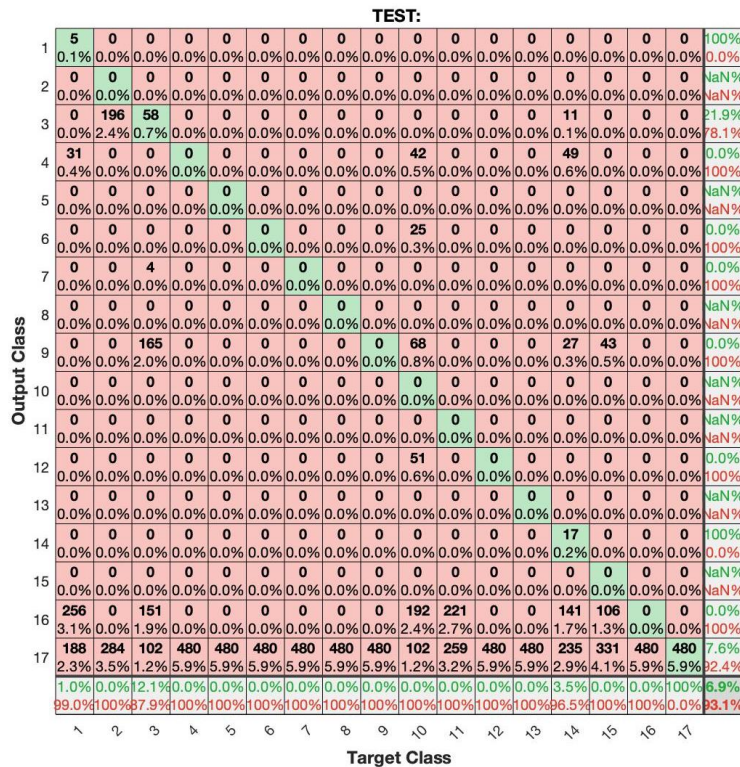


Figura 4.33. Matriz de confusión para el test de la red LSTM con salida tipo II



#### 4.3.2.2. Metodología con redes LSTM aplicando Clasificación.

Las redes LSTM para la clasificación cuentan con una capa de entrada (*Sequence Input*) y un cuerpo de capas LSTM al igual que las redes de regresión. También precisan de una capa *Fully Connected*, FC, previa a la salida requerida para la clasificación, que consta de una capa denominada *Softmax*, y una capa de clasificación.

La capa *softmax* se aplica entre la capa FC y la capa de clasificación final de la red. La función de activación de la unidad de salida es la función *softmax*, que viene dada por la siguiente ecuación:

$$y_r(x) = \frac{\exp(a_r(x))}{\sum_{j=1}^a \exp(a_j(x))} \quad (3)$$

Donde:  $0 \leq y_r \leq 1$  y  $\sum_{j=1}^a y_j = 1$

La capa de clasificación calcula la pérdida de entropía cruzada para los problemas de clasificación de clases múltiples. La capa de clasificación debe alojarse tras la capa *Softmax*.

La función de pérdida de entropía corresponde con:

$$loss = - \sum_{i=1}^N \sum_{j=1}^K t_{ij} \ln y_{ij} \quad (4)$$

Donde  $N$  es el número de observaciones,  $K$  el número de muestras,  $t_{ij}$  es el indicador de que la  $i$ -ésima muestra pertenece a la  $j$ -ésima clase, e  $y_{ij}$  es la salida para la muestra  $i$  para la clase  $j$ . [26]

La entrada a la red para los 17 fallos necesita ser introducida en la red como una matriz de celdas, de forma que la matriz de entrada  $X$  tendrá dimensiones de  $17 \times 1$ , donde cada una de las celdas contiene la matriz de datos pertenecientes a cada fallo. De este modo, la matriz  $X_E$  para el entrenamiento esta compuesta por 17 celdas, donde para cada celda encontramos una matriz con dimensiones  $52 \times 650$ . De manera análoga las matrices para la validación



$X_V$ , y para el test  $X_T$ , contienen 17 celdas con matrices de  $52 \times 150$  y  $52 \times 480$ , respectivamente.

La salida generada para este tipo de red tiene la particularidad de que debe ser definida como una variable categórica. De este modo, la salida  $Y_i$  es construida como una serie de clases donde cada celda pertenece al número de fallo. Por lo tanto:

$$Y_i = \{1, 2, 4, 5, 7, 8, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20, 21\} \quad (5)$$

Al igual que en las aplicaciones anteriores, se entrena la red con los ficheros de datos de entrenamiento y las observaciones 161 hasta 810, ya que las primeras 160 corresponden a observaciones de comportamiento normal previas al fallo.

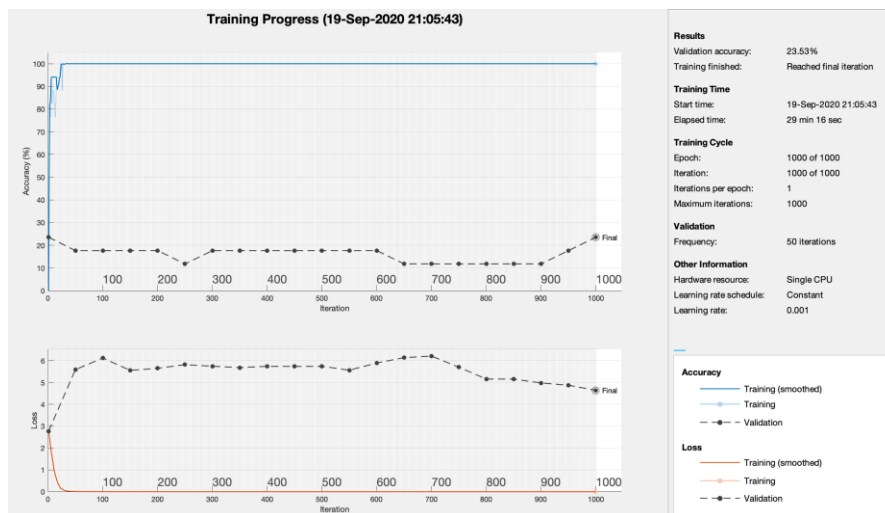


Figura 4.34. Proceso de entrenamiento para la red LSTM de clasificación

Los resultados para el entrenamiento se muestran en la matriz de confusión de la Figura 4.35. En este caso la matriz de confusión muestra el número de simulaciones, en este caso una, que están bien clasificadas, si se encuentran en la diagonal de la matriz, o si por el contrario ha sido incorrectamente clasificada. Para el entrenamiento observamos un 100% de acierto en la clasificación ya que todas las simulaciones se encuentran en la diagonal.

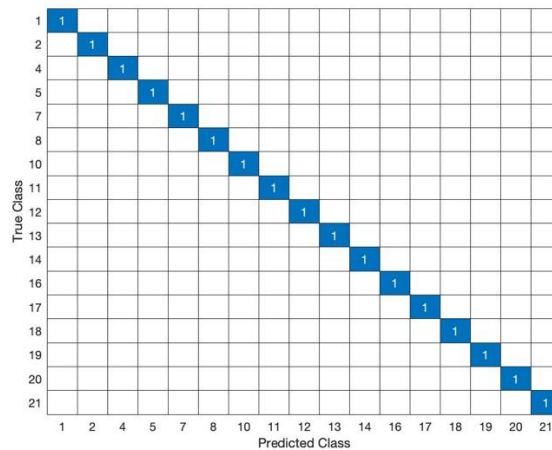


Figura 4.35. Matriz de confusión para el entrenamiento de la red LSTM de clasificación. Los procesos de validación y test se realizan con las observaciones de la misma manera que en apartados anteriores. Los resultados obtenidos en las matrices de confusión para la validación y el test se muestran en las Figura 4.36 y 4.37.

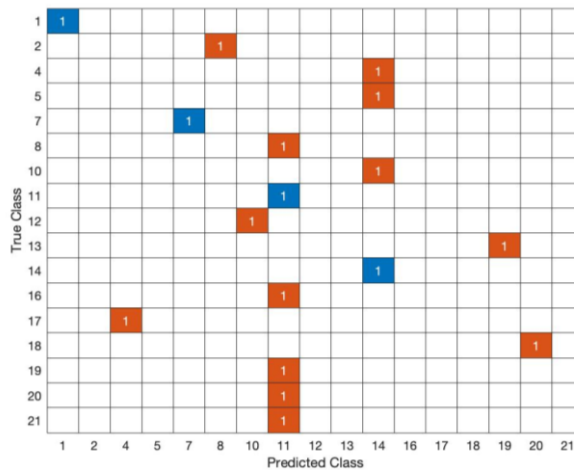


Figura 4.36. Matriz de confusión para la validación de la red LSTM de clasificación (23,5%)

Para el proceso de validación se consiguió una clasificación satisfactoria para los IDV(1), IDV(7), IDV(11) e IDV(14), lo que supone una tasa del 23,5% de acierto.

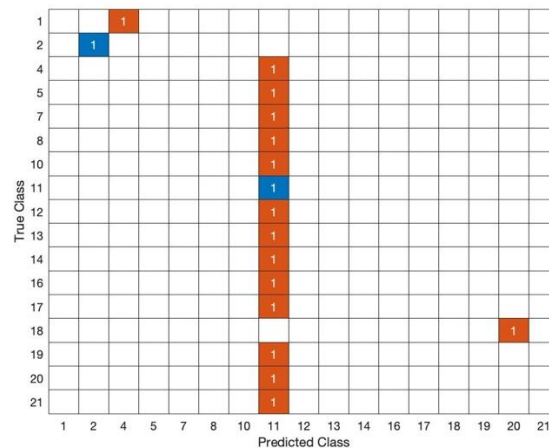


Figura 4.37. Matriz de confusión para el test de la red LSTM de clasificación (11,8%)

Los resultados para los datos de test fueron peores con una clasificación satisfactoria del 11,8% de solo dos fallos, IDV(2) e IDV(11).

Como se puede observar, la clasificación tanto para la validación como para los datos de test, ofrecen resultados muy bajos de acierto. Se supone insuficiente para un proceso de clasificación la utilización de una única simulación por fallo. A continuación, se pretende realizar el mismo proceso, con datos más extensos que suponen 500 simulaciones por fallo.

#### 4.3.2.3. Metadatos aplicados a la red LSTM de clasificación.

Hasta ahora hemos utilizado datos de fallos que nos permitían desarrollar un entrenamiento en las redes, que, aunque tenía efecto y conseguíamos buenos resultados para la fase de entrenamiento, la pérdida de eficacia para los datos de test, hacían su aplicación inviable.

Para ello una de las soluciones propuestas es la utilización de un gran volcado de datos en nuestra red LSTM. Hasta ahora utilizábamos matrices de entrenamiento con 800 observaciones de cada una de las 52 variables del proceso, para cada fallo analizado. De estas muestras una gran parte eran utilizadas para el proceso de entrenamiento, y el restante se utilizaba para el proceso de validación, previo al test.



Ahora vamos a aumentar considerablemente el tamaño de nuestra matriz de entrenamiento. La nueva matriz de fallos '*faultytraining*' para el entrenamiento está compuesta por 500 simulaciones (*simulationRun*), para cada simulación se han estimado 500 observaciones (*sample*) de 52 variables para cada fallo (*faultNumber*), de esta manera para cada una de las simulaciones tenemos una matriz de dimensiones  $10.000 \times 52$ .

Var1	faultNumber	simulationRun	sample	xmeas_1	xmeas_2	xmeas_3	xmeas_4	xmeas_5	xmeas_6
1	1	1	1	0.25038	3674	4529	9.232	26.889	42.402
2	1	1	2	0.25109	3659.4	4556.6	9.4264	26.721	42.576
3	1	1	3	0.25038	3660.3	4477.8	9.4426	26.875	42.07
4	1	1	4	0.24977	3661.3	4512.1	9.4776	26.758	42.063

Figura 4.38. Captura de las primeras 4 líneas de la matriz '*faultytraining*'

faultNumber = 1, 2, ..., 20.

simulationRun = 1, 2, ..., 500.

sample = 1, 2, ..., 500.

Lo que forma una matriz total de fallos para el entrenamiento '*faultytraining*' de  $5.000.000 \times 52$ .

Al igual que en casos anteriores vamos a proceder a la eliminación de los fallos que pueden causar más confusión en el entrenamiento de nuestra red. Las observaciones de los fallos IDV (3), IDV (9) e IDV (15) han sido eliminadas. En este caso si contamos con los datos para el entrenamiento y el test del fallo IDV (6), que, si será incluido en la simulación, por otro lado, por falta de datos se ha obviado el IDV (21).

Para los datos de entrenamiento, el procedimiento es el mismo que se viene aplicando para el entrenamiento de redes, primero debemos dividir las muestras para entrenar y reservar una proporción menor para la validación. En este caso el 80% de los datos serán utilizados para el aprendizaje profundo de nuestra red.

Tras la división de los datos, la matriz '*faultytraining*' resultante de entrenamiento será una matriz con dimensiones  $(3.400.000 \times 52)$  con 500 muestras ( $500 \times 52$ ) para cada uno de los 17 fallos, y 400 simulaciones para el



entrenamiento. Las 100 simulaciones restantes son utilizadas para la validación.

Para el testeo de la red, utilizamos otro fichero de datos llamado '*faultytesting*' compuesto por 500 simulaciones (*simulationRun*), y a diferencia de '*faultytraining*', para cada simulación se han estimado 960 observaciones (*sample*) de 52 variables para cada fallo (*faultNumber*), formando así una matriz de dimensiones  $9.600.000 \times 52$ . Al igual que realizamos en '*faultytraining*', las observaciones de los fallos IDV (3), IDV (9) e IDV (15) han sido eliminadas.

Al tratarse de una red de clasificación, la entrada a la red debe tener un formato de matriz por celdas, después de procesar la matriz '*faultytraining*' obtenemos la matriz de celdas ' $X_{Train}$ ' de dimensiones  $(6.800 \times 1)$  donde en cada una de las celdas de la única columna de la matriz hallamos una matriz double de  $(52 \times 500)$ . Donde la celda  $1 \times 1$  corresponde al IDV(1), la celda  $2 \times 1$  al IDV(2), la celda  $3 \times 1$  al IDV(4), y así sucesivamente para los 17 fallos introducidos en la red.

La matriz que introduzcamos en la red para el entrenamiento debe estar normalizada, en este caso normalizaremos nuestra matriz con la media y desviación del conjunto de datos a fin de no distorsionar las diferencias en el rango de valores de las variables. La razón de la normalización de la entrada a la red ha sido ya explicada con anterioridad, obedece a la necesidad de que las variables con mayor valor no dominen sobre otras en el proceso de entrenamiento.

Ahora debemos generar la salida deseada para el entrenamiento de nuestra red. En el proceso de clasificación de una red Long-Short-Term Memory la respuesta debe ser una variable categórica con las mismas dimensiones que la entrada a la red, en este caso debe ser una matriz de  $(6.800 \times 1)$ . Cada valor debe corresponder con el fallo al que corresponde la matriz ' $X_{Train}$ ' siendo el valor  $Y_{Train} (1,1) = 1$ ,  $Y_{Train} (2,1) = 2$ ,  $Y_{Train} (3,1) = 4$ , y así sucesivamente para los fallos introducidos en la red.



Con la salida y entrada preparadas para el entrenamiento de la red, el siguiente paso será definir nuestra red LSTM.

La red que hemos utilizado para esta simulación esta compuesta por una capa inicial denominada Sequence Input, al igual que utilizábamos para cualquier red LSTM, seguida de un conjunto de capas LSTM. En nuestro caso hemos utilizado tres capas ocultas LSTM con 52, 40 y 25 celdas o neuronas respectivamente. Todas ellas están precedidas por una capa dropout con una probabilidad asociada del 20%. Finalmente, una capa *Fully Connected* para 17 clases, capa *Softmax* y la capa final de clasificación.

Los parámetros de entrenamiento para la red LSTM utilizados en este caso han sido los que se muestran en la Tabla 12.

MaxEpochs	MiniBatchSize	InitialLearningRate	GradientThreshold
30	50	Default	1

Tabla 12. Parámetros utilizados para la red LSTM con los metadatos

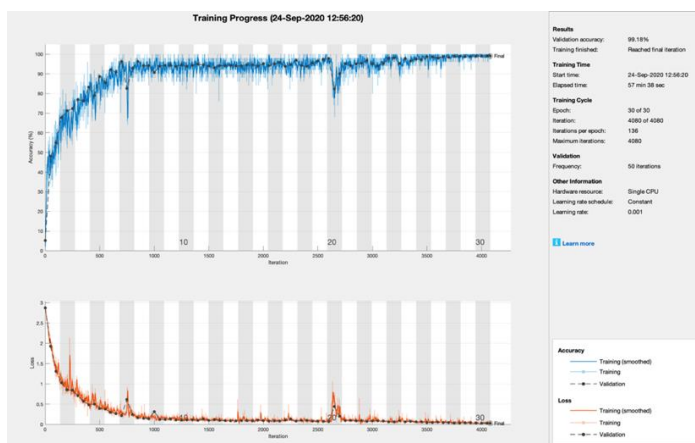


Figura 4.39. Proceso de entrenamiento de la red LSTM con los metadatos

Las matrices de confusión para los tres procesos de entrenamiento, validación y test se muestran en las Figuras 4.40 a 4.45. Los resultados de eficacia obtenidos se muestran en la Tabla 13.

Nº de Fallos	Train	Validation	Test
17	99,50%	99,20%	99,60%

Tabla 13. Resultados de la red LSTM con metadatos





## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos



Los resultados obtenidos son superiores al 99% de eficacia en la clasificación. El proceso de aprendizaje de la red ha requerido un largo tiempo de computación cercano a los 58 minutos.

**Confusion Matrix**

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NaN%
1	0	400	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
2	0	0	400	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
4	0	0	0	400	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
5	0	0	0	0	400	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
6	0	0	0	0	0	400	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
7	0	0	0	0	0	0	400	0	0	0	0	0	0	0	0	0	0	0	0	100%
8	0	0	0	0	0	0	0	400	0	0	0	0	0	0	0	0	0	0	0	100%
10	0	0	0	0	0	0	0	0	365	0	0	0	0	1	0	0	0	0	0	99.7%
11	0	0	0	0	0	0	0	0	0	400	0	0	0	0	0	0	0	0	0	100%
12	0	0	0	0	0	0	0	0	0	0	400	0	0	0	0	0	0	0	0	100%
13	0	0	0	0	0	0	0	0	0	0	0	400	0	0	0	0	0	0	0	100%
14	0	0	0	0	0	0	0	0	0	0	0	0	400	0	0	0	0	0	0	100%
16	0	0	0	0	0	0	0	0	0	0	0	0	0	35	0	0	0	0	0	91.9%
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	399	0	0	0	0	8.1%
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	400	0	0	0	100%
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	400	0	0	100%
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	400	0	100%
	NaN%	100%	100%	100%	100%	100%	100%	100%	91.2%	100%	100%	100%	100%	99.8%	100%	100%	100%	100%	100%	99.5%
	0.0%	5.9%	5.9%	5.9%	5.9%	5.9%	5.9%	5.9%	8.8%	5.9%	5.9%	5.9%	5.9%	0.2%	5.9%	5.9%	5.9%	5.9%	5.9%	0.5%

Figura 4.40. Matriz de confusión para el entrenamiento de la red LSTM de clasificación

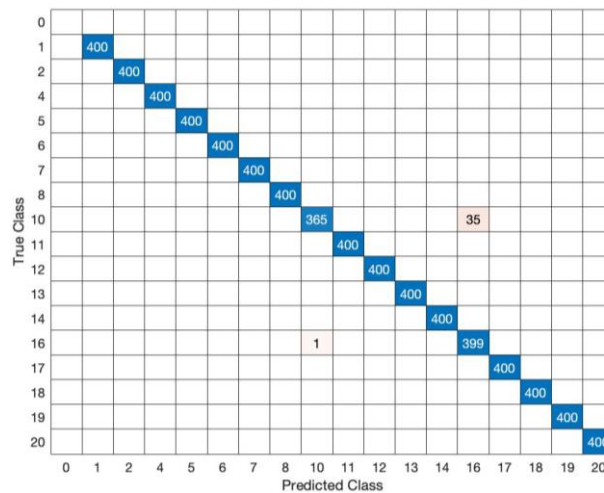


Figura 4.41. Matriz de confusión para el entrenamiento de la red LSTM de clasificación



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos



**Confusion Matrix**

Output Class	0	1	2	4	5	6	7	8	10	11	12	13	14	16	17	18	19	20	
0	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN%
1	0 0.0%	100 5.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	0 0.0%	100 5.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	100 5.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 5.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 5.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 5.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 5.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	86 5.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
11	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 5.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
12	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 5.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
13	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 5.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
14	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 5.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
16	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	14 0.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 5.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	87.7% 12.3%
17	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 5.9%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
18	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 5.9%	0 0.0%	0 0.0%	100% 0.0%
19	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 5.9%	0 0.0%	100% 0.0%
20	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 5.9%	100% 0.0%
	NaN%	100%	100%	100%	100%	100%	100%	86.0%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	99.2%
	NaN%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	14.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.8%

Figura 4.42. Matriz de confusión para la validación de la red LSTM de clasificación

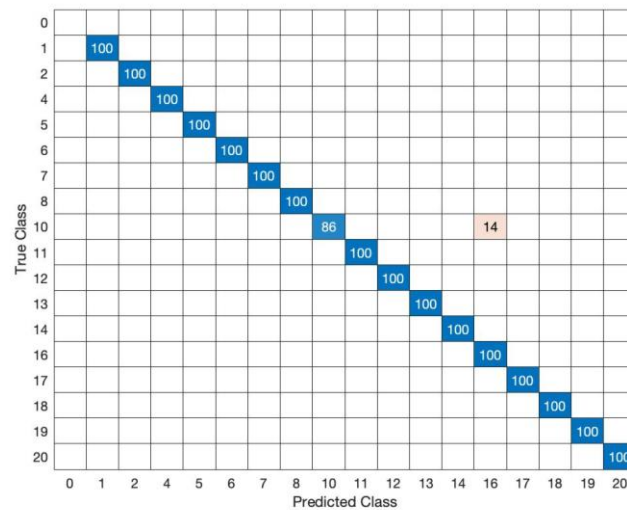


Figura 4.43. Matriz de confusión para la validación de la red LSTM de clasificación



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos



Figura 4.44. Matriz de confusión para el proceso de test de la red LSTM de clasificación

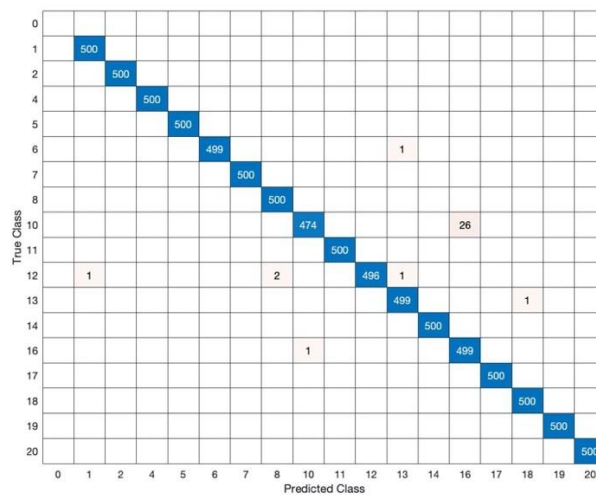


Figura 4.45. Matriz de confusión para el proceso de test de la red LSTM de clasificación



En este apartado se contrastan los diferentes resultados obtenidos para las diferentes técnicas de aprendizaje supervisado aplicados a lo largo del trabajo. Se han extraído los mejores resultados de cada técnica, los resultados para la situación con 17 fallos se muestran en la Tabla 14.

Técnica	Train	Validation	Test
ANN de propagación hacia adelante una capa	76,20%	75,30%	67,30%
ANN de propagación hacia adelante multicapa	84,50%	81,10%	73,40%
ANN de propagación hacia adelante Metadatos	77,30%	77,50%	68,40%
Red LSTM Regresión (Tipo II)	100,00%	15,60%	6,90%
Red LSTM Clasificación	100,00%	23,50%	11,80%
Red LSTM Clasificación Metadatos	99,50%	99,20%	99,60%

**Tabla 14.** Resumen de resultados para la clasificación de 17 fallos para las diferentes técnicas de aprendizaje supervisado aplicadas

Las redes neuronales de propagación hacia adelante arrojan resultados considerablemente buenos, aunque insuficientes para una clasificación fiable. Los resultados óptimos los encontramos con la aplicación de una red LSTM con un aprendizaje a partir de unos ficheros de datos mayores que los aplicados a las redes de propagación hacia adelante. Los resultados obtenidos con los datos iniciales para las redes LSTM ofrecen resultados insuficientes.



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos





## CAPÍTULO V: CONCLUSIONES Y TRABAJO FUTURO



## 5.1. CONCLUSIONES.

La irrupción de la Industria 4.0 ha aportado recursos y técnicas a la industria para convertirla en una fábrica inteligente. La gran cantidad de datos y mediciones extraídos de los procesos aporta información a las organizaciones en el soporte de la toma de decisiones, la importancia del big data en la actualidad requiere de aplicaciones y técnicas sofisticadas en el procesamiento de la información. En este trabajo se han utilizado estos datos para el desarrollo de métodos de detección y diagnóstico de fallos, en el contexto del control de calidad, siendo el principal objetivo trabajar con técnicas estadísticas como es el PCA para detectar fallos y varias técnicas de aprendizaje automático para su posterior clasificación y diagnóstico. La experimentación se ha basado en unos datos provenientes de un modelo estándar de proceso, la planta Tennessee Eastman.

En primer lugar, para la detección de fallos, en este trabajo se ha creado una aplicación para el control estadístico de procesos basado en el análisis de componentes principales para convertir un análisis de multitud de variables en un estudio univariante del que podemos extraer unos umbrales aptos para la detección de fallos en el modelo de la planta química Tennessee Eastman.

Los resultados han sido satisfactorios en casi la totalidad de los fallos tratados, con problemas de detección en los fallos conocidos como incipientes, los cuales han ofrecido resultados no concluyentes en su detección.

Para el diagnóstico de fallos (una vez que se detecta que ha habido un fallo, hay que decidir que tipo de fallo es para su posible eliminación y recuperación del proceso) se han utilizado técnicas de clasificación (donde en este caso cada clase es un tipo de fallo), y en concreto se ha trabajado con distintos tipos de redes neuronales artificiales. Por tanto, apoyándonos en los datos de fallo de la planta se han utilizado técnicas de aprendizaje supervisado basándonos en dos tipos de redes neuronales, las primeras de propagación hacia delante han ofrecido resultados muy satisfactorios para la clasificación en situaciones con no más de 7 fallos concurrentes. El éxito en la clasificación disminuye cuando



se pretende clasificar un número mayor de fallos a la vez. Las segundas redes desarrolladas en el trabajo, las redes LSTM, un caso especial de redes neuronales recurrentes, han ofrecido resultados nada concluyentes en la clasificación a partir de los datos de partida, debido a que este tipo de red neuronal incluida en el nuevo paradigma de Deep-learning necesitan muchos más datos para obtener un entrenamiento adecuado. Resultados muy diferentes han proporcionado este tipo de redes cuando hemos volcado datos mucho más extensos para el entrenamiento de la red LSTM, alcanzándose un 99% de eficacia en la clasificación de los 17 tipos de fallos que presenta la planta TEP.

## **5.2. TRABAJO FUTURO.**

Como futuros trabajos en este campo se sugiere continuar experimentando con redes LSTM, o con otro tipo de redes recurrentes en la clasificación de fallos. Los parámetros utilizados en este trabajo para este tipo de redes no han sido modificados con respecto a otros estudios, y se podrían usar diferentes combinaciones que mejoren el rendimiento o reduzcan el tiempo de computación necesario para conseguir una eficacia mayor. El estudio de otros tipos de arquitecturas de redes de aprendizaje profundo como redes Convolucionales, o el desarrollo de métodos que complementen la capacidad de las diferentes arquitecturas de las redes pueden aportar mejoras en el campo de la clasificación de fallos en procesos.





## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos





## BIBLIOGRAFÍA

- [1] V. Roblek, M. Mesko y A. Krapez, «A Complex View of Industry 4.0 SAGE Open,» 1 Junio 2016. [En línea]. Available: <https://doi.org/10.1177/2158244016653987..> [Último acceso: Octubre 2020].
- [2] J. Cooper y A. James, «Challenges for database management in the internet of things,» *IETE Technical Review*, vol. 26, nº 5, 2009.
- [3] M. A. Carboneras Checa y M. Bestratén Belloví, «"Integración de sistemas de gestión: prevención de riesgos laborales, calidad y MA",» Ministerio de Trabajo y Asuntos Sociales, España, 200#.
- [4] ISOTools, «ISOTools Excellence,» Octubre 2020. [En línea]. Available: <https://www.isotools.org>.
- [5] Á. Pequeño Alonso, «Mejora del control de calidad de un proceso mediante técnicas de aprendizaje automático. Trabajo Fin de Grado en Ingeniería Química.,» EII, Universidad de Valladolid, 2020.
- [6] D. Garcia-Álvarez y M. Fuente, «Análisis Comparativo de Técnicas de Detección de Fallos utilizando PCA,» *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 8, pp. 182-195, 2011.
- [7] W. A. Shewhart, *Economic Control of Quality of Manufactured Product*, New York: D. Van Nostrand Company, 1931.
- [8] A. Ruiz-Falcó Rojas, «Control Estadístico de Procesos,» Universidad Pontificia Comillas, 2006. [En línea]. Available: [https://www.academia.edu/9602972/CONTROL\\_ESTADÍSTICO\\_DE\\_PROCESOS\\_Control\\_Estad%C3%ADstico\\_de\\_Procesos](https://www.academia.edu/9602972/CONTROL_ESTADÍSTICO_DE_PROCESOS_Control_Estad%C3%ADstico_de_Procesos). [Último acceso: Octubre 2020].
- [9] R. Carro Paz y D. González Gómez, «Control Estadístico de Procesos,» 2012. [En línea]. Available: <http://nulan.mdp.edu.ar/1617/>. [Último acceso: Octubre 2020].



- [10] C. Huerga Castro, P. Blanco Alonso y J. Abad González, «Aplicación de los gráficos de control en el análisis de la calidad textil,» *Pecunia: Revista de la Facultad de Ciencias Económicas y Empresariales, Universidad de León*, vol. 0, nº 1, pp. 125-148, 1 Diciembre 2005.
- [11] R. Barzola, O. Honorio y F. Aparisi García, «Gráficos de Control de Calidad Multivariantes con Dimensión Variable,» Universidad Politécnica de Valencia, Valencia, 2013.
- [12] F. Aparisi y J. C. Garcia Díaz, «Aumento de la potencia del gráfico multivariante de Hotelling utilizando señales adicionales de falta de control,» *Estadística española*, vol. 43, nº 148, pp. 171-188, 2001.
- [13] A. Cursillo Ngua, «Diseño de sensores software no lineales para la estimación de las variables de calidad de los procesos, Trabajo Fin de Grado de Ingeniería Electrónica Industrial y Automática,» EII, Universidad de Valladolid, Valladolid, 2019.
- [14] D. Peña, *Análisis de Datos Multivariante*, McGraw-Hill, 2002.
- [15] A. Sanchez-Fernandez, M. Fuente y G. Sainz-Palmero, «Fault Detection in Wastewater Treatment Plants using Distributed PCA Methods,» de *20th IEEE International Conference on Emerging Technologies and Factory Automation, EFTA'2015*, Bilbao, 2015.
- [16] I. Goodfellow, Y. Bengio y A. Courville, *Deep Learning*, Cambridge, MA, USA: The MIT Press, 2016.
- [17] bismart, «bismart,» [En línea]. Available: <https://bismart.com/es/nosotros/>. [Último acceso: 12 octubre 2020].
- [18] X. Olabe Basogain, «Redes Neuronales Artificiales y sus Aplicaciones,» Escuela Superior de Ingeniería de Bilbao, EHU, [En línea]. Available: [https://ocw.ehu.eus/file.php/102/redes\\_neuro/contenidos/pdf/libro-del-curso.pdf](https://ocw.ehu.eus/file.php/102/redes_neuro/contenidos/pdf/libro-del-curso.pdf). [Último acceso: Octubre 2020].



- [19] M. Figueres Moreno, «Introducción a las Redes Neuronales Artificiales,» [En línea]. Available:  
<https://optimizacionheuristica.blogs.upv.es/files/2013/04/Introducción-Redes-Neuronales-ArtificialesMFM.pdf>. [Último acceso: Octubre 2020].
- [20] A. González Muñiz, «Aplicaciones de técnicas de inteligencia artificial basadas en aprendizaje profundo (deep learning) al análisis y mejora de la eficiencia de procesos industriales,» Universidad de Oviedo, 2018. [En línea]. Available:  
<https://digibuo.uniovi.es/dspace/handle/10651/45097>. [Último acceso: Octubre 2020].
- [21] H. Hao y Z. Jinsong, «Deep convolutional neural network model based chemical process fault diagnosis,» *Computers & Chemical engineering*, vol. 115, pp. 185-197, 2018.
- [22] R. Lachman, J. L. Lachman y E. C. Butterfield, *Cognitive psychology and information processing*, Routledge, 1979.
- [23] C. Olah, «Understanding LSTM Networks,» 27 Agosto 2015. [En línea]. Available:  
<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. [Último acceso: 3 Octubre 2020].
- [24] M. Márquez-Vera, O. López-Ortega, L. Ramos-Velasco, R. Ortega-Mendoza, B. Fernández Neri y N. Zúñiga-Peña, «Diagnóstico de Fallas mediante una LSTM y una Red Elástica,» *Revista Iberoamericana de Automática e Informática Industrial* (In press), 2020. [En línea]. Available:  
<https://polipapers.upv.es/index.php/RIAI/article/view/13611>. [Último acceso: Octubre 2020].
- [25] J. Downs y V. E. F., «A Plant-Wide Industrial Process Control Problem,» vol. 17, nº 3, pp. 245-255, 1993.



- [26] MathWorks, «MathWorks,» [En línea]. Available: [https://es.mathworks.com/help/deeplearning/ref/classificationlayer.html?s\\_tid=srchtitle](https://es.mathworks.com/help/deeplearning/ref/classificationlayer.html?s_tid=srchtitle). [Último acceso: Octubre 2020].



## Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos

