



**Universidad de Valladolid**



**ESCUELA DE INGENIERÍAS  
INDUSTRIALES**

**UNIVERSIDAD DE VALLADOLID**

**ESCUELA DE INGENIERIAS INDUSTRIALES**

**Grado en Ingeniería en Tecnologías Industriales**

**DETECCIÓN Y DIAGNÓSTICO DE  
FALLOS MEDIANTE OPERADORES  
LINGÜÍSTICOS**

**Autor:**

**Vallejo Yuste, Julián**

**Tutor:**

**Sainz Palmero, Gregorio Ismael  
Departamento de Ingeniería  
de Sistemas y Automática**

**Valladolid, Octubre de 2018**



# Resumen

La realización de este trabajo tiene como objetivo principal diseñar una metodología basada en *Machine Learning e Inteligencia Computacional (IC)* que haga posible la detección y diagnóstico de los fallos de una planta depuradora de aguas residuales a partir de información recogida por sistemas de monitorización. Con esta meta se tratarán de buscar los modelos y metodologías basados en IC más idóneos para dicha tarea. Por otro lado también se realizará un proceso de toma de decisiones basado en operadores lingüísticos.

# Summary

The aim of this project is to design a methodology that allows the detection and diagnosis of the faults of a sewage treatment plant through an automated system. In order to achieve this goal, we are going to try to find the most suitable models for this task by applying data processing and treatment techniques and systems based on neural networks. In addition, different linguistic operators will be used to make decisions.

# Palabras clave

Inteligencia Computacional, Detección de fallos, Diagnostico de fallos, Toma de decisiones, OWA

# Keywords

Computational Intelligence, Fault detection, Fault diagnosis, Decision making, OWA



# Índice general

Índice general.....	5
<b>1. Introducción y objetivos .....</b>	<b>7</b>
1.1. Marco del proyecto.....	7
1.2. Motivación .....	8
1.3. Objetivos .....	9
1.4. Organización de la memoria.....	9
<b>2.Aspectos teóricos.....</b>	<b>11</b>
2.1. Detección de fallos.....	11
2.2. Procesamiento de los datos .....	13
2.3. Selección de variables: Análisis de componentes principales (PCA).....	16
2.4. IC: Redes neuronales, neurodifusas y toma de decisiones OWA.....	18
2.4.1. Redes neurodifusas .....	18
2.4.1.1. ANFIS .....	20
2.4.2. Redes neuronales.....	21
2.4.2.1. Clasificación de las redes neuronales .....	22
2.4.2.1. Red neuronal de base radial (RBF) .....	24
2.4.2.2. Backpropagation.....	25
2.4.3. Toma de decisiones: Operadores OWA.....	26
<b>3.Técnicas de IC &amp; ML: Una propuesta.....</b>	<b>29</b>
3.1. Etapa 1: Procesamiento de los datos .....	31
3.2. Etapa 2: Selección de variables.....	31
3.2.1. Análisis de componentes principales (PCA) .....	31
3.2.2. Selección de variables: Reducción de la dimensionalidad .....	32
3.2.3. Selección de variables: Relevancia de las variables .....	33
3.2.4. Ranking de variables. Método OWA.....	34
3.3. Etapa 3: Modelado basados en Redes Neuronales.....	35
3.4. Etapa 4: Toma de decisiones .....	37

<b>4. Trabajo experimental.....</b>	<b>41</b>
4.1. Caso de estudio: Proceso Tennessee Eastman (TEP) .....	41
4.2. Metodología experimental .....	44
4.2.1. Datos de partida.....	44
4.2.2. Metodología utilizada.....	45
4.3. Resultados.....	61
4.3.1. Procesamiento de datos .....	61
4.3.2. Selección de variables .....	62
4.3.2.1. Análisis de componentes principales .....	62
4.3.2.2. Selección de variables .....	65
4.3.2.3. Ranking de variables: Método OWA.....	65
4.3.3. Modelos de detección y diagnóstico de fallos.....	67
4.3.3.1. Modelos globales de FDI .....	67
4.3.3.2. Modelos parciales de FDI .....	73
4.3.4. FDI: Toma de decisiones.....	81
<b>5. Conclusiones .....</b>	<b>83</b>
<b>6. Bibliografía.....</b>	<b>85</b>
<b>Índice de figuras .....</b>	<b>88</b>
<b>Índice de tablas .....</b>	<b>91</b>
<b>Anexo A: Tablas de relevancia de las variables.....</b>	<b>92</b>
<b>Anexo B: Rankings de variables .....</b>	<b>97</b>
<b>Anexo C: Gráficas de detección de fallos en modelos globales.....</b>	<b>100</b>
<b>Anexo D: Gráficas de errores de test en modelos parciales .....</b>	<b>108</b>

# Capítulo 1

## Introducción y objetivos

Este Trabajo de Fin de Grado ha sido realizado en el Departamento de Ingeniería de Sistemas y Automática de la Escuela de Ingenierías Industriales de la Universidad de Valladolid.

En él se va a abordar la detección y diagnóstico de fallos mediante técnicas de Inteligencia Computacional (IC). Además se propone una metodología para llevar a cabo estas tareas en una planta de tratamiento de aguas residuales, describiendo todos los procedimientos utilizados para su alcance.

### 1.1. Marco del proyecto

Actualmente una de las partes más importantes en lo referente a competitividad en la industria reside en la reducción de costes. Esto implica la utilización de procesos más baratos, automatización, disminución de residuos, cambio de equipos,... pero también abaratar las tareas de mantenimiento.

Todos los equipos están sujetos a diversos tipos de incidentes por varios motivos, lo que implica la necesidad de monitorización de los procesos que realizan para evaluar cuál es su rendimiento y garantizar que su funcionamiento se encuentra dentro de los márgenes de lo previsto. (García-Alvarez & Fuente, 2011)

Además, gracias a la etapa de diagnóstico, estas técnicas pueden conseguir que se eviten un gran número de averías que encarecerían mucho los procesos e incluso podrían causar grietas en la seguridad de cualquier sistema productivo.

Esta tarea puede dividirse en varias fases diferentes (Puigjaner, Ollero, Prada, & Jiménez, 2006):

- En la primera de ellas se **detecta la existencia de un fallo** en el sistema a partir de los datos recogidos por los sensores.
- Posteriormente se deben **identificar cuáles son las variables más relevantes** para su diagnóstico.
- Después tiene lugar la **diagnóstico del fallo** en el que este se somete a una clasificación donde se determina el tipo de fallo que es, donde se encuentra, el tamaño del mismo y la causa que lo ha producido.
- La última fase consiste en la **recuperación del funcionamiento normal** del proceso en sí, es decir, la eliminación del fallo.

## 1.2. Motivación

Este TFG se enmarca dentro del proyecto denominado Desarrollo de técnicas de control predictivo jerárquico basado en sistemas multiagentes. Aplicación a sistemas de gran escala, DPI2015-67341-C2-2-R, Ministerio de Economía y Competitividad.

En este proyecto, uno de los temas que se abordan son las técnicas para la detección y diagnóstico de fallo mediante técnicas inteligentes en distintos tipos de plantas industriales.

En este contexto, el presente TFG aborda el problema de la detección de fallos mediante *Inteligencia computacional* (IC), tanto para el modelado como para la toma de decisiones en un proceso FDI (*Fault detection and isolation*).

En la actualidad, cualquier inconveniente que obligue a parar un proceso productivo supone unas pérdidas económicas muy elevadas para la empresa encargada de esa producción. Incluso podría suponer una pérdida de seguridad del equipo que trabaja en esa planta.

El hecho de monitorizar un proceso industrial con el fin de encontrar esos fallos o incidencias que pueda sufrir el sistema puede suponer un gran avance tanto en términos económicos como de seguridad.

Una correcta detección y diagnóstico de fallos en la planta puede resultar costoso económicamente pero es muy fácilmente amortizable y las ventajas son muchas. Sin embargo el sistema que permita esa detección debe ser el idóneo para el proceso y debe aportar una cierta fiabilidad de que los fallos van a ser detectados.



### 1.3. Objetivos

Los principales **objetivos** que se buscan en este TFG son:

- Diseño de una metodología basada en datos para la detección y diagnóstico de fallos.
- Conocer todas las etapas desde el procesamiento de datos, la modelización y la toma de decisiones necesarias, así como técnicas de *Inteligencia Computacional (IC)* y *Machine Learning (ML)* que permitan llevarlas a cabo.
- Implementar estas técnicas y conocer los retos que suponen.
- Conocer cómo llevar a cabo un proceso de experimentación en un caso de *benchmarking*.
- Aplicar todo lo anterior al caso de una planta de tratamiento de aguas residuales Tennessee Eastman (*Richard Braatz, 2002*).

### 1.4. Organización de la memoria

El contenido de esta memoria se ha organizado de la siguiente manera:

- **Capítulo 2: Aspectos teóricos.** En el segundo capítulo se describen de forma resumida los principales conceptos teóricos de las técnicas que van a ser utilizadas para resolver nuestro problema.
- **Capítulo 3: Técnicas de IC & ML: Una propuesta.** En este capítulo se describe cual es la metodología y sus etapas para afrontar la tarea de detección de fallos en base a técnicas de Inteligencia computacional (IC) y *Machine learning (ML)*.
- **Capítulo 4: Trabajo experimental.** En este capítulo se presenta el *benchmark* utilizado y cuáles son los resultados y análisis de aplicar la propuesta realizada en el capítulo anterior a dicho *benchmark*.
- **Capítulo 5: Conclusiones.** En el quinto capítulo se indican cuales son las conclusiones del trabajo realizado en los anteriores capítulos aportando un pequeño resumen de los resultados.

- **Capítulo 6: Bibliografía.** Finalmente se indican cuales han sido las referencias bibliográficas utilizadas en la realización de este trabajo

## Capítulo 2

### Aspectos teóricos

En este capítulo se describen de forma resumida los fundamentos teóricos de las técnicas utilizadas.

#### 2.1. Detección de fallos

Antes de hablar de la detección de fallos como tal debemos describir que se considera exactamente un fallo. Un **fallo** se considera un cambio que ocurre de forma inesperada en un sistema productivo, que puede deberse o no a un defecto en sus componentes (Tapia, 2017). Estos cambios deben ser estudiados para tomar las medidas oportunas y prevenir sus consecuencias.

El campo relacionado con la detección de fallos se ha convertido en uno de los más activos en cuanto a investigación en los últimos años. Su principal objetivo es el descubrimiento de funcionamientos anómalos en un proceso. Estos fallos se propagan de forma rápida en los sistemas pudiendo llegar a dañar los equipos, lo que ha fomentado la búsqueda de técnicas para su detección y para el aislamiento de las variables que intervienen en los mismos.

Además a esta tarea se le une la de diagnóstico de fallos, encargada de buscar el verdadero origen de ese funcionamiento anormal. Ambas cosas unidas permitirán conseguir procesos industriales más óptimos y eficientes.

Existen varios tipos de técnicas de detección de fallos. El primer tipo son las **basadas en los datos**, que tan solo necesitan el historial de datos recogidos por los sensores de la planta, siendo un proceso mucho más rápido y fácil en cuanto a desarrollo. En estas técnicas se pueden englobar:

- **Valores límite:** Se establecen unos valores límites a partir de los cuales el sistema estaría presentando un fallo. Estos límites dependen del historial de los datos recogidos en la planta.
- **Técnicas estadísticas multivariantes:** En plantas con una gran cantidad de variables estas están relacionadas entre sí. Por ello se pueden utilizar estas relaciones para la detección de fallos. Algunas de estas técnicas son el Análisis de componentes principales (PCA), el Análisis del discriminante de Fisher (FDA),...
- **Redundancia física:** Se compara la señal de salida con la de otros elementos que sean iguales.
- **Análisis de frecuencia:** Todos los fallos pueden estar representados por un patrón de amplitudes y frecuencias que lo caracterizan. Esos patrones pueden ayudar en su detección.

Otro tipo de técnicas de detección de fallos son las **basadas en métodos analíticos**. Se caracterizan por su robustez y se basan en el procesamiento, mediante determinados algoritmos, de las entradas y salidas calculando así su diferencia. Cuando esta diferencia es distinta a cero se podría estar produciendo un fallo. En una segunda etapa de la técnica se examina esa diferencia para decidir si existe o no fallo.

El último tipo de técnicas son las **basadas en el conocimiento**:

- **Lógica difusa:** Son sistemas que hacen posible el tratamiento de información imprecisa. Realizan la combinación de diversas variables de entradas definidas en términos de conjuntos borrosos a través de un conjunto de reglas que produce una o varias salidas. (Pueyo, 2005)
- **Redes neuronales:** Se basan en modelar la forma de procesamiento de la información en sistemas nerviosos biológicos (Izaurieta & Saavedra). Cuentan con un proceso previo de aprendizaje en el que el modelo adquiere la experiencia necesaria para detectar patrones de comportamiento similares. El principal hándicap de este método es que necesita de datos de todos los modos de funcionamiento.
- **Búsqueda causal:** Es una técnica centrada en el seguimiento de cada uno de los fallos hasta su origen o causa. Se basa en la representación de la causalidad mediante grafos directos dirigidos. (Tapia, 2017)
- **Sistemas expertos:** Relacionan los síntomas que pueden apreciarse en los datos con la causa que lo ha producido a través de la formación de reglas.
- **Algoritmos genéticos:** Son técnicas adaptativas muy robustas utilizadas para la resolución de problemas de búsqueda, aprendizaje y optimización. Están inspirados en la evolución natural y genética de los organismos vivos. (Vasco, 2015)

## 2.2. Procesamiento de los datos

El principal objetivo del análisis de un conjunto de datos es la extracción de conclusiones para la toma de decisiones o resolución de problemas. El inconveniente es que en numerosas ocasiones las muestras contienen algunos datos incompletos, ruido o inconsistencias que se detectan en el propio proceso de análisis, con los gastos económicos y de tiempo que eso conlleva. Por ello se han desarrollado distintas técnicas orientadas a la preparación de los datos para su análisis.

Por tanto la **motivación** de esta tarea es la manipulación de los datos en bruto para que el contenido de información en los mismos sea más accesible y su análisis permita llegar a mejores conclusiones.

Los principales **problemas** que pueden encontrarse en una muestra de datos son los siguientes (Hernández & Rodríguez, 2009):

- **Datos con ruido:** Se puede deber a errores en diversas medidas tomadas por los sensores, o a errores en la transmisión de los datos, o a características propias del sistema.
- **Extracción de atributos:** En ocasiones en la muestra no se han registrado algunos datos por no considerarse importantes durante la realización del propio registro.
- **Existencia de datos irrelevantes:** A veces el conjunto de datos contiene algunos registros parecidos a otros ya incluidos. Así, la complejidad de la muestra puede reducirse si los registros irrelevantes son eliminados y solo se analizan los más importantes.
- **Cantidad de datos elevada:** Otro de los factores importantes es el volumen de los datos de la muestra, que puede ser determinante en la velocidad del análisis y las características de los equipos que deben realizarlo.
- **Atributos faltantes o insuficientes:** Es uno de los principales problemas en algoritmos de aprendizaje.
- **Valores de atributos faltantes:** En ocasiones en algunos de los datos registrados existen atributos que no están completos en su totalidad.
- **Poca cantidad de datos:** Se considera otro de los principales problemas en algoritmos de aprendizaje, para los que la cantidad de datos debe ser elevada.
- **Múltiples fuentes de datos:** En sistemas en los que existen diversas fuentes encargadas de coleccionar los datos pueden producirse algunos inconvenientes para la unificación de los mismos.
- **Datos desde múltiples niveles de granularidad:** Si los datos provienen de varios niveles de granularidad puede complicar su comparación y análisis.

Para dar **solución** a estos posibles problemas y favorecer la realización de su análisis se llevan a cabo diversas técnicas que permitan mejorar la calidad de la muestra y que permitan obtener mayor cantidad y mejor información. Estas son la limpieza, integración, transformación y reducción de la muestra.

La primera tarea a realizar es la **limpieza de datos**. Esta etapa consiste en el llenado de los datos que faltan, la corrección de los que sean inconsistentes y la resolución de posibles redundancias.

Para poder detectar anomalías en los datos existen distintos algoritmos basados en *clustering* como el Algoritmo de agrupación jerárquica, o el Algoritmo DBSCAN.

En cuanto a los **datos faltantes (*missing values*)**, el primer paso es identificar el motivo de la pérdida de esos valores. La pérdida puede clasificarse en (Rubin, 1987):

- Missing completely at random: La probabilidad de pérdida es la misma para el conjunto de los datos.
- Missing at random: La probabilidad de pérdida de un dato depende de la información observada.
- Missing not at random: La probabilidad de pérdida está condicionada por los valores perdidos.

Para la solución del problema de pérdida de datos o datos faltantes existe un gran número de alternativas. Algunas de ellas son:

- Análisis teniendo en cuenta únicamente las muestras en la que todas las variables contienen información.
- Análisis con el conjunto de datos, incluidos aquellos en los que no todas las variables contienen información. El principal problema de esto es que algunas variables contarán con mayor información que las demás.
- Ponderación: En casos en los que ninguna de las variables contiene información se le da más peso a aquellas muestras que están completas.
- Uso de imputación numérica: Los datos se completan basándose en el resto de información del resto del conjunto.

En este apartado también se incluye la búsqueda de valores que difieren mucho del resto (**outliers**). Para la detección de estos valores anómalos existen diferentes métodos.

Alguno de los más utilizados son el de Boxplot, que considera como outliers leves aquellos valores que se encuentren a 1,5 veces el valor intercuartil y extremos los que estén a 3 veces ese valor.

También se puede utilizar el método de Grubbs que utiliza el siguiente estadístico:

$$T = \frac{|X - \bar{X}|}{s} \quad (\text{Ec. 1})$$

La segunda fase de procesamiento de los datos es la de **integración**. Su principal objetivo es prevenir las inconsistencias y redundancias en los datos y mejorar la calidad y velocidad de análisis de los mismos. Para ello se puede recurrir al análisis correccional.

Esta técnica consiste en medir cuanto están correlacionados dos datos, A y B. Para ello se utiliza la siguiente expresión:

$$\frac{P(A \wedge B)}{P(A)P(B)} \quad (\text{Ec. 2})$$

Si el resultado de esta operación es mayor que la unidad ambos datos están correlacionados positivamente. Sin embargo, si el resultado es menor la correlación será negativa. Si es igual a la unidad no existe correlación entre A y B.

La denominada **transformación de datos** (3ª fase de procesamiento) incluye varias tareas como la generalización, la agregación, el suavizado o la normalización de la muestra.

La generalización consiste en la agrupación de los datos de bajo nivel en otros conceptos de más alto nivel.

En cuanto al proceso de agregación, este basa en la realización de operaciones de síntesis o agregación de datos. Se suele utilizar con el fin del análisis de datos de distintos niveles de granularidad.

Las técnicas de suavizado persiguen la reducción del ruido de los datos mediante técnicas como el *binning*, *clustering* o la regresión.

Por último, la normalización consiste en el escalado de los atributos a un rango pequeño de valores. Es muy útil en casos en los que se estudian diferentes variables y cada una de ellas toma valores en una escala distinta a las demás. Existen diferentes técnicas para realizar este proceso como por ejemplo la normalización Min Max o la normalización z-score.

La **normalización Min- máx** se basa en la realización de una transformación lineal de los datos de partida. De esta forma se obtiene el valor normalizado a partir del máximo y mínimo de los datos.

Así para cada dato se procede de esta forma:

$$\text{valor normalizado} = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)} \quad (\text{Ec. 3})$$

La principal ventaja de este tipo de normalización es que conserva la relación entre los datos originales.

En cuanto a la técnica de **normalización z-score** se realiza en función de la media y desviación estándar de los datos. Es una de las más utilizadas a nivel mundial y se implementa para cada variable de la siguiente forma:

$$\text{valor normalizado} = \frac{x_i - \mu}{\sigma} \quad (\text{Ec. 4})$$

Este tipo de normalización se utiliza cuando no se conocen los valores máximo y mínimo de los datos.

Por último, a las tareas de procesamiento se une la de **reducción de los datos**. El objetivo de esto es obtener una representación reducida de la muestra sin perder su integridad.

Existen numerosas técnicas que se utilizan para esta reducción:

La primera consiste en una reducción de la dimensión, con la que se detectan y eliminan atributos que no son muy relevantes en el conjunto.

Por otra parte se utilizan técnicas de compresión de datos, que intentan reducir su tamaño sin que su variabilidad se vea afectada. Un ejemplo de estas técnicas puede ser el Análisis de componentes principales (PCA) o la transformada de wavelet (DWT).

También se puede llevar a cabo una reducción de la numerosidad, en la que los datos pequeños se sustituyen por modelos paramétricos de los mismos. Algún ejemplo de este tipo de técnicas pueden ser el *clustering* o el muestreo.

Por último, en ocasiones se utiliza la discretización. Esta técnica reemplaza los datos por rangos en los que se encuentran los mismos.

### **2.3. Selección de variables: Análisis de componentes principales (PCA)**

El análisis de componentes principales (PCA) es una técnica estadística de síntesis que puede permitir, entre otras cosas, afrontar la reducción de la dimensionalidad de un conjunto de datos de partida formado por un gran número de variables interrelacionadas entre sí.

Las principales ventajas de la reducción de la dimensionalidad son la disminución del costo computacional de los equipos que realizan los análisis



permitiendo que estos sean más rápidos y la mejor calidad de los resultados de minería de datos por la disminución del ruido en la muestra. (Zapana, 2018)

Esto se consigue mediante la transformación de las variables en un nuevo conjunto de ellas, que ya no estén interrelacionadas entre sí pero que intenten conservar la mayor cantidad posible de variabilidad de los datos.

Con esta transformación los datos originales se proyectan en un nuevo sistema de referencia. Este nuevo sistema contará con un primer eje (Primer componente principal) que capturará la mayor variabilidad de los datos. Un segundo eje contará con los datos que tienen la segunda variabilidad más alta. De esta forma el nuevo sistema contará con varios ejes correspondientes a diferentes variabilidades de la información.

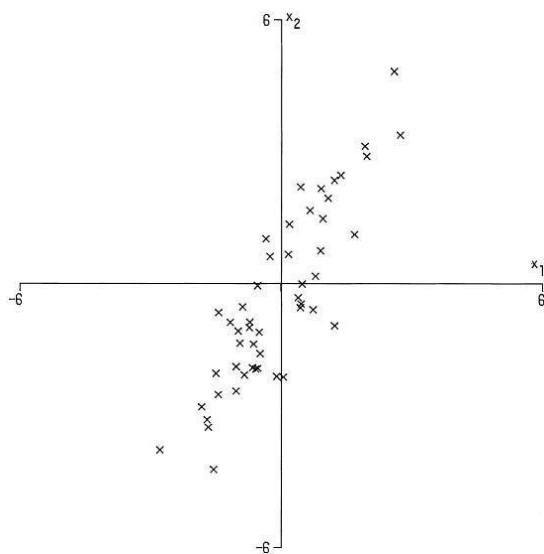


Fig 1: Datos de partida

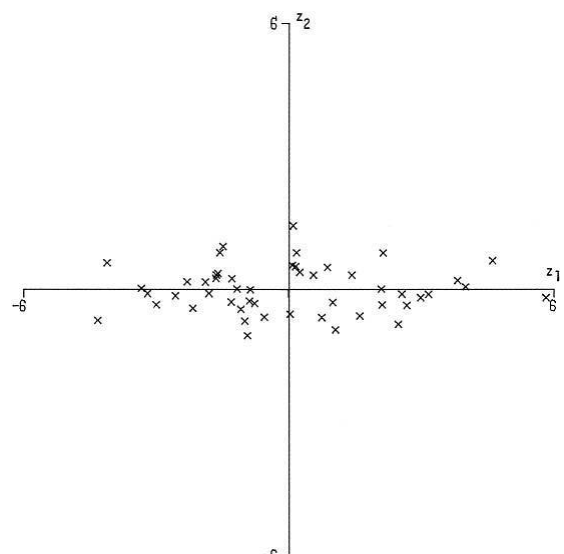


Fig 2: Datos referenciados a la primera componente principal

La forma de proceder en la realización de este método permite reducir una matriz de datos extraídos de la planta de  $m$  variables en  $n$  instantes de tiempo. Para ello, tras el procesamiento de los datos mediante las técnicas explicadas en la sección anterior, se buscará una función lineal donde los datos tengan máxima varianza.

Dicha función tendrá este aspecto:

$$\alpha_1 x_1' = \alpha_{11} x_1 + \alpha_{12} x_2 + \dots + \alpha_{1n} x_n \quad (\text{Ec. 5})$$

Donde:

- $n$  - Número de observaciones del bloque  $i$
- $x_i$  - componente  $i$  del autovector que determina la posición del nuevo sistema de referencia

- $x_i$  – Dato en la base original
- $\alpha_1$  – Parámetro que indica la influencia de la componente  $x_1$  en  $x_1'$

Posteriormente se busca otra función lineal  $\alpha_2' x_2$ , independiente de la anterior, que recoja la máxima variabilidad. Así se continuará obteniendo los datos proyectados en las distintas componentes principales.

La última función de este tipo que se pueda obtener será la función  $\alpha_n' x_n$ .

Matemáticamente se utilizan los conceptos de autovalor y autovector de la matriz de covarianza de los datos. Calculando el autovalor más grande de dicha matriz se obtendrá la primera componente principal. El segundo más grande corresponderá a la segunda componente principal y así sucesivamente.

## 2.4. Inteligencia Computacional: Redes neuronales, neurodifusas y toma de decisiones OWA

La **Inteligencia computacional** es una de las ramas en auge de la ingeniería y se puede definir como una “disciplina que busca dar claridad de acción ante situaciones de decisión confusas y que posee herramientas las cuales sirven para entender claramente la situación de decisión, extienden la memoria y la capacidad cognoscitiva y ayudan a superar la incertidumbre encontrando la mejor alternativa.” (El Hamzaoui, 2014)

Las principales ventajas que ofrece son la alta calidad de las decisiones que desprende, el incremento de la productividad, el ahorro de tiempo y costes que puede ofrecer y la mayor satisfacción conseguida en tanto en la plantilla de trabajadores como en el cliente final.

En cuanto a los tipos de sistemas basados en inteligencia computacional son los ya mencionados anteriormente (lógica difusa, redes neuronales, búsqueda causal, sistemas expertos, algoritmos genéticos,...). Puesto que este trabajo se implementará distintas redes neuronales y neurodifusas y se utilizarán operadores para toma de decisiones en procesos de detección y diagnóstico de fallos esta sección se centrará en dichos métodos.

### 2.4.1. Redes neurodifusas

Las redes neurodifusas son sistemas que reúnen tanto características propias de las redes neuronales, como su capacidad para generar patrones de comportamiento o de generalizar, con los postulados de la lógica difusa, basados en las funciones de membresía, que dan la posibilidad de trabajar con variables lingüísticas propias de los humanos. (Caamano, 2018)

Así su interés reside en (Villada, García, & Molina, 2011):

- Pueden aplicarse los mismos algoritmos de aprendizaje que en las redes neuronales.
- Son capaces de aportar integración del conocimiento.
- Pueden extraer conocimiento a partir de reglas de la muestra de datos.

El funcionamiento de este tipo de redes se basa en las neuronas difusas, que simulan morfología de una neurona. Se pueden encontrar dos configuraciones, la primera en la que el modelo responde a las entradas provenientes del motor de interferencia difusa.

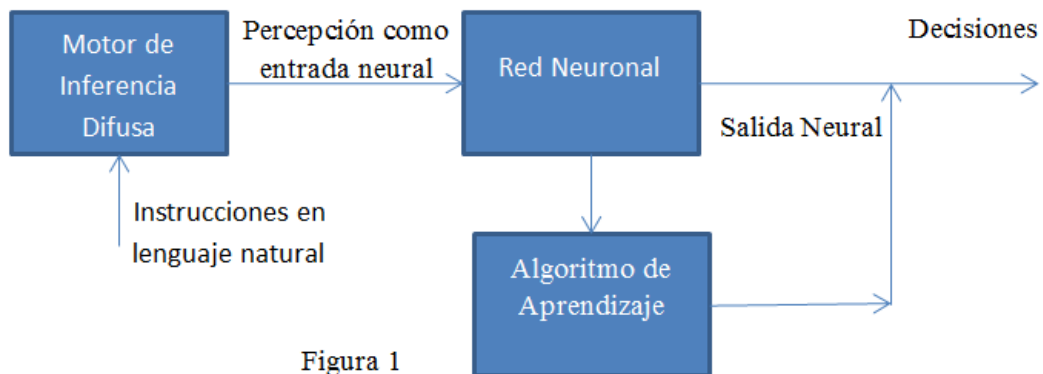


Fig 3: Estructura de red neurodifusa (1)

En la segunda configuración, las salidas provenientes de una red neuronal multicapa entran a un motor de interferencia difuso que se encarga de ajustar las funciones de membresía de la parte encargada de tomar decisiones, la correspondiente a la lógica difusa.

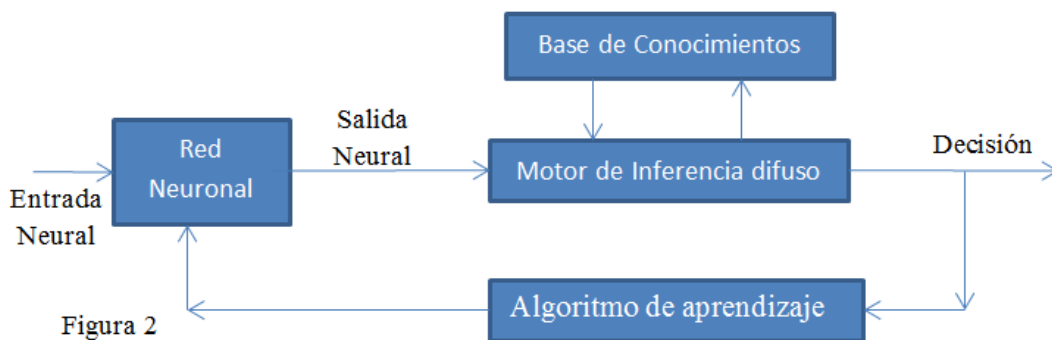


Fig 4: Estructura de red neurodifusa (2)

2.4.1.1. ANFIS

Una de las redes que van a ser utilizadas en este TFG es la denominada como ANFIS (sistema de inferencia difusa basado en red adaptativa). Es una red neuronal adaptativa basada en la implementación del modelo de inferencia difuso Takagi-Sugeno. (Correa-Henao & Montoya-Suárez, 2013)

Este tipo de redes están formadas por varios nodos, cada uno encargado de transformar la señal que recibe en una señal de salida del propio nodo. Cuenta con la ventaja de que permite la generación de modelos a partir de pares de datos de entrada/salida.

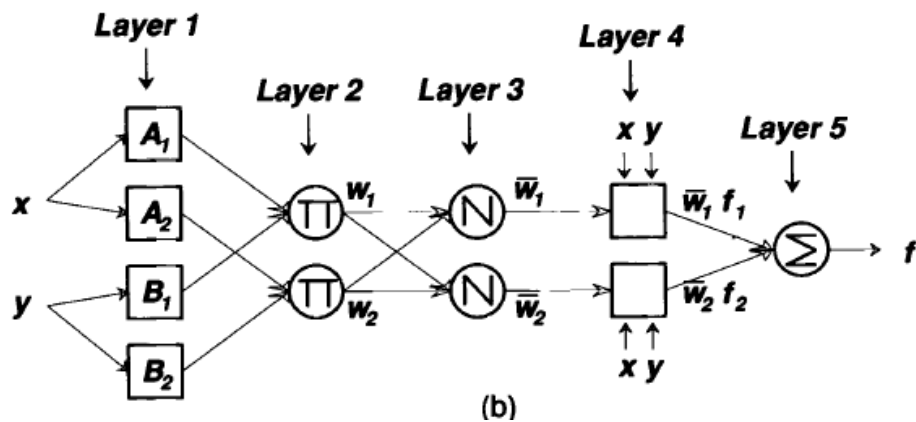


Fig 5: Estructura de una red neurodifusa ANFIS

En cuanto a su **estructura**, tal y como puede verse en la figura anterior cuenta con 5 capas que se describen a continuación. (Jyh-Shing Roger Jang, 1997)

**Capa 1:** Todos los nodos de esta primera capa cumplen una función de este tipo:

$$O_{1,i} = \mu_{A_i}(x) \text{ para } i=1,2 \tag{Ec. 6}$$

En esta función x e y representa los datos de entrada y  $O_{1,i}$  indica la calidad de membresía de un conjunto difuso A y especifica el grado en que la entrada dada x (o y) satisface al cuantificador A.

Una de las funciones de membresía utilizada para este tipo de red puede ser la función de campana generalizada, que cuenta con el conjunto de parámetros  $a_i$ ,  $b_i$  y  $c_i$  en función de los cuales la forma de esa campana varía obteniendo en consecuencia distintas funciones de membresía.

$$\mu_A = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b}} \tag{Ec. 7}$$

**Capa 2:** La segunda capa corresponde a un nodo el cuyo resultado es el producto de todas las señales de entrada que llegan a él. Por ejemplo:

$$O_{2,i} = w_i = \mu A_i(x) \mu B_i(y) \text{ para } i=1,2 \quad (\text{Ec. 8})$$

Cada una de las salidas de los nodos de la segunda capa representa la fuerza de disparo de alguna de las reglas.

**Capa 3:** En la tercera capa, cada uno de los nodos calcula la fracción entre la potencia de disparo de la regla  $i$  y la suma de todas las fuerzas de disparo. Las salidas de estos nodos reciben el nombre de fuerzas de disparo normalizadas.

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2} \quad (\text{Ec. 9})$$

**Capa 4:** Los nodos de la cuarta capa tendrán una salida del tipo:

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \quad (\text{Ec. 10})$$

en donde la potencia de disparo normalizada se multiplica por una función en la que  $q_i$ ,  $p_i$  y  $r_i$  son distintos parámetros del nodo. Estos parámetros se denominan parámetros consecuentes.

**Capa 5:** La última capa cuenta tan solo con un nodo, encargado de calcular el sumatorio de las señales entrantes:

$$O_{5,i} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (\text{Ec. 11})$$

## 2.4.2. Redes neuronales

El principal objetivo de las redes neuronales artificiales es la simulación del cómputo del cerebro humano en sistemas digitales. Las personas tenemos una capacidad para computar datos e ideas de una forma altamente compleja. Esta complejidad se basa en la organización de la información en neuronas que hacen posible tareas como el reconocimiento de patrones, modelado, etc.... de forma más rápida a la que lo puede conseguir un sistema digital de los disponibles en el mercado.

Esto se realiza mediante un procesador distribuido de forma masivamente paralela formada por procesamiento simple de unidades que tienen una propensión natural a almacenar conocimiento experiencial y hacerlo disponible para su uso. (Haykin, 2009)

Las redes neuronales requieren un proceso de aprendizaje en que adquirirán "experiencia" desde su entorno. Este proceso se denomina algoritmo de aprendizaje y en el que se modifican los pesos psinópticos, que regulan las intensidades de conexión entre las neuronas.

### 2.4.2.1. Clasificación de las redes neuronales

Existe una gran variedad de redes neuronales que pueden ser clasificadas ateniendo a distintos aspectos de las mismas. (Ballesteros, 2017)

Las redes pueden clasificarse **según su arquitectura**, es decir según se disponen sus neuronas. Estas se distribuyen dando origen a varias capas que a su vez se organizan para dar lugar a la red neuronal. Así existen dos tipos:

- **Red neuronal monocapa:** Son aquellas que tan solo cuentan con una capa de neuronas. En este grupo se incluyen las llamadas 'autorecurrentes', que son aquellas que tienen conexiones a sí mismas. Se utilizan mucho en circuitos eléctricos.

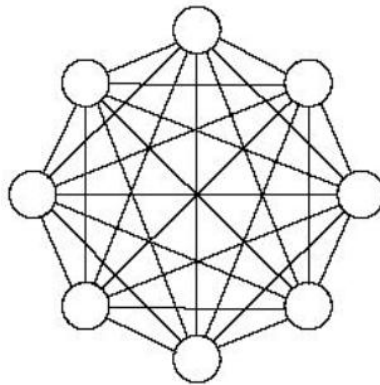


Fig 6: Ejemplo de red monocapa

- **Redes neuronales multicapa:** Se distinguen de las anteriores en que estas están formadas por varias capas de neuronas.

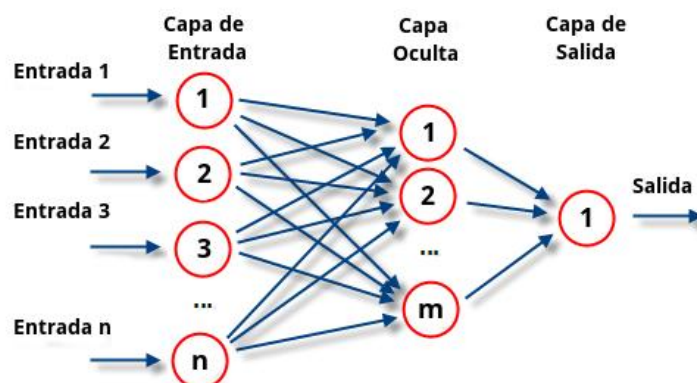


Fig 7: Ejemplo de red multicapa (Perceptrón)

Estas capas pueden distribuirse de dos formas, la primera en las señales de las capas intermedias siempre se dirigen hacia delante, hacia la salida y otra forma en la que algunas de las señales de la salida o de las capas intermedias son devueltas hacia atrás.

También pueden clasificarse atendiendo a cómo es su aprendizaje, distinguiéndose dos tipos:

- **Aprendizaje supervisado:** Para este tipo de aprendizaje, a la red se le presentan tanto los datos de entrada como la salida deseada. De esta forma los pesos psinopticos se van regulando hasta que la salida obtenida se parezca lo máximo posible a la deseada.

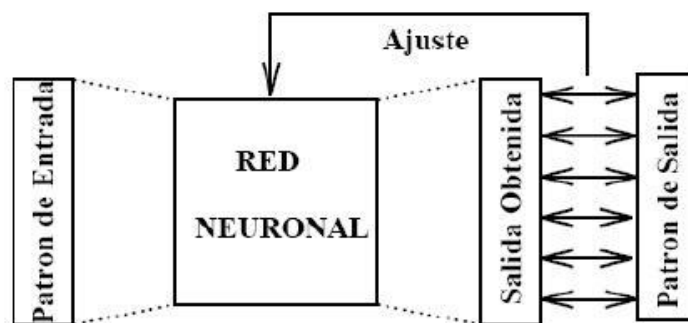


Fig 8: Aprendizaje supervisado

- **Aprendizaje no supervisado:** En su entrenamiento la red no cuenta con la salida deseada, tan solo con los datos de entrada. Utiliza la función de densidad de probabilidad de los datos para la extracción de patrones y reglas en el conjunto de los datos de entrada.

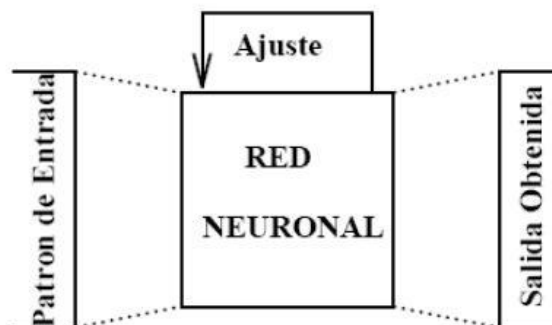


Fig 9: Aprendizaje no supervisado

### 2.4.2.1. Red neuronal de base radial (RBF)

Las Redes Neuronales radiales son sistemas formados por varias capas con conexiones hacia delante. En cuanto una de las neuronas ocultas se activan, las neuronas de la capa de salida reacciona con una respuesta combinación lineal de las ocultas. (Colmenares)

En aspectos generales, la RBF se considera una red altamente eficiente en su fase de entrenamiento y que responde muy bien ante un gran número de datos.

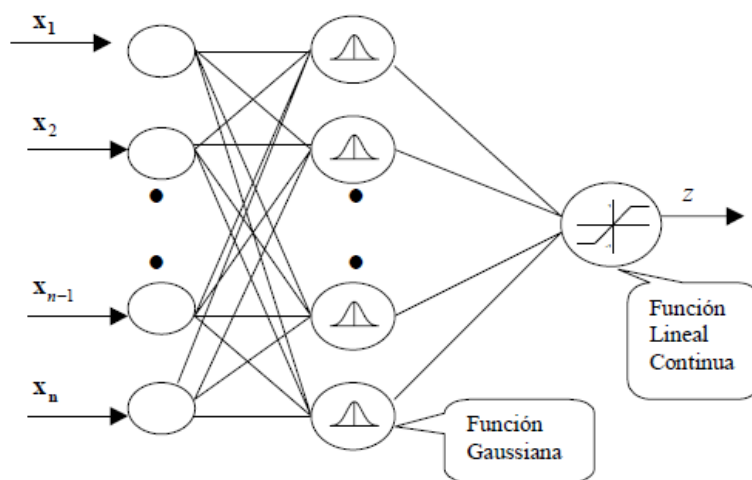


Fig 10: Estructura de una red neuronal de base radial (RBF)

Este tipo de Red consta de tres partes, una capa de entrada (encargada de la transmisión de señales a las neuronas oculta de la red), la capa oculta (cuya función es realizar una transformación radial de carácter no lineal de los datos recibidos) y por último una capa de salida (que realiza una combinación lineal de las activaciones de la capa oculta).

En cuanto a su topología es necesario señalar que cada una de las neuronas de la capa oculta se regula por dos parámetros: centro y ancho. Normalmente existe un centro distinto para cada uno de los nodos. En cuanto al ancho, señalar que se refiere al término que se emplea para identificar cual es la amplitud de la campana de Gauss originada por la transformación radial.

En primer lugar el nodo de la capa oculta se encarga de calcular cual es la distancia euclidiana \$d\$ entre el vector de entrada (con \$n\$ observaciones), a ese nodo en particular y al centro gravitatorio de ese nodo (\$c\$):

$$d = |x - c| = \sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2 + \dots + (x_n - c_n)^2} \quad (\text{Ec. 12})$$

Este es uno de los parámetros de la entrada para activar las neuronas ocultas. Así la función radial suele tener un aspecto del tipo \$G=\exp(-d^2)\$. Si también



empleáramos el parámetro ancho de la campana, como es nuestro caso la función tendría esta forma:  $G = \exp(-d^2/a)$ .

Una vez se han activado los nodos de la capa oculta, se realiza la transformación lineal hasta la capa de salida. Para ello a cada una de las neuronas ocultas le corresponderá un peso ( $w$ ) que multiplicado por la función de activación dará lugar a la función lineal continua de salida.

$$z = \sum w_i G \quad (\text{Ec. 13})$$

En cuanto al entrenamiento de este tipo de redes, su aprendizaje se basa en el cálculo de los centros, desviaciones y pesos de las capas ocultas.

En una primera fase el algoritmo de entrenamiento se encarga de calcular cuales son los centros y amplitudes de cada una de las neuronas de la capa oculta. Posteriormente, una vez que esas neuronas están definidas, el sistema calcula los pesos de las conexiones a la capa de salida y los umbrales de estas neuronas.

#### 2.4.2.2. Backpropagation

La última de las redes neuronales utilizadas en este trabajo es la red Perceptron basada en el algoritmo de backpropagation (hacia atrás).

Este es un algoritmo de entrenamiento supervisado. Su entrenamiento consta de dos fases. En la primera la red recibe los datos de entrada y los transmite hacia delante generando un valor de salida. Posteriormente este valor de salida es comparado con el valor esperado dando lugar a un error. En la segunda fase esta información se enviará hacia atrás provocando un reajuste del proceso con el objetivo de optimizarlo. Esta optimización consiste en la variación de los pesos de la función de activación para la corrección del error producido.

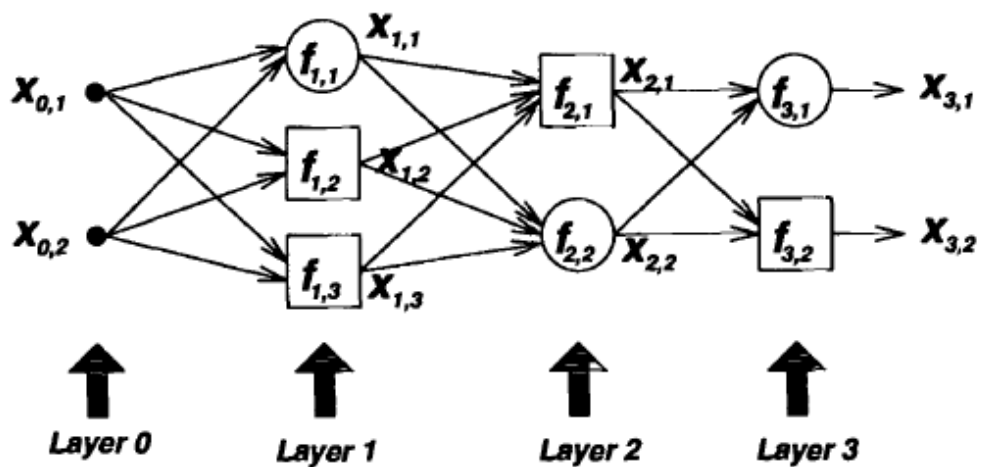


Fig 11: Estructura de una red Backpropagation

En cuanto a su estructura este tipo de red se compone por un total de tres capas. La primera, de entrada, una capa oculta intermedia y una capa de salida.

Su error de salida puede medirse utilizando el método del error cuadrático medio, que expresa la desviación elevada al cuadrado entre el valor estimado de la salida y el real. Si la entrada cuenta con P datos ese error se puede definir como

$$E_p = \frac{1}{2} \sum_{K=1}^{N(L)} (d_k - x_{L,k})^2 \quad (\text{Ec. 14})$$

donde  $d_k$  la componente k-ésima del vector de valores deseados  $p$  y  $X_{L,k}$  el valor de salida de la componente k-ésima con el vector de entrada  $p$ .

Después se procede a realizar el cálculo de las derivadas parciales del error con respecto a los pesos de que unen la ultima capa oculta con la de salida y a los que unen la capa de entrada con las primeras ocultas.

$$\Delta W = -\gamma \frac{\partial E_p}{\partial W} \quad (\text{Ec. 15})$$

Siendo  $\gamma$  un parámetro que se conoce como factor de aprendizaje.

Así se ajustarán los pesos en función de los resultados de estas derivadas. Para ello se recurre a algoritmos recursivos que comienzan por la capa de salida hasta la de entrada y van ajustando todos los pesos así:

Capa de salida:

$$W_{kj}^0(t+1) = W_{kj}^0(t) + \Delta W_{kj}^0(t+1) \quad (\text{Ec. 16})$$

Capa oculta:

$$W_{ji}^h(t+1) = W_{ji}^h(t) + \Delta W_{ji}^{h0}(t+1) \quad (\text{Ec. 17})$$

### 2.4.3. Toma de decisiones: Operadores OWA

Los operadores OWA, propuestos por Yager (Yager, 1996), buscan alcanzar una solución común a un problema en casos en los que existen varias alternativas diferentes. Además son considerados una de las principales familias de operadores de agregación ponderados en el campo de la toma de decisiones.

En un conjunto  $C = \{c_1, c_2, \dots, c_k\}$  donde  $c_i \in [0,1]$  un operador OWA de dimensión  $k$  **se define** como una función  $[0,1]^k \rightarrow [0,1]$  con un vector de pesos  $W = [w_1 w_2 w_3 \dots w_k]^T$  en el que  $w_i \in [0,1]$  y  $\sum_{i=1}^k w_i = 1$  (Palomares, Liu, Xu, & Luis, 2012)

$$OWA(c_1, c_2, \dots, c_k) = \sum_{n=1}^k w_n b_n \quad (\text{Ec. 18})$$

siendo  $b_n$  el  $n$ -ésimo mayor valor en el conjunto  $c$

Además estos operadores se pueden clasificar de acuerdo al concepto orness, que es una medida de optimismo que permite a partir del vector de pesos decidir cuan cercano se encuentra operador al operador mínimo (OR) o al máximo (AND).

Hay diversas formas de calcular los pesos de uno de estos operadores. Un ejemplo es a través de cuantificadores Regular Increasing Monotone (RIM).

Un cuantificador RIM  $Q$  se puede definir como un conjunto difuso en el intervalo  $[0,1]$  donde dada una proporción  $r$  (también el intervalo  $[0,1]$ ),  $Q(r)$  indica el grado en que  $r$  satisface la semántica definida en  $Q$ . (Palomares, Liu, Xu, & Luis, 2012)

Los cuantificadores de este tipo satisfacen las siguientes propiedades:

- $Q(0)=0$
- $Q(1)=1$
- Si  $r_1 > r_2$ ,  $Q(r_1) > Q(r_2)$

Si hablamos de un cuantificador RIM lineal se utilizan los parámetros  $\alpha$  y  $\beta$ , ambos existentes en el intervalo  $[0,1]$ . Así, su función de pertenencia se define como:

$$Q(r) = \begin{cases} 0 & \text{si } r < \alpha \\ \frac{r-\alpha}{\beta-\alpha} & \text{si } \alpha < r < \beta \\ 1 & \text{si } r > \beta \end{cases} \quad (\text{Ec. 19})$$

Para el cálculo de los pesos  $w_n$  se utilizará el siguiente método mediante:

$$w_i = \left(\frac{i}{k}\right)^{\frac{1-orness}{orness}} - \left(\frac{i-1}{k}\right)^{\frac{1-orness}{orness}} \quad (\text{Ec. 20})$$



## Capítulo 3

### Técnicas de IC & ML: Una propuesta

En este nuevo capítulo se propone una metodología basada en las técnicas de inteligencia computacional explicadas en el capítulo anterior. En ella se describen cuales son los principales pasos a seguir para la para la detección y diagnostico de fallos en una planta industrial.

En la siguiente página se encuentra un diagrama de flujo que muestra de una forma resumida esta metodología propuesta para la resolución de un problema como el que nos ocupa.

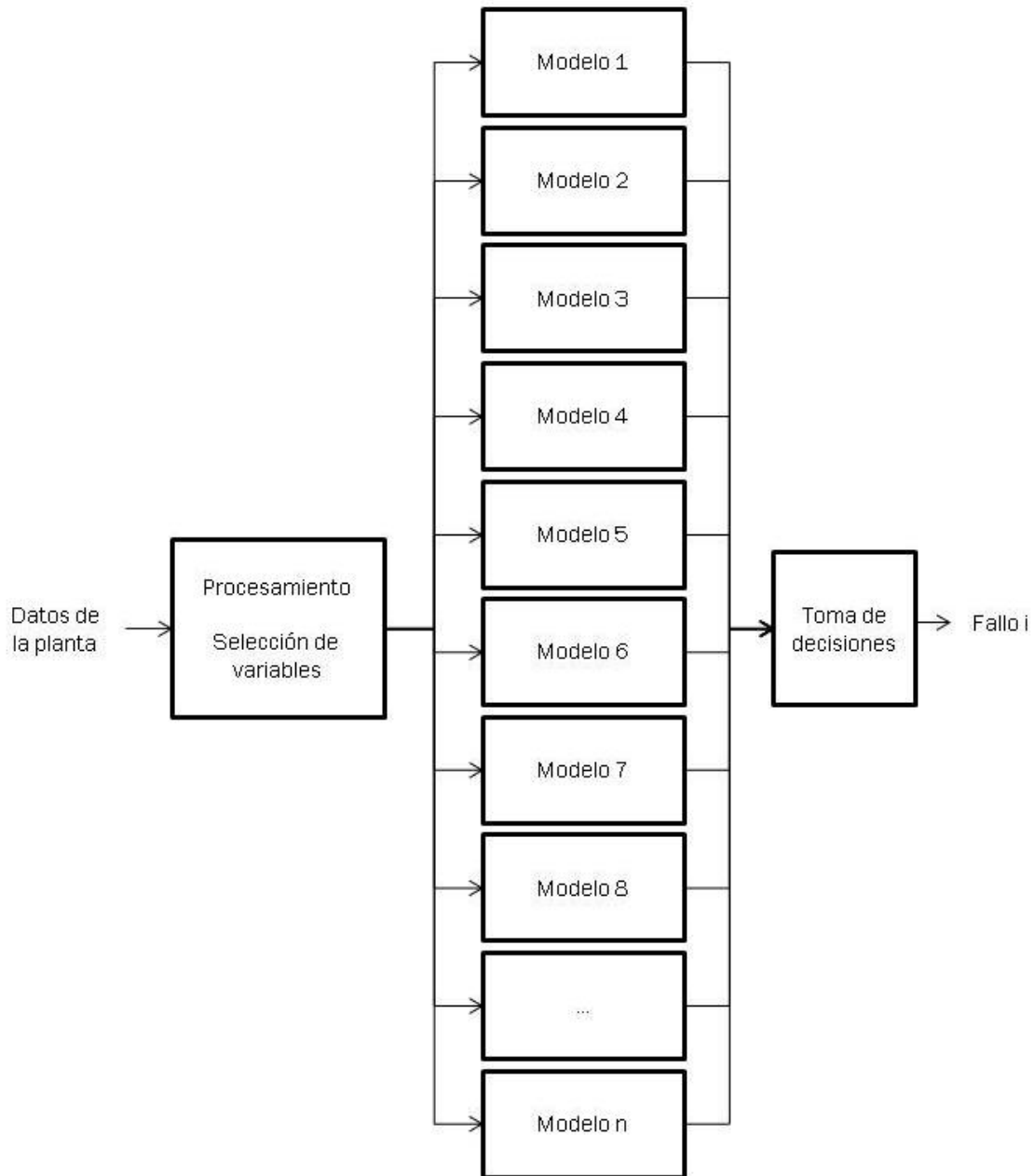


Fig 12: Diagrama de flujo de la metodología propuesta

El principal objetivo que queremos conseguir con nuestra metodología es alcanzar un sistema de  $n$  modelos que, en conjunto, sea capaz de detectar un fallo e identificar que fallo es.

Para comenzar, el conjunto de los datos se somete a diversas tareas de procesamiento para poder ser utilizados en posteriores etapas.

Más tarde se procederá a realizar una selección de variables. Para ello se comenzará utilizando el análisis de componentes principales (PCA) y con los

resultados que aporta intentaremos calcular cual es la variabilidad de cada una de las variables, creando con ayuda de OWA dos tipos de rankings de variables: uno utilizado para lo que llamaremos modelos globales y otros para modelos parciales.

El siguiente paso será búsqueda los  $n$  modelos más adecuados para nuestro problema, aquellos que sean capaces de detectar un fallo de manera clara y no lo confundan con lo demás. Para ello se van a llevar a cabo distintos experimentos realizados con redes neuronales y neurodifusas. Así, se entrenarán diversas redes con el fin de encontrar el mejor modelo global para cada red y el mejor modelo parcial para cada red y tipo de fallo (o modo de funcionamiento).

Una vez encontrados esos modelos se crearán diversos sistemas conformados por combinaciones de los mismos para estudiar cual es su tasa de éxito y decidir cuál es nuestro resultado final.

### **3.1. Etapa 1: Procesamiento de los datos**

En primer lugar el conjunto de datos se someterá a diversas técnicas de procesamiento con el fin de poder ser utilizados en etapas posteriores. Las principales tareas a realizar en dicha fase son las descritas anteriormente: la transformación, limpieza, integración y reducción de los datos.

---

#### **M. propuesta 1: Procesamiento de datos**

---

```
1:  for datos fallo 1:n do
2:    Limpieza de los datos
3:    Integración de los datos
4:    Transformación de los datos
5:    Reducción de los datos
6:  end for
```

---

Todas estas técnicas han sido explicadas en el capítulo 2 de aspectos teóricos.

### **3.2. Etapa 2: Selección de variables**

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

Una vez los datos han sido sometidos a la fase de procesamiento, se aplicará el análisis de componentes principales (PCA) con el fin de reducir la dimensionalidad de la muestra. Tal y como se ha contado en capítulos

anteriores, con ello conseguiremos la reducción del costo computacional de los equipos encargados del estudio de los datos y la mejor calidad de los resultados de minería de datos por la disminución del ruido en la muestra.

Esta técnica puede resultar útil para nuestro caso de estudio puesto que contamos con un gran número de variables interrelacionadas entre sí. El análisis de componentes principales podría ayudarnos a reducir el número de variables de los datos perdiendo una variabilidad relativa.

Al realizar el análisis de componentes principales (explicado en el capítulo 2 de aspectos teóricos) se obtiene una matriz de cargas  $\alpha_{ij}$  que representan el peso de cada variable  $i$  en cada componente principal  $j$ .

Esta matriz está conformada por los valores de los  $\alpha_{ij}$  que se indicaban en las transformaciones vistas en los conceptos teóricos.

$$\alpha_1 x'_1 = \alpha_{11} x_1 + \alpha_{12} x_2 + \dots + \alpha_{1p} x_p \quad (\text{Ec. 21})$$

Además se obtienen los autovalores y autovectores de la matriz de covarianzas que representan cada una de las componentes principales y los valores de las variables  $X_1, X_2, \dots, X_p$ . Después esos autovalores, que corresponden a las distintas componentes principales, se ordenan para futuras etapas.

---

#### **M. propuesta 2:** Análisis de componentes principales

---

```
1: for datos fallo 1:n do
2:   (Autovalores, Autovectores, Cargas)=pca(matriz covarianzas de los datos)
3:   Ordenar(Autovalores)
4: end for
```

---

### **3.2.2. Selección de variables: Reducción de la dimensionalidad**

Una vez calculadas las componentes principales se procede seleccionar cuales son las que mayor variabilidad de información contienen. Una vez obtenida la variabilidad que contiene cada una de ellas, decidiremos cuales conservar y cuáles no, intentando buscar un equilibrio entre el número de componentes principales seleccionadas y la variabilidad de la información contengan esas componentes.

Las componentes con mayor variabilidad son las que tienen un autovalor más alto, ese es el principal motivo por el que se han ordenado en la etapa anterior. Para poder medir cuanta variabilidad aporta una componente principal  $i$  se calculará el cociente entre su autovalor y la suma de autovalores del conjunto. (Jolliffe, 1986)



$$\text{Variabilidad componente principal } i = \frac{\lambda_i}{\sum \lambda_i} \quad (\text{Ec. 22})$$

Una vez hecho eso se seleccionarán aquellas que en conjunto representen un porcentaje lo suficientemente alto de variabilidad pero que permitan reducir la dimensionalidad de la muestra.

Para ello se ordenaran las variables de mayor a menor según la variabilidad conservada por cada una de ellas. Después se seleccionaran aquellas cuya variabilidad en conjunto sumen distintos porcentajes de variabilidad prefijados de antemano.

Habiendo obtenido el número de componentes principales necesarias para conservar cada uno de los porcentajes de variabilidad se buscará un equilibrio entre ambos aspectos.

---

#### **M. propuesta 3:** Variabilidad para i PC

---

```
1:  for autovalores do
2:    for porcentajes de variabilidad do
3:      i=0                               //Nº de componente principales
4:      variabilidad_conservada=0
5:      while variabilidad_conservada<porcentaje de variabilidad do
6:        i= i+1
7:        variabilidad_conservada= variabilidad_conservada+  $\lambda_i / \sum \lambda_i$ 
8:      end while
9:    end for
10: end for
```

---

### **3.2.3. Selección de variables: Relevancia de las variables**

El siguiente paso del proceso será seleccionar las variables con mayor relevancia en la muestra para decidir con cuales se trabajará en los modelos.

Como se ha explicado anteriormente, al aplicar el análisis de componentes principales se obtiene una matriz de cargas que implican el peso de cada una de las variables en las componentes principales.

Partiendo de esa matriz podemos evaluar en porcentaje la 'relevancia' que representa una variable frente al total.

Para conseguir este porcentaje se realizará el cociente entre el sumatorio de las cargas correspondientes a la variable i (en todas las componentes principales) y el sumatorio de todas las cargas.

$$R_i = \frac{\sum_{j=1}^n \alpha_{ij}}{\sum_{k=1}^n (\sum_{j=1}^n \alpha_{kj})} \quad (\text{Ec. 23})$$

---

**M. propuesta 4:** Relevancia de cada variable

---

```
1:  for fallo 1:n do
2:    for variable=i:n do
3:      for componente principal=j:n do
4:        Relevancia de una variable i  $R_i = \frac{\sum_{j=1}^n \alpha_{ij}}{\sum_{k=1}^n \sum_{j=1}^n \alpha_{kj}}$ 
5:      end for
6:    end for
7:  end for
```

---

Así obtendremos para cada uno de los modos de funcionamiento, cual es la relevancia de cada una de las variables respecto al conjunto. Es decir, para el modo de funcionamiento (fallo) 1 sabremos la relevancia de la cada variable respecto al total. Lo mismo para cada modo de funcionamiento.

### 3.2.4. Ranking de variables. Método OWA.

Para continuar con la resolución del problema se propone la utilización de operadores OWA para la elaboración de dos tipos de rankings:

- Ranking global de las variables con mayor relevancia teniendo en cuenta todos los modelos a la vez.
- *Ranking de las variables con mayor relevancia para cada modo de funcionamiento.*

---

**M. propuesta 5:** Metodología propuesta para la elaboración de rankings

---

```
Ranking para usar en modelos globales
1:  for fallo i:n do
2:    Ranking global=OWA(var. entrenamiento fallo i:n, var. test fallo i:n)
3:  end for
Ranking para usar en modelos parciales
4:  for fallo i:n do
5:    Ranking fallo i=OWA(var. entrenamiento fallo i, var. test fallo i)
6:  end for
```

---

Con estos rankings se utilizarán para realizar dos tipos de experimentos, unos basados en modelos globales (utilizando todos los datos a la vez) y unos

basados en modelos parciales (utilizando los datos de cada fallo por separado). Todos estos experimentos se realizan con el objetivo de crear un sistema capaz de detectar los fallos de la mejor forma posible.

### **3.3. Etapa 3: Modelado basados en Redes Neuronales**

Después se procederá a experimentar con varios modelos para encontrar con ellos cual es más eficiente (y bajo qué parámetros) para cada modo de funcionamiento.

Cómo se ha comentado en la sección anterior, se generarán dos tipos de modelos:

- 1. Modelos globales:** Cada una de las redes se entrena bajo distintas combinaciones de parámetros pero siempre con el conjunto de datos de entrenamiento (habiendo seleccionado el número de variables elegidas del ranking global). Así mismo se realizará una etapa de testeo con el conjunto de datos de test.
- 2. Modelos parciales:** También se entrenarán distintos modelos parciales. Para cada una de las tres redes se tendrán tantos modelos como número de modos de funcionamiento existan en la planta. Los modelos correspondientes a cada fallo se entrenarán con sus datos de entrenamiento bajo distintas combinaciones de parámetros (habiendo seleccionado las variables que copan el ranking parcial referido a su modo de funcionamiento). También se realizará un testeo de cada modelo con los datos de testeo correspondientes a su fallo.

Cada uno de los entrenamientos y test arrojarán un error de cada una de las dos etapas. Así se seleccionarán como mejores modelos globales para cada red aquellos que menor valor de esos errores hayan conseguido, y se hará lo mismo para obtener los mejores modelos parciales para cada red y cada fallo.

Existen muchos tipos de redes neuronales y neurodifusas, pero en este trabajo se utilizarán la red neuronal de base radial (RBF), sistemas adaptativos de interferencia neurodifusa (ANFIS) y la red conocida como Backpropagation.

Las salidas esperadas que se aportan al sistema para todos los modelos corresponden al número de fallo de los datos con los que se entrena.

**M. propuesta 6:** Experimentación

---

```
Modelos globales
1:  for fallo i=1:n do
2:    (error_entrenamiento, error_test) = Anfis(datos fallo i)
3:    Selección del mejor modelo anfis = min(error_entrenamiento, error_test)
4:    (error_entrenamiento, error_test) = rbf(datos fallo i)
5:    Selección del mejor modelo rbf = min(error_entrenamiento, error_test)
6:    (error_entrenamiento, error_test) = backpropagation (datos fallo i)
7:    Selección del mejor modelo backpropagation= min(error_entrenamiento,
error_test)
8:  end for
Modelos parciales
9:  for fallo i=1:n do
10: (error_entrenamiento, error_test) = Anfis(datos fallo i)
11: Selección del mejor modelo anfis = min(error_entrenamiento, error_test)
12: (error_entrenamiento, error_test) = rbf(datos fallo i)
13: Selección del mejor modelo rbf=min(error_entrenamiento, error_test)
14: (error_entrenamiento, error_test) = backpropagation (datos fallo i)
15: Selección del mejor modelo backpropagation = min(error_entrenamiento,
error_test)
16: end for
```

---

Una vez se han seleccionado los mejores modelos (aquellos con menores valores de error) se realizará un nuevo testeo con el fin de verificar la validez de los mismos.

Aquí cada uno de los modelos se testeará con el conjunto de los datos y los resultados se reflejarán en distintas matrices:

- 1. Modelos globales:** La matriz expresará en porcentaje la cantidad de datos del fallo i que el modelo detecta como cada caso j.
- 2. Modelos parciales:** Reflejarán el error de test obtenido cuando el modelo correspondiente al caso i se testea con el caso j.

**M. propuesta 7:** Testeo de los modelos seleccionados

---

```
Modelos globales
1:  for fallo i=1:n do
2:      Inicializar k
3:      for fallo j do
4:          for Modelo={Mejor_Anfis, Mejor_RBF, Mejor_BP} do
5:              If (abs(modelo(datos fallo j)-i))<k
6:                  k= abs(modelo(datos fallo j)-i)           //Se guarda el n° de
                                                            fallo detectado
7:              end if
8:          end for
9:          Matriz_confusion_modelo(i,j)=                    //Suma el % que
                                                            supone detectar una
                                                            vez un fallo
                =Matriz_confusion_modelo(i,j)+(0.125)
10:     end for
11: end for
Modelos parciales
12: for fallo i=1:n do
13:     for fallo j do
14:         for Modelo={Mejor_Anfis_Fallo_i,                //Modelos entrenados
                                                            con fallo i
                Mejor_RBF_Fallo_i, Mejor_BP_Fallo_i} do
15:             Error_test=modelo(fallo j)                  //Testeo con fallo j
16:             Matriz_confusion_modelo(i,j)=Error_Test
17:         end for
18:     end for
19: end for
```

---

### 3.4. Etapa 4: Toma de decisiones

Después de haber seleccionado los mejores modelos de cada red, estos se agruparan en 4 configuraciones diferentes que conformarán 4 sistemas que permitirán la detección de fallos. Estos sistemas contarán con una configuración como la presentada en la siguiente imagen:

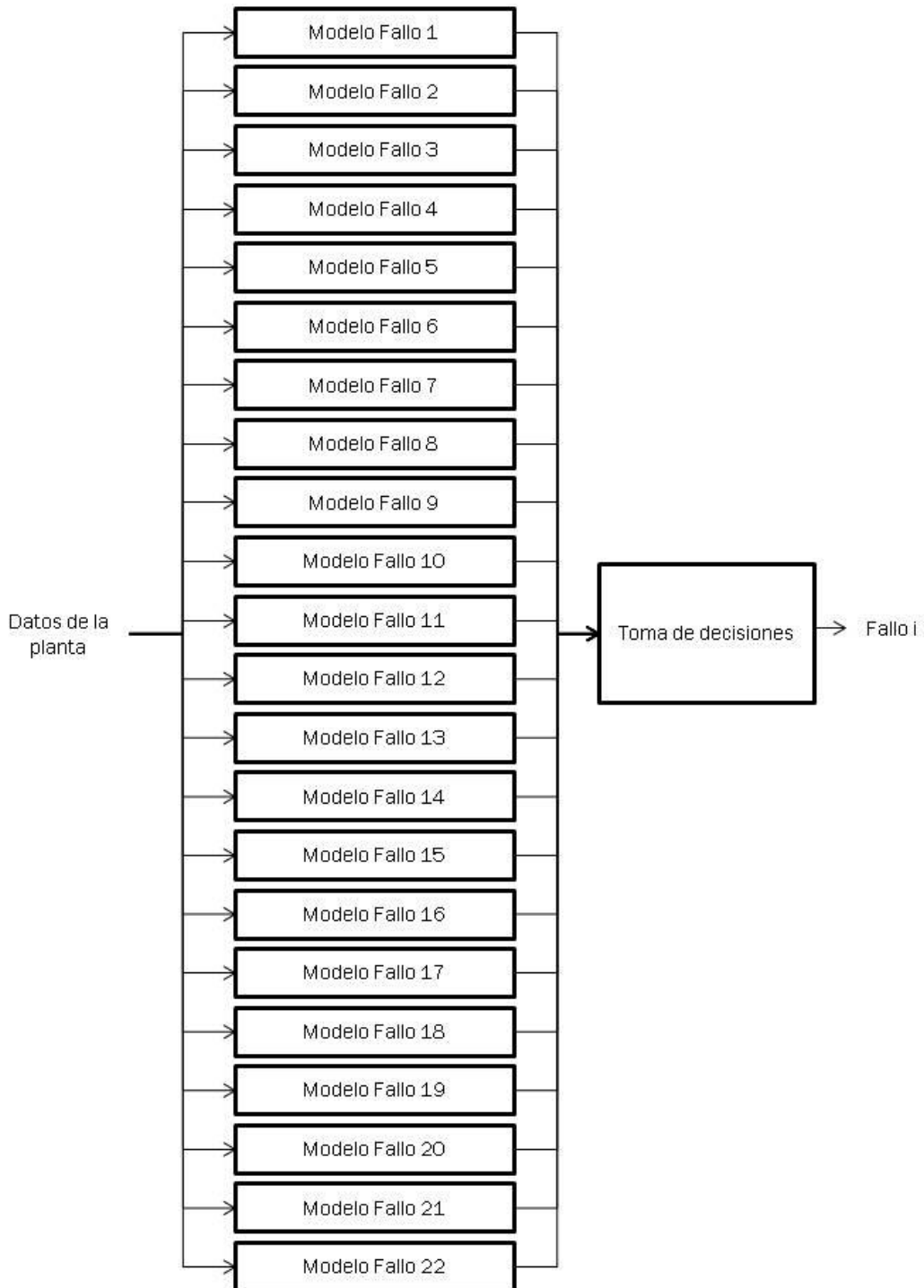


Fig 13: Esquema de modelos para la detección de fallos

Cada uno de esos sistemas contará con 22 modelos a los que se les introducirán los datos de salida. Cada uno de ellos dará una respuesta al comportamiento de la planta y con el conjunto de las 22 respuestas se realizará una toma de decisiones en la que se aclare cuál es el fallo o modo de funcionamiento en el que se encuentra la planta.

En el primer conjunto se utilizarán solo modelos parciales entrenados con Anfis, aquellos que se habían considerado como mejor resultados encontrados para cada uno de los fallos.

El segundo será igual que el anterior pero los 22 modelos se corresponderán a los mejores modelos parciales entrenados con RBF.

El tercero con Backpropagation y en el último se seleccionarán los mejores modelos de entre las tres redes.

Por último la cuarta configuración, que será la finalmente elegida, contará con una combinación de los modelos basados en las tres redes. Se seleccionará para cada tipo de fallo la red que menor resultado de error de test haya aportado cuando se enfrenta a los datos del mismo fallo con el que ha sido entrenado y mayor con los demás, es decir el modelo de cualquiera de las tres redes que mejor detecte cada fallo.

Como se ha dicho el conjunto de datos volverá a ser testeado por los 22 modelos de cada sistema. Cada uno de esos 22 modelos aportará un resultado de error de test. En la toma de decisiones, se elegirá como funcionamiento existente en la planta al modo de funcionamiento o fallo que menor error de test aporte.

Así, se medirá cual es el porcentaje de éxito de cada uno de los cuatro sistemas.





## Capítulo 4

### Trabajo experimental

En el cuarto capítulo de esta memoria se va a describir el benchmark que se va a emplear para testear la propuesta realizada en el capítulo anterior. Se describirán como se ha llevado a cabo esta tarea y la metodología empleada.

#### 4.1. Caso de estudio: Proceso Tennessee Eastman (TEP)

La planta que a emplear es la correspondiente al Proceso Tennessee Eastman (TEP)

Este proceso puede verse esquematizado en la siguiente figura:

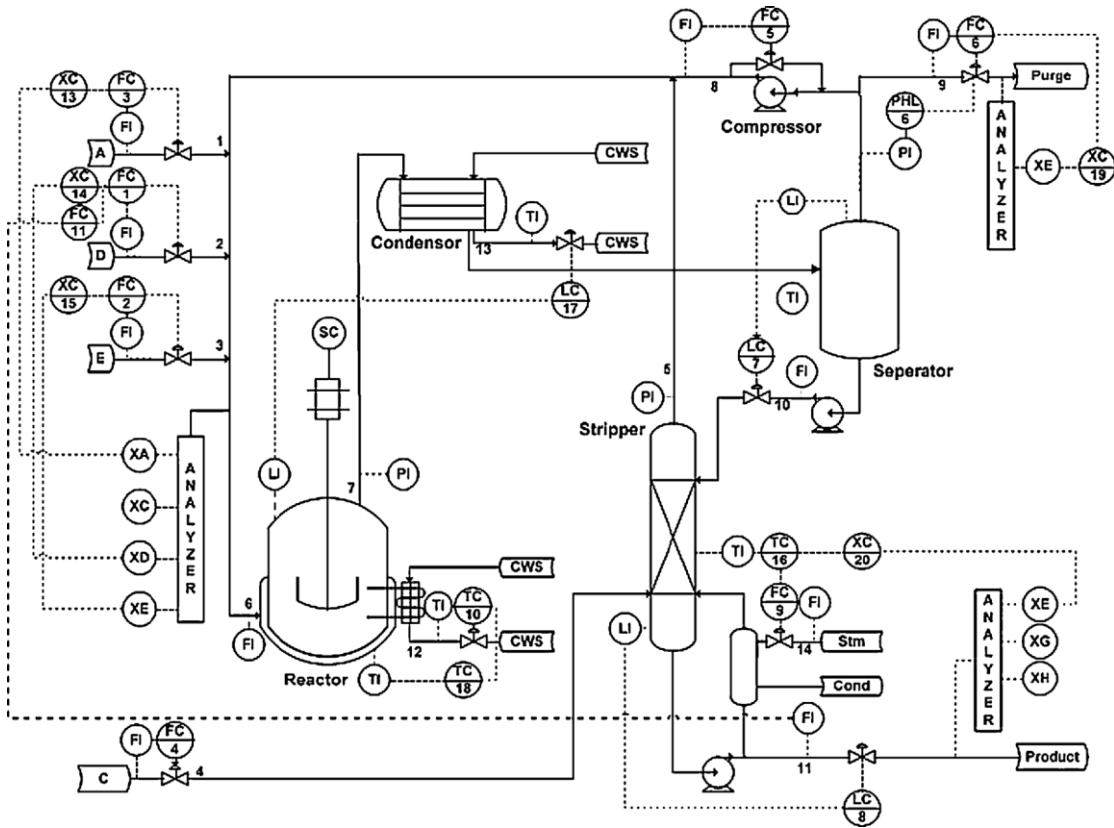
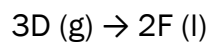
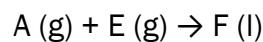
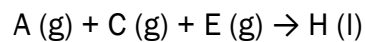
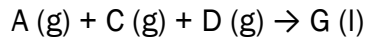


Fig 14: Esquema del proceso Tennessee Eastman

El proceso TEP fue creado específicamente con el fin de ofrecer un proceso realista que permita evaluar el control y monitoreo de procesos industriales. Así simula un proceso realista que contiene ocho componentes A, B, C, D, E, F, G y H y que está sometido a diversas condiciones de operación que podrían variar su funcionamiento. (Russell, Chiang, & Braatz, 2000)

El proceso cuenta con cinco unidades claramente definidas: reactor, condensador, separador flash, separador stripper y un condensador del producto. El reactor se alimenta con 4 reactivos gaseosos (A, C, D y E) y un inerte (B) donde se forman los productos G y H en fase líquida. También se forma un subproducto F también en fase líquida.

Las reacciones que tienen lugar en la planta son:



En el Proceso Tennessee Eastman se ponen en juego un total de 52 variables:

Variables de la planta	
1	Flujo de alimentación del reactante D (Punto 2 de la Fig 14: Esquema del
2	Flujo de alimentación del reactante E (Punto 3 de la Fig 14: Esquema del
3	Flujo de alimentación del reactante A (Punto 1 de la Fig 14: Esquema del
4	Flujo de alimentación del reactante C (Punto 4 de la Fig 14: Esquema del
5	Válvula de reciclaje del compresor
6	Válvula de purga (Punto 9 de la Fig 14: Esquema del proceso Tennessee
7	Flujo de líquido del separador (Punto 10 de la Fig 14: Esquema del
8	Flujo de producto líquido de stripper (Punto 11 de la Fig 14: Esquema del
9	Válvula de vapor del separador
10	Flujo de agua de refrigeración del reactor
11	Velocidad del agitador
12	Flujo reciclado en el punto 8 de la Fig 14: Esquema del proceso
13	Velocidad de alimentación del reactor (Punto 6 de la Fig 14: Esquema
14	Presión del reactor
15	Nivel del reactor (%)
16	Temperatura del reactor
17	Tasa de purga (Punto 9 de la Fig 14: Esquema del proceso Tennessee
18	Temperatura del producto separado
19	Presión del producto separado
20	Nivel del producto separado (%)
21	Flujo de los productos del separador en el punto 10 de la Fig 14:
22	Nivel del separador stripper
23	Presión del separador stripper
24	Flujo de los productos del separador stripper en el punto 10 de la Fig 14:
25	Temperatura del separador stripper
26	Flujo de vapor del separador stripper
27	Trabajo del compresor
28	Temperatura de salida del agua de refrigeración del reactor
29	Temperatura de salida del agua de refrigeración del separador
30	Porcentaje molar del componente A en el punto 6
31	Porcentaje molar del componente B en el punto 6
32	Porcentaje molar del componente C en el punto 6
33	Porcentaje molar del componente D en el punto 6
34	Porcentaje molar del componente E en el punto 6
35	Porcentaje molar del componente F en el punto 6
36	Porcentaje molar del componente A en el punto 9
37	Porcentaje molar del componente B en el punto 9
38	Porcentaje molar del componente C en el punto 9
39	Porcentaje molar del componente D en el punto 9
40	Porcentaje molar del componente E en el punto 9
41	Porcentaje molar del componente F en el punto 9
42	Porcentaje molar del componente G en el punto 9
43	Porcentaje molar del componente H en el punto 9
44	Porcentaje molar del componente D en el punto 11
45	Porcentaje molar del componente E en el punto 11
46	Porcentaje molar del componente F en el punto 11
47	Porcentaje molar del componente G en el punto 11
48	Porcentaje molar del componente H en el punto 11
49	Desconocida
50	Desconocida
51	Desconocida
52	Desconocida

Tabla 1: Variables de la planta

Por otra parte, los principales fallos que pueden presentarse en la planta son:

Principales fallos de la planta	
1	Variaciones en la relación de alimentación de A / C con B constante
2	Variación de B con relación A/C constante.
3	Salto en la temperatura de entrada del reactivo D
4	Variaciones bruscas de temperatura de entrada de agua de refrigeración del reactor
5	Variaciones bruscas de la temperatura de entrada del agua de refrigeración del condensador
6	Una pérdida de alimentación
7	Perdida de presión de C en el punto 4 de la figura.
8	Variaciones aleatorias de los flujos de entrada de A, B y C
9	Variación aleatoria de la temperatura D (punto 2 de la figura)
10	Variación aleatoria de la temperatura de C (punto 4 de la figura)
11	Variación aleatoria de la temperatura del flujo de entrada de agua de refrigeración al reactor
12	Variación aleatoria de la temperatura del flujo de entrada de agua de refrigeración al condensador
13	Disminución de la cinética de reacción
14	Problemas con la válvula de agua de enfriamiento del reactor
15	Problemas con la válvula de agua de refrigeración del condensador
16	Fallo desconocido
17	Fallo desconocido
18	Fallo desconocido
19	Fallo desconocido
20	Fallo desconocido
21	Fallo desconocido

*Tabla 2: Fallos de la planta*

## 4.2. Metodología experimental

A continuación se va a describir la metodología experimental seguida para comprobar la propuesta de detección de fallos descrita en el capítulo anterior de este TFG.

### 4.2.1. Datos de partida

Los datos a estudiar se recogen en un total de 42 ficheros de dos tipos:

- **Ficheros de entrenamiento:** Reflejan la evolución de 52 variables de nuestra planta a lo largo de 480 instantes de tiempo. Se denominan de la siguiente forma dXX.dat donde XX corresponde a la numeración del fichero.
- **Ficheros de test:** Contienen datos de la evolución de las mismas 52 variables a lo largo de 960 instantes de tiempo. Se considera, en base a lo observado en dichos datos que desde el instante 0 hasta el instante 160 reflejan el funcionamiento nominal de la planta. Se denominan de una forma parecida a los anteriores, siguiendo el patrón dXX\_te.dat donde XX corresponde a la numeración del fichero.

Para cada uno de los 22 casos o modos de funcionamiento existirá un fichero de cada tipo.

El modo de funcionamiento número 22 corresponde al modo de funcionamiento en el que no existe ningún fallo en la planta.

#### **4.2.2. Metodología utilizada**

Para nuestro caso de estudio se realiza en primer lugar el procesamiento de los datos. Este procesamiento consistirá en la detección de valores que difieran mucho de la muestra (Outliers), la búsqueda de posibles missing values, y una normalización del conjunto mediante zscore, cómo se ha comentado en el Cap.2.2. Procesamiento de los datos

---

##### **M. experimental 1: Procesamiento de datos**

---

```
1: for fallo 1:22 do
2:   Detección de Otlers
3:   Missing Values
4:   Normalización zscore
5: end for
```

---

Posteriormente se aplicará un análisis de componentes principales a cada uno de los modos de funcionamiento. Para ello el sistema calcula la matriz de covarianzas de los datos de cada modo de funcionamiento y sus autovalores y autovectores.

Cómo se ha explicado anteriormente obtendremos una matriz de autovectores, que corresponden a cada una de las componentes principales, una matriz de autovectores que indica cuales son los valores en el sistema original, y una matriz de cargas que indica el factor con el que cada variable participa en cada una de las componentes principales.

---

##### **M. experimental 2: Metodología experimental para el PCA**

---

```
   Análisis de componentes principales:
1: for fallo 1:22 do
2:   (Autovalores, Autovectores, Cargas)=pca(matriz covarianzas de los datos)
3: end for
```

---

Estos resultados se utilizarán para ayudar a la comprensión de todas las relaciones que puedan existir entre las distintas variables e intentar disminuir lo que se pueda algunos datos que puedan ser redundantes.

Tal y como se ha descrito antes cada una de las componentes principales representa un porcentaje de variabilidad de la información. Por ello se buscará un equilibrio entre el número de componentes principales y la variabilidad de información conservada. Se va a estudiar cuantas componentes principales deberíamos tener para que nuestros datos cuenten con un 70%, 80% y 90% de la información.

---

**M. experimental 3:** nº de PC según la variabilidad

---

```
1: for fallo i=1:22 do
2:   num_de_variables=0 //Inicializar el contador para cada caso
3:   while variabilidad<{70,80,90}
4:     for autovalor i do
5:       num_de_variables= num_de_variables+1
6:       variabilidad = variabilidad +(λi/ Σλi)
7:     end for
8:   end while
9: end for
```

---

Analizando los resultados obtenidos se considera que con el número de componentes principales para conseguir mantener un 70% de la varianza se puede alcanzar un nivel aceptable de representación de la totalidad de la muestra. En el Cap.4.3 de resultados se puede observar el número de componentes necesarias para alcanzar ese 70%.

Una vez se tienen las componentes principales de cada modo, se estudiará la relevancia de cada una de las variables según el ‘peso’ que tenga cada una de ellas en el conjunto.

En el caso de nuestra planta se cuenta con un total de 52 variables, que como se ha dicho puede resultar un número muy elevado que podría convertir nuestro estudio en un estudio poco preciso. Esto motiva que el siguiente paso de nuestra metodología sea precisamente la selección de variables.

Para realizarla nos basamos en las componentes principales y sus parámetros obtenidos mediante PCA. Al realizar el análisis de componentes principales obtenemos una matriz de cargas o pesos. Estos parámetros corresponden al  $\alpha_{11}, \alpha_{12}, \dots, \alpha_{ij}$  de la expresión:

$$\alpha_1 x_1' = \alpha_{11} x_1 + \alpha_{12} x_2 + \dots + \alpha_{1p} x_p \quad (\text{Ec. 24})$$

Por ejemplo el parámetro  $\alpha_{11}$  indica el porcentaje de ‘importancia’ de la variable 1 en la componente principal 1.

Para realizar este procedimiento se estudiará caso a caso cual es la suma del peso de cada una de las variables en todas las componentes principales. Por ejemplo, para la variable uno se sumaran  $\alpha_{11}, \alpha_{12}, \alpha_{13}, \dots, \alpha_{1n}$  siendo n el

número de variables determinadas para este caso conservando un 70% variabilidad de la información.

Posteriormente se calculará, para ese caso, cual es el la 'importancia' de esa variable respecto a las demás. Para el ejemplo anterior esa 'importancia la calcularíamos como:

$$\frac{\sum_{i=1}^n \alpha_{1i}}{\sum_{j=1}^{52} (\sum_{i=1}^n \alpha_{ji})} \quad (\text{Ec. 25})$$

Así, obtendríamos para cada caso la relevancia de cada una de las variables dentro del conjunto. Estas tablas se pueden ver en los anexos del proyecto.

---

#### M. experimental 4: Relevancia de variables

---

```

1:  for fallo=1:22 do
2:    for variable=1:52 do
3:      for componente principal=1:52 do
4:        Relevancia de una variable i  $R_i = \frac{\sum_{j=1}^{52} \alpha_{ij}}{\sum_{k=1}^{52} \sum_{j=1}^{52} \alpha_{kj}}$ 
5:      end for
6:    end for
7:  end for

```

---

Así las variables podrán ordenarse en cada caso en función su relevancia en el conjunto. El problema es que contamos con estas listas de variables ordenadas para los datos de test y entrenamiento por separado.

Tal y como se ha descrito en la propuesta del capítulo anterior, se necesitan diversos rankings de variables que se utilizarán en los dos tipos de modelos descritos. En dichos rankings las posiciones más altas estarán ocupadas por las variables con más relevancia, y deben haberse tenido en cuenta los datos de test y entrenamiento en conjunto, por lo que es necesario el proceso de toma de decisiones. Los rankings necesarios son:

- **Ranking para modelos globales:** En ellos se utilizan los datos de entrenamiento y test del conjunto de datos al completo. Para realizar el ranking de variables se utilizan los operadores OWA con el parámetro orness 0,6.
- **Rankings para modelos parciales:** Se generará un ranking para cada uno de los tipos de fallo a partir sus propios datos. También se realizarán con OWA (orness=0,6)

La función RIM que se utiliza es  $Q_{\alpha} = r^{\alpha}$  (Ec. 26)

---

**M. experimental 5: Rankings**

---

```
1: Ranking global:
2: For fallo=1:22 do
3:   Ranking_variables_global=Metodo OWA (Orden variables entrenamiento,
   orden variables test, orness=0.6)
4: end for
Rankings parciales
5: For fallo i=1:22 do
   Ranking_variables_fallo i=Metodo OWA (Orden variables entrenamiento
   fallo i, orden variables test fallo i, orness=0.6)
6: end for
```

---

Se seleccionan tres variables para hacer frente a un posible problema de maldición de la dimensionalidad, que expresa que para dimensiones de los datos altas se obtiene una baja precisión al estimar funciones de densidad de la muestra. (Delicado, 2008)

Después de esto, realizaremos diversos experimentos con los **modelos globales**. Estos experimentos se realizarán con el conjunto de datos de entrenamiento y test correspondiente a las tres variables que copan su ranking.

Así se procede a generar distintos modelos con cada una de las redes (ANFIS, RBF, Backpropagation) utilizando diferentes combinaciones de parámetros con el fin de encontrar el modelo más óptimo (que tenga menos errores de entrenamiento y de test).

Como en las tres redes el entrenamiento es supervisado (véase tipos de redes en el capítulo anterior) se debe indicar una salida deseada para cada entrada. Así se irán ajustando los pesos psinópticos basándose en la diferencia entre la salida real de la red y la deseada.

Las salidas deseadas en todos estos casos indicarán que a una entrada de datos del caso 1 le corresponde una salida de valor 1, ante una entrada del caso 2 una salida que valga 2, y así sucesivamente.

Los parámetros utilizados para los experimentos serán los siguientes:

- **Anfis:**
  - *Radio de influencia: 0.01 – 0.7*
  - *Número de épocas de entrenamiento: 10 -1000*
- **RBF:**
  - *Spread: 1 - 4*
  - *Número de neuronas: 10 – 500*



- **Backpropagation:**
  - *Número de neuronas: 10 – 5000*
  - *Iteraciones de entrenamiento: 5 – 500*

Como se ha explicado en nuestra muestra existen dos tipos de datos, unos destinados al entrenamiento y otros destinados al testado (archivos XX\_TE.dat)

---

**M. experimental 6:** Experimentos con modelos globales

---

```
Definir salidas deseadas de los modelos:
1: for fallo=1:22 do
2:     Salidas entrenamiento((fallo-1)*480+1:480*fallo,1)=fallo*1
3:     Salidas test((fallo-1)*800+1:800*fallo,1)=1*fallo
4: end

Anfis
5: for epocas=10:10:1000 do
6:     for radio=0.1:0.05:0.7 do
7:         (Error entrenamiento, error test) = anfis (datos entrenamiento, datos
            test, épocas, radio, salidas entrenamiento, salidas test)
8:     end for
9: end for

RBF
10: Mean squared error goal=0.001
11: Numero de neuronas=25
12: for spread=1:4 do
13:     for número de neuronas=10:10:500 do
14:         (Error entrenamiento, error test) = rbf (datos entrenamiento, datos
            test, Número de neuronas, spread, salidas entrenamiento, salidas
            test)
15:     end for
16: end for

BackPropagation
17: for iteraciones de entrenamiento=5:20:500 do
18:     for numero de neuronas=10:10:5000 do
19:         (Error entrenamiento, error test) = backpropagation (datos
            entrenamiento, datos test, Número de neuronas, iteraciones, salidas
            entrenamiento, salidas test)
20:     end for
21: end for
```

---

Una vez seleccionados los mejores modelos (aquellos que tienen menor error de entrenamiento y test para cada red), estos serán testados por el conjunto de datos y así se construirá una matriz en la que se indique el porcentaje de veces

que ante datos de un modo de funcionamiento  $i$  el sistema los clasifica dentro de cada modo de funcionamiento  $j$ .

---

**M. experimental 7: Testeo de los modelos globales seleccionados**

---

```
Modelos globales
1:  for fallo i=1:22 do
2:      k=23
3:      for fallo j=1:22 do
4:          for Modelo={Mejor_Anfis, Mejor_RBF, Mejor_BP} do
5:              If (abs(modelo(datos fallo j)-i))<k
6:                  k= abs(modelo(datos fallo j)-i)           //Se guarda el n° de
                                                                fallo detectado
7:              end if
8:          end for
9:          Matriz_confusion_modelo(i,j)=                     //Suma el % que
                                                                supone detectar una
                                                                vez un fallo
                                                                =Matriz_confusion_modelo(i,j)+(0.125)
10:      end for
11:  end for
```

---

Los resultados de los experimentos y las matrices citadas se encuentran en el siguiente epígrafe de resultados.

Otra parte de la experimentación de este trabajo se realizará con los **modelos parciales**. Estos se generarán mediante el entrenamiento de las mismas redes neuronales con cada uno de los casos en particular. Es decir, para el modelo de cada uno de los fallos se utilizarán sus propios datos de test y entrenamiento habiendo seleccionado las variables según su propio ranking.

El objetivo de estos experimentos será encontrar el mejor modelo posible (aquel con menores errores de test y entrenamiento) y sus parámetros para cada modo de funcionamiento.

Al igual que en los modelos globales, las redes usadas se basan en un algoritmo supervisado, y por ello hay que proporcionarles la salida deseada. En cada fallo, esa salida corresponderá al número de fallo.

Los parámetros que usaremos para los experimentos son:

**1. Anfis:**

En este algoritmo se variará el valor de dos parámetros: El número de épocas (Desde 10 hasta 1000 de 20 en 20) y el radio, que en cada caso se ha escogido los valores que dan un resultado.

Fallo	Radio
1	0.1:0.01:0.42
2	0.1:0.01:1
3	0.1:0.01:0.69
4	0.1:0.01:0.56
5	0.1:0.01:0.44
6	0.1:0.01:1
7	0.1:0.01:0.51
8	0.1:0.01:0.84
9	0.1:0.01:0.70
10	0.1:0.01:0.49
11	0.1:0.01:0.57
12	0.1:0.01:0.71
13	0.1:0.01:0.69
14	0.1:0.01:0.59
15	0.1:0.01:0.51
16	0.1:0.01:0.51
17	0.1:0.01:0.54
18	0.1:0.01:1
19	0.1:0.01:0.55
20	0.1:0.01:0.61
21	0.1:0.01:0.49
22	0.1:0.01:0.46

Tabla 3: Parametros modelos parciales ANFIS  
(Nº épocas=10:10:1000)

## 2. RBF:

En el caso de esta red se variará el spread entre 1 y 4 y el número de neuronas (Desde 10 hasta 500 de 10 en 10).

## 3. Backpropagation:

En cuanto al tercer algoritmo se variará el número de iteraciones de entrenamiento (De 10 a 5000 en intervalos de 10) y el número de neuronas (Desde 5 hasta 500 de 10 en 10).

**M. experimental 8: Experimentos con modelos parciales**

---

```
Anfis:
1:  for fallo=1:22 do
2:      for epocas=10:20:1000 do
3:          r2= {0.42, 1, 0.69, 0.56, 0.44, 1, 0.51, 0.84, 0.7, 0.49, 0.57, 0.71,
4:              0.69, 0.59, 0.51, 0.51, 0.54, 1, 0.55, 0.61, 0.49, 0.46}
5:          for radio=0.1:0.01:r2 do
6:              Salidas entrenamiento((fallo-1)*480+1:480*fallo,1)=fallo*1
7:              Salidas test((fallo-1)*800+1:800*fallo,1)=1*fallo
8:              (Error entrenamiento, error test) = anfis (datos entrenamiento,
9:                  datos test, épocas, radio, salidas entrenamiento, salidas test)
10:         end for
11:     end for
12: end for

RBF:
13: for fallo=1:22 do
14:     Mean squared error goal=0.001
15:     Numero de neuronas=25
16:     for spread=1:4 do
17:         for número de neuronas=10:10:500 do
18:             Salidas entrenamiento((fallo-1)*480+1:480*fallo,1)=fallo*1
19:             Salidas test((fallo-1)*800+1:800*fallo,1)=1*fallo
20:             (Error entrenamiento, error test) = rbf (datos entrenamiento,
21:                 datos test, Número de neuronas, spread, salidas entrenamiento,
22:                 salidas test)
23:         end for
24:     end for
25: end for

BackPropagation:
26: for fallo=1:22 do
27:     for Iteraciones de entrenamiento=10:10:5000 do
28:         for numero de neuronas=5:10:500 do
29:             Salidas entrenamiento((fallo-1)*480+1:480*fallo,1)=fallo*1
30:             Salidas test((fallo-1)*800+1:800*fallo,1)=1*fallo
31:             (Error entrenamiento, error test) = backpropagation (datos
32:                 entrenamiento, datos test, Número de neuronas, iteraciones,
33:                 salidas entrenamiento, salidas test)
34:         end for
35:     end for
36: end for
```

---

Una vez se ha seleccionado cual es el mejor modelo de cada red y cada uno de los datos se procede a realizar un testeo de todos los datos para comprobar la validez de cada uno de los modelos.

Para realizar esta tarea se testeará el mejor modelo entrenado para cada fallo con el conjunto de los datos. Así se crea una matriz en la que se refleja el error de testear el modelo del fallo i con los datos del fallo j.

---

**M. experimental 9: Testeo de los modelos parciales seleccionados**

---

```
1: for fallo i=1:22 do
2:     for fallo j=1:22 do
3:         for Modelo={Mejor_Anfis_Fallo_i,           //Modelos entrenados
                     Mejor_RBF_Fallo_i, Mejor_BP_Fallo_i} do      con fallo i
4:             Error_test=modelo(fallo j)           //Testeo con fallo j
5:             Matriz_confusion_modelo(i,j)=Error_Test
6:         end for
7:     end for
8: end for
```

---

Los resultados de los experimentos y estas matrices se pueden encontrar en el siguiente epígrafe de resultados.

Además se generarán distintas gráficas en las que se aprecien estos resultados. Estas representarán el error de test al testear cada uno de los modelos del fallo i con los datos correspondientes al resto de fallos.

El último apartado de la metodología es la **toma de decisiones**. Como ya se ha explicado en la propuesta se crearán 4 configuraciones de modelos con el objetivo de conseguir un sistema de detección de fallos lo más eficaz posible.

Estos sistemas creados contarán con 22 modelos, uno por cada fallo o modo de funcionamiento.

Cada uno de esos 22 modelos será sometido a un proceso de test con el conjunto de datos de la planta ofreciendo así un error de test. Estos errores de test servirán para tomar la decisión de cuál es el modo de funcionamiento existente en ese momento en la planta.

Para cada sistema el modelo que aporte el menor error de test será elegido como fallo que describe el funcionamiento de la planta en ese instante.

El proceso utilizado para medir el éxito aportado por cada uno de los sistemas será el siguiente:

1. Para cada sistema, los 22 modelos se entrenan conforme a los parámetros descritos anteriormente.
2. Todos ellos testean con el conjunto de fallos para obtener el error de test de cada uno en cada dato de entrada.
3. Para cada dato de entrada se busca el modelo que genere el menor error.

4. Si el modelo seleccionado corresponde con el dato de entrada la detección habrá sido exitosa.
5. Se repite el proceso con todos los datos de entrada para calcular el porcentaje de éxito.

---

**M. experimental 10:** Toma de decisiones

---

```
1:  for sistema=1:4 do
2:      Entrenamiento de modelos (22)
3:      Inicializar k, éxito, fracaso
4:      for fallo j=1:22 do
5:          for dato_fallo_j=1:800 do
6:              for Modelo=1:22 do
7:                  Error_test=Modelo(dato_fallo_j)
8:                  If (Error_test<k) do
9:                      k=Error_test
10:                     Modelo_error_minimo=Modelo
11:                 end if
12:             end for
13:             If fallo=Modelo_error_minimo do
14:                 éxito=éxito+1
15:             end if
16:             else
17:                 fracaso=fracaso+1
18:             end else
19:         end for
20:     end for
21: end for
```

---

A continuación se pasan a describir cuales son las cuatro configuraciones de modelos utilizados y cuáles son los parámetros con los que se entrenarán dichos modelos.

El primero de estos cuatro sistemas será uno en el que los 22 modelos corresponden a los entrenados con la red **Anfis**. Cada uno de estos modelos está específicamente diseñado para la detección de un fallo, y el conjunto de ellos podría detectar cualquiera de los fallos típicos de la planta y diagnosticarlo de una forma correcta.

El resultado conseguido será mejor cuanto menor sea el error de test del fallo *i* para el modelo *i* pero mayor sea el error de test medio cuando ante un modelo *i* se introducen datos de un fallo *j*.

La configuración de este sistema es la siguiente, donde pueden apreciarse los distintos valores de los parámetros del radio de influencia y del número de épocas de entrenamiento:

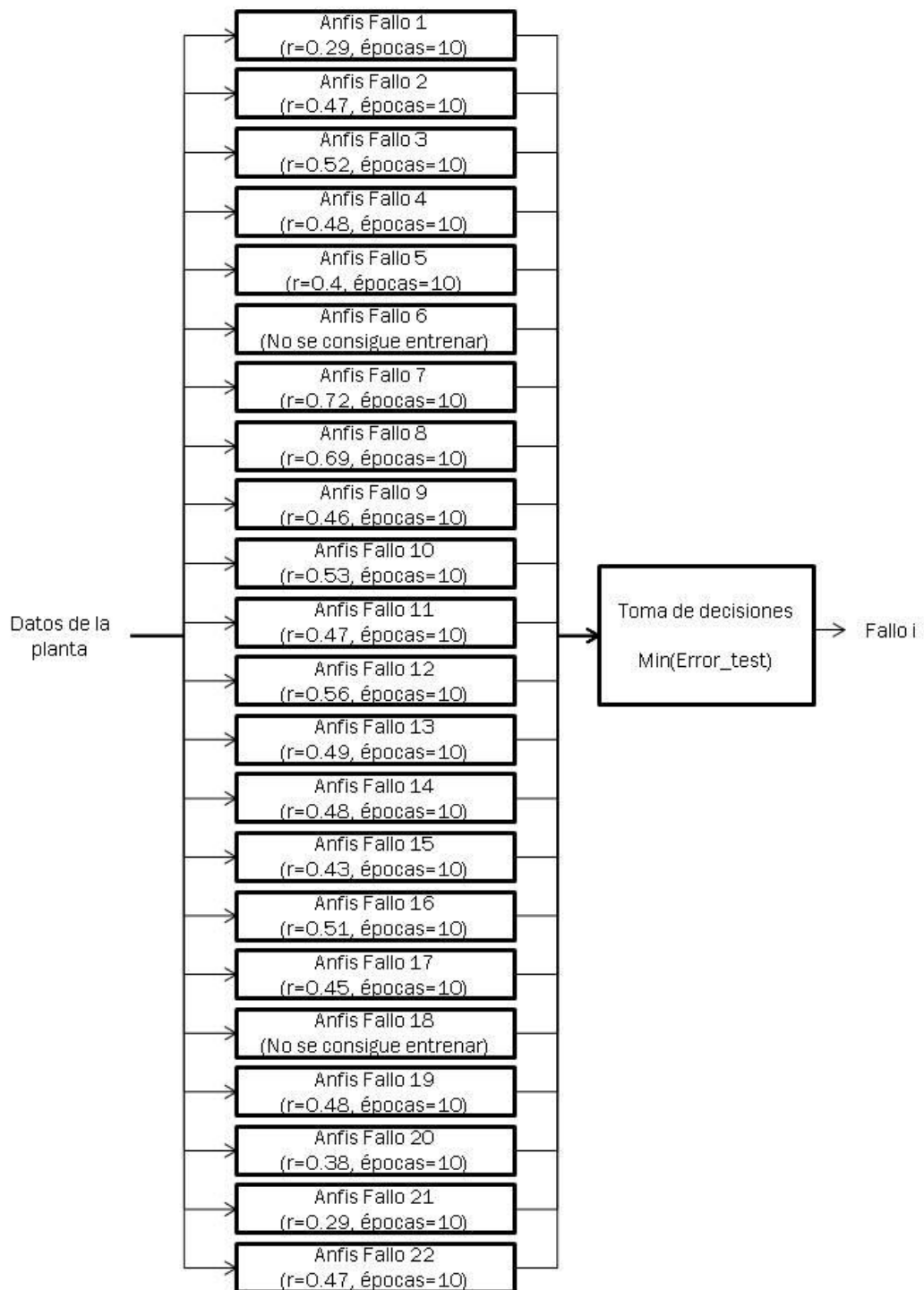


Fig 15: Esquema de sistema de FDI con modelos ANFIS

El segundo de los sistemas de detección de fallos que se creará será el basado en los modelos entrenados por la red neuronal **RBF**. En este caso contará con 22 modelos, e igual que en el caso anterior cada uno de ellos será especialista en la detección de uno de ellos.

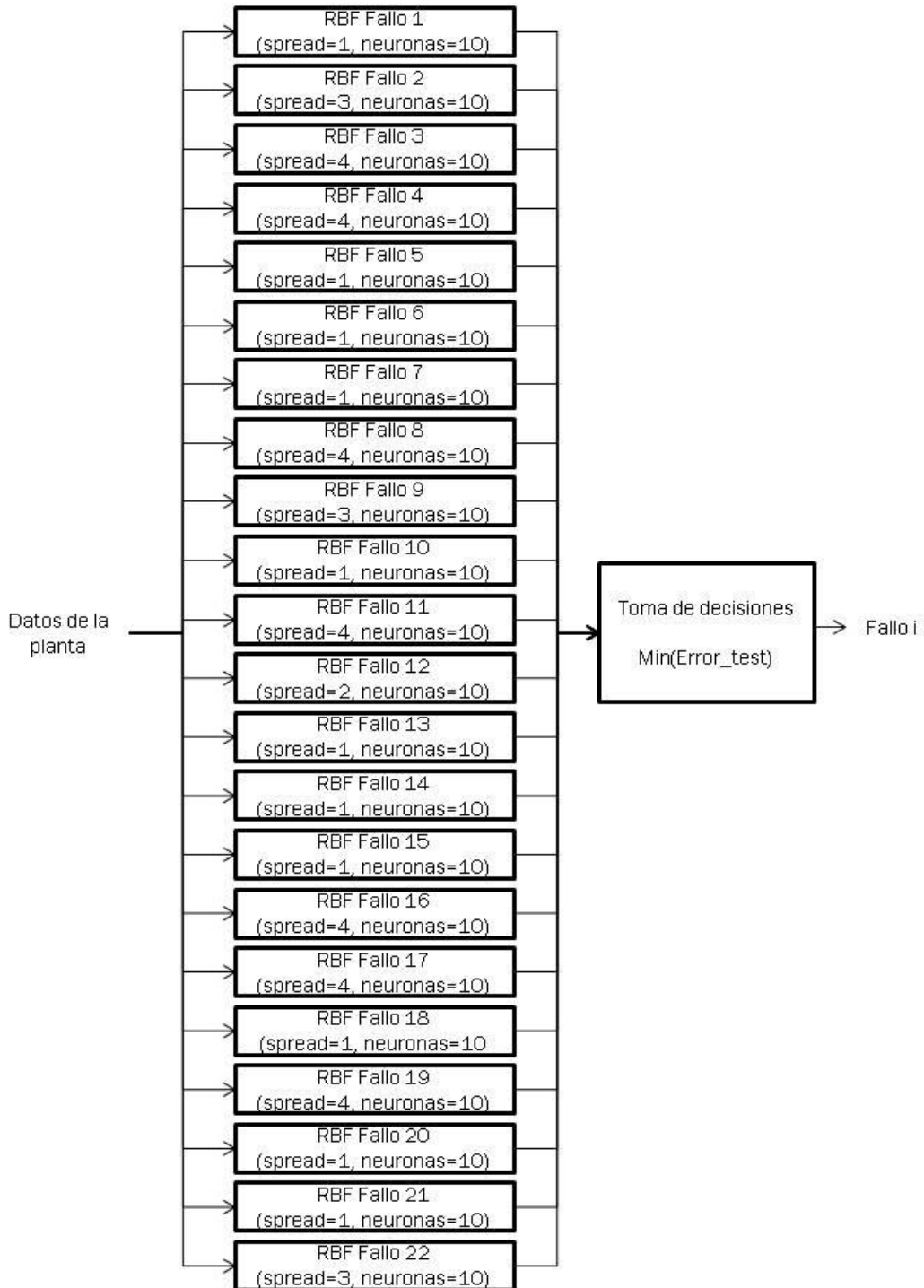


Fig 16: Esquema de sistema de FDI con modelos RBF



El tercero de los sistemas propuesto tiene que ver con los 22 mejores modelos entrenados con la red **Backpropagation**. Al igual que los dos anteriores cada uno de estos está específicamente entrenado para detectar un tipo de fallo.

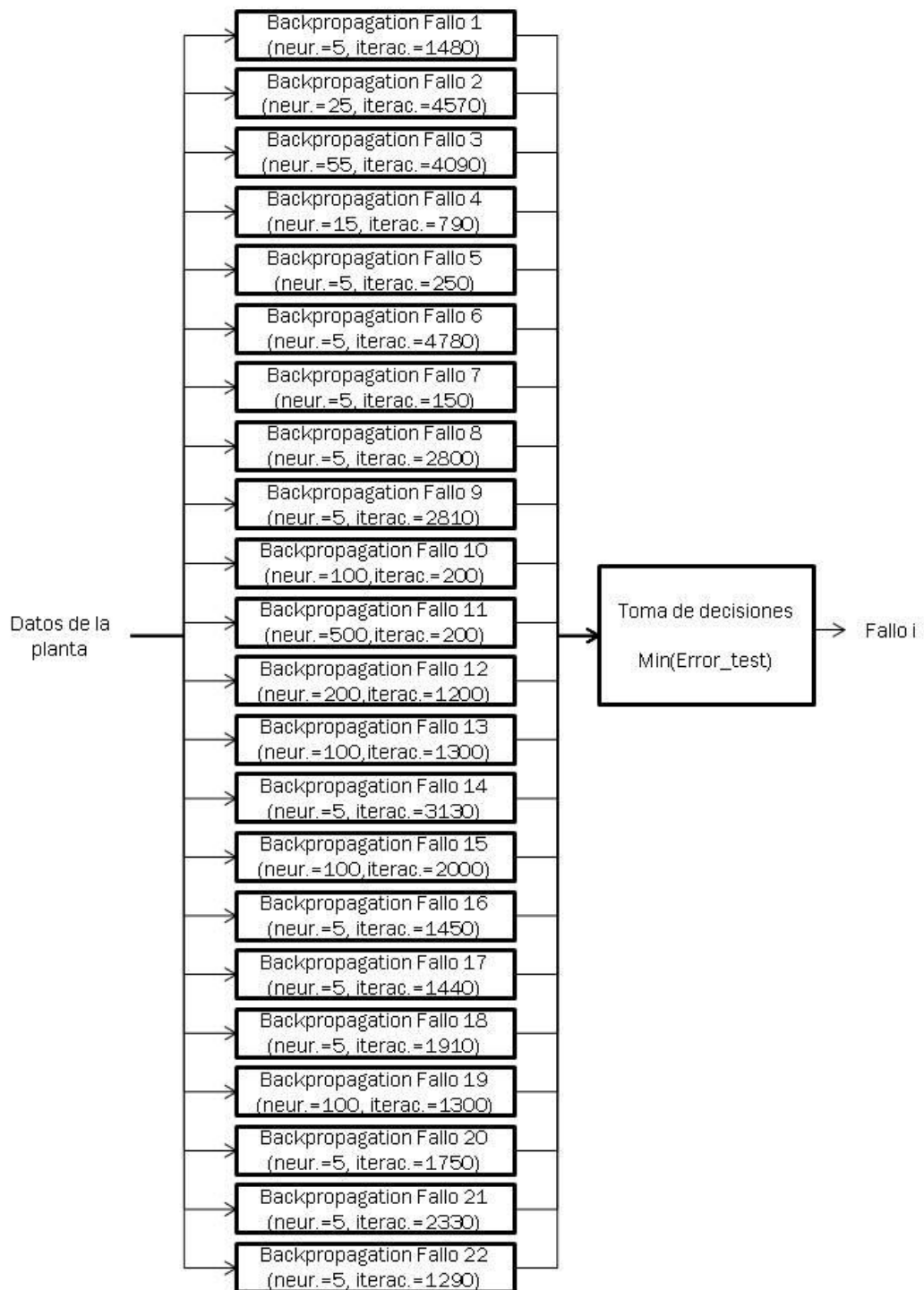


Fig 17: Esquema de sistema de FDI con modelos Backpropagation

El cuarto sistema consistirá en la combinación de 22 modelos, uno referente a cada fallo j. El modelo de cada uno de los modos de funcionamiento será el

mejor modelo elegido entre los entrenados con las redes **ANFIS**, **RBF** y **Backpropagation**.

Para esa selección se medirá cual es el error de test del modelo  $i$  con el fallo  $j=i$  y la media de los errores de test cuando el modelo  $i$  se testea con los casos  $j \neq i$ .

Los errores calculados por cada una de estas redes está en un intervalo distinto, por lo que en primer lugar será necesario escalarlos para poder hacer la comparación entre sus valores.

En la siguiente tabla se puede apreciar el resultado de las tres redes.

Modelo i	ANFIS		RBF		BACKPROPAGATION	
	Error de test Fallo i Modelo i	Error test Fallo j Modelo i $i \neq j$	Error de test Fallo i Modelo i	Error test Fallo j Modelo i $i \neq j$	Error de test Fallo i Modelo i	Error test Fallo j Modelo i $i \neq j$
1	0,711	1,000	0,500	1,000	0,500	1,000
2	0,711	1,000	0,500	1,000	0,500	1,000
3	0,711	1,000	0,500	1,000	0,500	1,000
4	0,711	1,000	0,500	1,000	0,500	1,000
5	0,711	1,000	0,500	1,000	0,500	1,000
6	0,500	0,500	0,500	1,000	0,500	1,000
7	0,711	0,999	0,500	1,000	0,500	1,000
8	0,711	0,997	0,500	1,000	0,500	1,000
9	0,711	0,995	0,500	1,000	0,500	1,000
10	0,711	0,993	0,500	1,000	0,500	1,000
11	0,711	0,991	0,500	1,000	0,500	1,000
12	0,711	0,991	0,500	1,000	0,500	1,000
13	0,711	0,993	0,500	1,000	0,500	1,000
14	0,711	0,995	0,500	1,000	0,500	1,000
15	0,711	0,997	0,500	1,000	0,500	1,000
16	0,711	0,999	0,500	1,000	0,500	1,000
17	0,711	0,999	0,500	1,000	0,500	1,000
18	0,500	0,500	0,500	1,000	0,500	1,000
19	0,711	1,000	0,500	1,000	0,500	1,000
20	0,711	1,000	0,500	1,000	0,500	1,000
21	0,711	1,000	0,500	1,000	0,500	1,000
00	0,711	1,000	0,500	1,000	0,500	1,000

Tabla 4: Cuarto sistema de modelos para FDI usando ANFIS, RBF y Backpropagation

Ante estos resultados modelos más favorables (con menor error si  $j=i$  y mayor error si  $j \neq i$ ) son los correspondientes a las redes RBF y Backpropagation. Si hacemos la relación:

$$\frac{\text{Error test } (j=i)}{\text{Error test } (j \neq i)} \quad (\text{Ec. 27})$$

Podremos observar cual es el modelo más favorable. Haciendo esto y teniendo en cuenta que existen más decimales de los mostrados en la tabla vemos que los mejores modelos para todos los fallos son los correspondientes a RBF, por lo que el cuarto sistema de modelos coincide con el segundo (**Fig 16: Esquema de sistema de FDI con modelos RBF**).

Como se ha especificado, en cada sistema se elegirá como resultado de la toma de decisiones el modelo que menor error de test aporte. Este tipo de decisiones equivale a un OWA con:  $w_n = 1$  y  $w_j = 0 \quad \forall j \neq n$ . (Lorenzón, 2011)

## 4.3. Resultados

### 4.3.1. Procesamiento de datos

Tal y como se explica en el Cap 4.2. Metodología experimental en cuanto al procesamiento de la muestra se realiza, la búsqueda de *outliers*, la detección de missing values, y la normalización de los datos de la muestra.

En lo que respecta a los valores faltantes o missing values la muestra de datos no contiene ninguno. Todas las muestras contienen información para el conjunto completo de variables.

Los resultados conseguidos con esta normalización se pueden ver reflejados en la representación de una variable cuando los datos no están normalizados y cuando sí lo están.

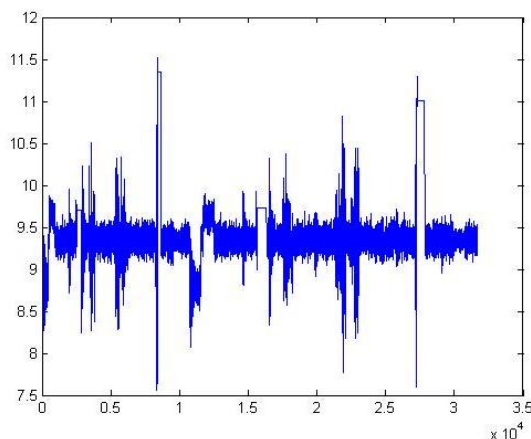


Fig 18: Ejemplo de datos sin normalizar (variable 4)

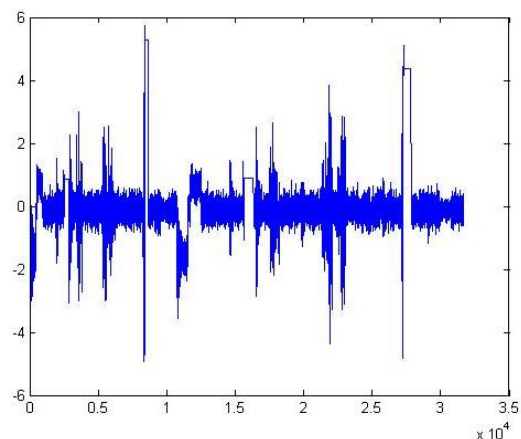


Fig 19: Ejemplo de datos normalizados (variable 4)

Como se puede ver en las anteriores figuras el hecho de llevar a cabo este proceso de normalización supone que los datos sin normalizar contenidos en el intervalo  $[7.5, 11.5]$  del eje de ordenadas pasan a ocupar el intervalo  $[-6, 6]$  de dicho eje. Es decir, los datos correspondientes a todas las variables pasan a centrarse alrededor del eje de abscisas.

Esto supone numerosas ventajas para nuestro proceso:

- Acelerará el proceso de aprendizaje
- Evitará problemas de precisión y posibles desbordamientos aritméticos (overflows)
- Define todas las variables en las mismas escalas numéricas.

### 4.3.2. Selección de variables

#### 4.3.2.1. Análisis de componentes principales

En primer lugar se calcula variabilidad aportada por cada una de las componentes principales en el conjunto. En la siguiente tabla se encuentran las primeras 20 componentes, que son las que aportan una variabilidad de la información significativa:

Componente	Datos de entrenamiento	Datos de test
1	16,8%	15,9%
2	15,4%	14,9%
3	8,1%	8,1%
4	7,5%	6,9%
5	6,8%	6,6%
6	6,8%	6,1%
7	5,3%	6,1%
8	5,0%	5,2%
9	4,5%	4,7%
10	3,9%	4,0%
11	3,6%	3,4%
12	3,5%	3,3%
13	3,1%	3,2%
14	1,4%	2,0%
15	1,3%	1,3%
16	1,1%	1,1%
17	0,7%	1,1%
18	0,7%	0,9%
19	0,6%	0,7%
20	0,6%	0,6%

**Tabla 5:** Variabilidad aportada por cada PC

Una vez aplicado el método de análisis de componentes principales (PCA) y habiendo obtenido cual es el porcentaje de variabilidad que supone conservar cada una de las componentes principales, se trata de encontrar un equilibrio entre el número de componentes principales seleccionadas y la variabilidad de la información que representan.

Para ello se ha analizado el número de componentes principales que en conjunto supongan al menos un 70 %, un 80% y un 90% de la variabilidad de la información. En la siguiente tabla se representan el número de componentes

principales a seleccionar para conservar al menos los distintos porcentajes de la información en los datos de entrenamiento. Además refleja el porcentaje de componentes principales que supondría pasar de un 70% a un 80% y de un 80% a un 90% de la variabilidad.

Número de componentes principales					
Modo de funcionamiento	70% de variabilidad	Diferencia 70% vs 80%	80% de variabilidad	Diferencia 80% vs 90%	90% de variabilidad
D01	7	+ 3 (42%)	10	+4 (40%)	14
D02	5	+3 (60%)	8	+5 (62.5%)	13
D03	9	+2 (22%)	11	+3 (27%)	14
D04	8	+3 (37.5%)	11	+3 (27%)	14
D05	8	+3 (37.5%)	11	+4 (36%)	15
D06	2	+0 (0%)	2	+6 (300%)	8
D07	5	+3(60%)	8	+5 (62.5%)	13
D08	4	+3(75%)	7	+5 (71%)	12
D09	9	+2 (22%)	11	+3 (27%)	14
D10	9	+3 (33%)	12	+3 (25%)	15
D11	9	+2(22%)	11	+3 (27%)	14
D12	5	+2 (40%)	7	+5 (71%)	12
D13	5	+3 (60%)	8	+5 (62.5%)	13
D14	5	+3 (60%)	8	+4 (50%)	12
D15	8	+3 (37.5%)	11	+3 (27%)	14
D16	9	+2 (22%)	11	+4 (36%)	15
D17	5	+3 (60%)	8	+4 (50%)	12
D18	1	+1 (100%)	2	+1 (50%)	3
D19	7	+2 (28.5%)	9	+4 (44%)	13
D20	9	+3 (33.3%)	12	+3 (25%)	15
D21	8	+3 (37.5%)	11	+3 (27%)	14
D00	8	+3 (37.5%)	11	+2 (18%)	13

Tabla 6: PC según la variabilidad conservada (Datos entrenamiento)

Para facilitar la decisión del porcentaje de variabilidad conservada también se ha calculado el número de componentes principales que habría que conservar a mayores si pasamos de un 70% a un 80% de variabilidad y de un 80% a un 90%. Los valores medios se pueden ver en la siguiente tabla:

Diferencia entre		Aumento de PC	Aumento de PC (%)
70%	80%	+2.5	+42.24%
80%	90%	+2.71	+53.15%

Tabla 7: Aumento de PC al aumentar la variabilidad (Datos entrenamiento)

Tal y como se ha explicado en el capítulo 3 (Metodología experimental) de este documento, nuestro caso de estudio cuenta con dos tipos de datos. El primer

tipo, al que ya hemos aplicado el análisis de componentes principales, y un segundo tipo aún sin analizar.

Estos nuevos datos constan de dos partes: En la primera de esas partes se sitúan los 160 instantes en los que el funcionamiento es el nominal, y el resto, desde el instante 161 hasta el 960 que el funcionamiento tiene fallo.

Procedemos también a realizar el mismo proceso con los ficheros de test pero analizando solo los instantes de tiempo en los que lo tiene, es decir, pasando el instante 160.

Número de componentes principales					
Modo de funcionamiento	70% de la variabilidad	Diferencia 70% vs 80%	80% de la variabilidad	Diferencia 80% vs 90%	90% de la variabilidad
D01_te	9	+3 (33%)	12	+3 (25%)	15
D02_te	7	+3 (42%)	10	+4 (40%)	14
D03_te	9	+2 (22%)	11	+4 (36%)	15
D04_te	9	+2 (22%)	11	+3 (27%)	14
D05_te	9	+2 (22%)	11	+4 (36%)	15
D06_te	2	+0 (0%)	2	+4 (200%)	6
D07_te	7	+3 (42%)	10	+4 (40%)	14
D08_te	6	+3 (50%)	9	+5 (55%)	14
D09_te	9	+2 (22%)	11	+4 (36%)	15
D10_te	10	+2 (20%)	12	+4 (33%)	16
D11_te	9	+3 (33%)	12	+3 (25%)	15
D12_te	4	+2 (50%)	6	+6 (100%)	12
D13_te	3	+2 (66%)	5	+5 (100%)	10
D14_te	5	+3 (60%)	8	+4 (50%)	12
D15_te	9	+2 (22%)	11	+4 (36%)	15
D16_te	9	+3 (33%)	12	+3 (25%)	15
D17_te	5	+3 (60%)	8	+5 (62%)	13
D18_te	1	+1 (100%)	2	+1 (50%)	3
D19_te	7	+2 (28%)	9	+4 (44%)	13
D20_te	9	+3 (33%)	12	+3 (25%)	15
D21_te	8	+3 (37%)	11	+3 (27%)	14
D22_te	9	+2 (22%)	11	+3 (27%)	14

**Tabla 8:** PC según la variabilidad conservada (Datos test)

Realizamos el mismo análisis que con los datos de test. En la siguiente tabla se puede observar cuantas componentes principales son necesarias para aumentar la variabilidad (valores medios):



Diferencia entre		Aumento de PC	Aumento de PC (%)
70%	80%	+2.31	+37.5%
80%	90%	+3.77	+50.14%

Tabla 9: Aumento de PC al aumentar la variabilidad (Datos test)

Así, se toma la decisión de conservar un 70% de la información para llevar a cabo los experimentos.

#### 4.3.2.2. Selección de variables

El siguiente paso de la metodología es el cálculo de la relevancia que tiene cada variable con respecto al conjunto, tanto en los datos referidos a entrenamiento como a los de test. Estos resultados se utilizarán para construir los rankings de variables de la siguiente sección.

Debido a la extensión de las mismas, estas tablas se pueden ver en los anexos de este proyecto.

#### 4.3.2.3. Ranking de variables: Método OWA

Tras obtener la relevancia de cada una de las variables en cada uno de los fallos considerados se procede a realizar diferentes tipos de rankings de variables según su relevancia.

Como ahora se ha explicado en los capítulos anteriores, se realizarán dos tipos de rankings:

- El primero basados en la muestra completa de datos, en el que se ordenan las variables teniendo en cuenta los datos correspondientes a todos los fallos. Estos (tal y como se ha explicado) se utilizarán para la creación de **modelos globales**.
- El segundo tipo de rankings, utilizado para **modelos parciales**, en los que se tendrán en cuenta los valores de las variables en cada fallo en particular.

Para realizar dichos rankings se utilizará los operadores OWA.

En primer lugar se realizará un ranking de las variables más ‘relevantes’ para modelos globales. Al realizar dicho ranking se escogen las tres variables que ocupan los puestos más altos. En primer lugar se selecciona la variable número 25, en segundo lugar la número 16 y por último la número 14.

Posición en el ranking:	Variable:
1	25
2	16
3	14

**Tabla 10:** Variables relevantes modelos globales (orness=0.6)

El ranking completo puede encontrarse en el apartado de anexos del trabajo.

Por otra parte, como se ha explicado, se realiza un nuevo ranking para cada uno de los 22 fallos de forma individual y de la misma forma se seleccionan las tres variables más relevantes de cada uno.

Ranking para el fallo:	Variable en la 1ª posición	Variable en la 2ª posición	Variable en la 3ª posición
1	1	2	5
2	16	18	17
3	14	34	19
4	34	27	18
5	1	17	14
6	1	2	12
7	1	2	24
8	4	2	1
9	14	25	19
10	3	24	12
11	15	26	16
12	1	4	28
13	1	3	2
14	25	22	18
15	21	14	29
16	14	20	12
17	16	17	22
18	1	17	3
19	16	26	18
20	15	13	16
21	25	22	33
00	34	20	22

**Tabla 11:** Variables relevantes para cada fallo (orness=0.6)

Todos los rankings completos pueden verse en el apartado de anexos de la memoria.

### 4.3.3. Modelos de detección y diagnóstico de fallos

Cómo se recoge en la metodología experimental, expuesta en el tercer capítulo de este documento, el siguiente paso será el entrenamiento de las redes neuronales.

Las tres redes elegidas (ANFIS, RBF y Backpropagation) se entrenarán de dos formas: con el conjunto de los datos de entrenamiento para generar los modelos globales y con los datos de entrenamiento de cada modo de funcionamiento (fallos) por separado dando lugar a los modelos parciales.

Además en cada uno de estos casos se entrenarán las redes con distintas combinaciones de parámetros buscando obtener cuales son aquellos que consiguen un entrenamiento y un test con menor error.

Un menor error implica que el modelo distingue mejor un modo de funcionamiento de otro, es decir, pone más exigencia a la hora de clasificar un dato como un modo de funcionamiento.

#### 4.3.3.1. Modelos globales

Aquí se explican los resultados de los distintos experimentos realizados y cuáles son los parámetros que para cada red generan menor error.

- **Anfis:**

Tal y como se expresa en la metodología experimental (capítulo 3) para esta red se han variado dos parámetros. Estos han sido:

- *Radio de influencia*
- *Número de épocas de entrenamiento*

Con todos los experimentos realizados los mejores resultados conseguidos son:

Algoritmo	Modelo	Radio de influencia	Nº de épocas	Error de entrenamiento	Error de test
Anfis	Global	0.56	10	37,9534555	134,726678

*Tabla 12: Mejor resultado para modelos globales Anfis*

- **RBF:**

En esta segunda red neuronal los parámetros con los que se han realizado los experimentos han sido:

- *Spread*
- *Número de neuronas*

Habiendo realizado los experimentos con todas las combinaciones posibles de estos parámetros, en la siguiente tabla se pueden encontrar cuales han sido los mejores resultados obtenidos.

Algoritmo	Modelo	Spread	Nº neuronas	Error <sub>ent</sub>	Error <sub>test</sub>
RBF	Global	3	460	107,074248	95,7653688

**Tabla 13:** Mejor resultado para modelos globales RBF

- **Backpropagation:**

En la red de backpropagation los experimentos se han realizado con todas las combinaciones posibles de los parámetros:

- Número de neuronas
- Número de iteraciones

Los mejores resultados obtenidos han sido:

Algoritmo	Modelo	Iteraciones de entrenamiento	Nº neuronas	Error <sub>ent</sub>	Error <sub>test</sub>
Backpropagation	Global	5	370	38,580454	131,86

**Tabla 14:** Mejor resultado para modelos globales Backpropagation

A primera vista nos damos cuenta de que tanto los errores de test como de entrenamiento mínimos alcanzados para los modelos globales con cada una de las redes son valores muy altos. Considerando que los rangos utilizados para las variaciones de los parámetros han sido suficientes, esto podría indicarnos que los errores cometidos al entrenar una red con tantos tipos de fallos y datos no aportan resultados demasiado buenos.

Una vez extraídos los mejores modelos para cada una de las redes se procede crear las **matrices de confusión** explicadas en la propuesta (Cap.3.3. Etapa 3: Modelado basados en Redes Neuronales) con el fin de poder interpretar mejor el resultado. Para ello se va a testear con cada uno de los modelos el conjunto de datos.

Así, en una matriz de confusión se expresará en porcentaje, cuantas de las veces el modelo reconoce que ante la entrada de un fallo  $j$ , ese fallo corresponde a un modo de funcionamiento  $j$  y cuantas a un caso  $i$ .

De esta forma cuanto más grandes sean los porcentajes de la diagonal de la matriz mejor se ajustará dicho modelo a nuestros datos.

Además con el objetivo de apreciar mejor los resultados se procede a graficarlos. En estas gráficas, que se encuentran en los anexos del trabajo, se aprecia cuantas veces testeando con el caso i detecta cada uno de los casos.

• Anfis:

		TEST																					
Fallo i	Fallo j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
		1	0	0	0	0	0	0	0	0	0,5	2,625	3	31,87	61,62	0,375	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	1,75	65,37	32,87	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0,125	33	66,75	0,125	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0,125	34,5	65,25	0,125	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	2,375	32,87	64,5	0,25	0	0	0	0	0	0	0	0	0
6	0	0,5	2	10,87	21,87	30,5	20,75	7,125	1,125	1,75	2,625	0,875	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0,125	0,25	2,25	32	63,75	1,5	0,125	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0,125	0,5	9,5	30,25	56,25	3,25	0,125	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0,25	25,62	74,12	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0,125	0,125	31,87	67,87	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0,25	29	70,5	0,25	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	5	33,87	52	6,75	2,125	0,25	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	4,875	24,37	53,5	13	3	1	0,25	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	30,5	69,37	0,125	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	34,5	65,37	0,125	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0,125	30,87	68,62	0,375	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0,25	33,5	66,25	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0,125	1,5	3,875	19,12	1,75	1	0,25	0,375	22,62	42,62	6,75	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0,125	33,12	66,62	0,125	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0,375	34,87	64,75	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	31,37	68,37	0,25	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0,5	32,37	66,87	0,25	0	0	0	0	0	0	0	0	0	0

Tabla 15: Mejor modelo global ANFIS fallo i testado con el fallo j (Radio=0,56, N°épocas=10)

• RBF:

		TEST																					
Fallo i	Fallo j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
		1	0,25	0	0,625	1	0,375	3,25	4	4,375	6,125	9,125	9,875	27,37	33	0,5	0	0,125	0	0	0	0	0
2	0,625	0,625	0,625	0,75	3,375	5,625	9,25	13,25	18,12	18,25	13,75	9,75	6	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0,5	1,125	3,375	10,87	36	47,75	0,375	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0,25	0,875	3,5	10	35,12	50	0,25	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0,375	2,375	1,875	3,75	3,125	4,125	10	36,37	37,62	0,25	0,125	0	0	0	0	0	0	0	0
6	1,5	0,625	0,875	0,75	1,25	1,875	1,25	1,625	1,625	0,75	0,75	0,75	0,25	0	14,62	5,5	5,125	2,625	7,625	2,25	2,5	45,87	5
7	0,375	0,125	0,125	0	0,625	1,625	4	3,375	6,25	10,37	11,75	22,25	37,87	0,875	0,375	0	0	0	0	0	0	0	0
8	0,75	0,25	0,25	0,125	1	8,625	14,12	16,75	15,75	11,75	10	10,37	8,875	0,625	0,375	0	0,125	0	0,125	0	0	0,125	0
9	0	0	0	0	0	0	0	0,375	0,625	3,125	14	30,62	50,87	0,25	0,125	0	0	0	0	0	0	0	0
10	0	0	0	0,125	0	0	0,125	0,875	3,625	11	13,87	27,12	42,87	0,125	0	0,125	0	0	0	0	0	0	0,125
11	0	0	0	0	0	0	0,125	0,25	1	4,25	14,12	27,25	52,5	0,25	0,125	0	0	0	0,125	0	0	0	0
12	1,25	0,25	0,25	0,5	2	7,25	11	13,62	13,37	12,25	11,87	12	9,75	1,625	1,125	0,25	0,375	0,125	0,125	0,125	0,125	0,125	0,75
13	8,875	0,5	0,625	1	2,5	7	7,625	9,375	8,25	9,75	8,125	9,875	8,625	3,125	3,625	2,25	2,25	1,5	1,625	0,875	0,25	2,375	
14	0	0	0	0	0	0,125	0	0,125	0,375	4,125	9,25	30,75	55,12	0,125	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0,125	0,75	1,375	4,625	12,75	33,5	46,87	0	0	0	0	0	0	0	0	0	0
16	0,125	0	0	0	0	0,125	0,125	1,5	3,125	5,875	15,62	31,75	41,37	0,125	0,25	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0,125	0,25	1,375	6	13,37	29,75	48,75	0,375	0	0	0	0	0	0	0	0	0
18	0,75	0	0	0	0,125	0,75	2	1,125	1,125	3,125	2,875	6,25	5,75	0,25	1	0,5	1	0,5	0,125	0,125	0,25	72,37	
19	0	0	0	0	0	0	0	0,625	0,75	2,375	9,375	31,75	54,37	0,5	0,125	0,125	0	0	0	0	0	0	0
20	0	0	0	0	0	0,125	0,25	1,125	3	6,625	13,62	28,37	46,75	0,125	0	0	0	0	0	0	0	0	0
21	0,125	0	0,25	0,25	1	2,625	4,875	10,12	10,37	12,87	16,62	22	18,87	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0,125	0,875	3,375	9,87	34,12	51,12	0,37	0	0	0,125	0	0	0	0	0	0

Tabla 16: Mejor modelo global RBF fallo i testado con el fallo j (Spread=3, N° neuronas=460)

- **Backpropagation**

		TEST																						
Fallo i \ Fallo j		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
E N T R E N A M I E N T O	1	0	0	0	0	0	0	0	0,125	3	6,875	35	48,87	5,625	0,5	0	0	0	0	0	0	0	0	
	2	0	0	0	0	0	0	0	0	0,375	29,62	50,75	12,37	5,25	1,625	0	0	0	0	0	0	0	0	
	3	0	0	0	0	0	0	0	0	0	1,375	46,87	50,5	1,25	0	0	0	0	0	0	0	0	0	
	4	0	0	0	0	0	0	0	0	0	1,125	50	48,37	0,5	0	0	0	0	0	0	0	0	0	
	5	0	0	0	0	0	0	0	0	0,5	4,25	44,37	48,25	2,625	0	0	0	0	0	0	0	0	0	
	6	0	0	0	0	0	0,5	46,87	45,37	2,125	2	2,875	0,25	0	0	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	1,125	5,625	44,87	38,5	6,5	2,625	0,75	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0,125	2,375	15	29,87	27	14	8,125	3,375	0,125	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0	0,5	38,5	59,12	1,875	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	0	1,625	44	51,5	2,875	0	0	0	0	0	0	0	0	0	0
	11	0	0	0	0	0	0	0	0	0	0,625	42,87	55	1,5	0	0	0	0	0	0	0	0	0	0
	12	0	0	0	0	0	0	0	0	0,125	1,25	16	29,5	27	16,37	6,375	2,625	0,75	0	0	0	0	0	0
	13	0	0	0	0	0	0	0	0,25	8,875	22,62	22,25	22,62	11,25	6	3,75	2	0,375	0	0	0	0	0	0
	14	0	0	0	0	0	0	0	0	0	0,5	45	53,37	1,125	0	0	0	0	0	0	0	0	0	0
	15	0	0	0	0	0	0	0	0	0	1,125	50,25	46,12	2,5	0	0	0	0	0	0	0	0	0	0
	16	0	0	0	0	0	0	0	0	0	2,375	42,75	51,25	3,625	0	0	0	0	0	0	0	0	0	0
	17	0	0	0	0	0	0	0	0	0	1,125	47,37	49,75	1,75	0	0	0	0	0	0	0	0	0	0
	18	0	0	0	0	0	0	0	0	0,875	2,25	6,125	10,87	18	54,25	5,75	1,75	0,125	0	0	0	0	0	0
	19	0	0	0	0	0	0	0	0	0	1,5	46,75	50,5	1,25	0	0	0	0	0	0	0	0	0	0
	20	0	0	0	0	0	0	0	0	0	2,875	45,12	49,62	2,375	0	0	0	0	0	0	0	0	0	0
	21	0	0	0	0	0	0	0	0	0	5,125	57,62	36,62	0,625	0	0	0	0	0	0	0	0	0	0
	22	0	0	0	0	0	0	0	0	0	1,75	45,25	52,37	0,625	0	0	0	0	0	0	0	0	0	0

Tabla 17: Mejor modelo global Backpropagation fallo i testado con el fallo j (Iteraciones=5, N°neuronas=370)



A la vista de las tres matrices concluimos que ninguno de los tres modelos es capaz de distinguir un tipo de fallo de otro.

### 4.3.3.2. Modelos parciales de FDI

La segunda parte de los experimentos consiste en realizar el entrenamiento de esas mismas redes neuronales con los datos correspondientes a cada uno de los modos de funcionamiento por separado.

Estas redes serán entrenadas variando diversos parámetros dependiendo de la red, de forma que consigamos los mejores parámetros para cada una de ellas.

- **Anfis:**

Al igual que en los entrenamientos con todos los datos al mismo tiempo, en estos se variará:

- *Radio de influencia*
- *Número de épocas de entrenamiento*

En la siguiente tabla se refleja para cada uno de los modos de funcionamiento cuales han sido los parámetros con los que se obtiene mejor error de test y entrenamiento.

Red	Modelo	Caso	Radio de influencia	Épocas entren.	Error <sub>ent</sub>	Error <sub>test</sub>
Anfis	Parcial	1	0,29	10	0,9	0,9
Anfis	Parcial	2	0,47	10	0,9	0,9
Anfis	Parcial	3	0,52	10	0,9	0,9
Anfis	Parcial	4	0,48	10	0,9	0,9
Anfis	Parcial	5	0,4	10	0,9	0,9
Anfis	Parcial	7	0,38	10	0,9	0,9
Anfis	Parcial	8	0,72	10	0,9	0,9
Anfis	Parcial	9	0,69	10	0,9	0,9
Anfis	Parcial	10	0,46	10	0,9	0,9
Anfis	Parcial	11	0,53	10	0,9	0,9
Anfis	Parcial	12	0,47	10	0,9	0,9
Anfis	Parcial	13	0,56	10	0,9	0,9
Anfis	Parcial	14	0,49	10	0,9	0,9
Anfis	Parcial	15	0,48	10	0,9	0,9
Anfis	Parcial	16	0,43	10	0,9	0,9
Anfis	Parcial	17	0,51	10	0,9	0,9
Anfis	Parcial	19	0,45	10	0,9	0,9
Anfis	Parcial	20	0,51	10	0,9	0,9
Anfis	Parcial	21	0,48	10	0,9	0,9
Anfis	Parcial	00	0,38	10	0,9	0,9

**Tabla 18:** Modelos parciales ANFIS (mejores resultados)

- **RBF:**

En cuanto a la red neuronal de base radial, los parámetros que se utilizan para conseguir el mejor modelo son:

- *Spread*
- *Número de neuronas*

Así los mejores resultados obtenidos para cada caso son:

Red	Modelo	Caso	Spread	Nº neuronas	Error <sub>ent</sub>	Error <sub>test</sub>
RBF	Parcial	1	1	10	2,3863E-29	2,3863E-29
RBF	Parcial	2	3	10	7,88861E-29	7,88861E-29
RBF	Parcial	3	4	10	5,48258E-29	5,06186E-29
RBF	Parcial	4	4	10	2,84779E-28	2,85305E-28
RBF	Parcial	5	1	10	1,97215E-29	1,97215E-29
RBF	Parcial	6	1	10	5,7508E-28	5,7508E-28
RBF	Parcial	7	1	10	0	0
RBF	Parcial	8	4	10	1,39155E-27	1,39155E-27
RBF	Parcial	9	3	10	7,88861E-29	9,04561E-29
RBF	Parcial	10	1	10	1,13596E-28	1,13596E-28
RBF	Parcial	11	4	10	6,20255E-27	5,65771E-27
RBF	Parcial	12	2	10	1,02236E-27	1,02236E-27
RBF	Parcial	13	1	10	2,75575E-28	2,55591E-28
RBF	Parcial	14	1	10	0	0
RBF	Parcial	15	1	10	4,3198E-27	4,3198E-27
RBF	Parcial	16	4	10	5,64193E-27	5,6251E-27
RBF	Parcial	17	4	10	2,8399E-27	2,8399E-27
RBF	Parcial	18	1	10	4,54384E-28	3,15544E-28
RBF	Parcial	19	4	10	3,23117E-27	3,23117E-27
RBF	Parcial	20	1	10	3,15544E-28	3,15544E-28
RBF	Parcial	21	1	10	6,10894E-27	6,36558E-27
RBF	Parcial	22	3	10	2,19324E-26	2,22732E-26

**Tabla 19:** Modelos parciales RBF (mejores resultados)

- **Backpropagation:**

Para esta red se buscan los modelos con menor error variando estos parámetros:

Los resultados obtenidos en esos experimentos son:

Red	Modelo	Caso	Iteraciones de entren.	Neuronas	Error <sub>ent</sub>	Error <sub>test</sub>
Backprop.	Parcial	1	1480	5	4,6582E-32	4,9196E-32
Backprop.	Parcial	2	4570	25	7,0628E-31	6,4631E-31
Backprop.	Parcial	3	4090	55	3,1932E-30	4,2682E-30
Backprop.	Parcial	4	790	15	2,3686E-30	3,2324E-30
Backprop.	Parcial	5	250	5	3,5663E-31	3,2836E-31
Backprop.	Parcial	6	4780	5	5,9758E-28	4,7686E-28
Backprop.	Parcial	7	150	5	3,6156E-31	3,4118E-31
Backprop.	Parcial	8	2800	5	3,9772E-31	3,5302E-31
Backprop.	Parcial	9	2810	5	8,283E-31	7,0997E-31
Backprop.	Parcial	10	200	100	1,3798E-29	1,8921E-29
Backprop.	Parcial	11	200	500	3,7931E-29	4,1198E-29
Backprop.	Parcial	12	1200	200	2,1003E-29	2,2317E-29
Backprop.	Parcial	13	1300	100	2,3015E-29	2,221E-29
Backprop.	Parcial	14	3130	5	8,3488E-31	8,4803E-31
Backprop.	Parcial	15	2000	100	1,1307E-29	1,1431E-29
Backprop.	Parcial	16	1450	5	1,7092E-30	1,6684E-30
Backprop.	Parcial	17	1440	5	6,8368E-31	8,3619E-31
Backprop.	Parcial	18	1910	5	1,6303E-30	6,4687E-31
Backprop.	Parcial	19	1300	100	4,3545E-29	4,394E-29
Backprop.	Parcial	20	1750	5	3,7865E-30	3,5341E-30
Backprop.	Parcial	21	2330	5	2,6295E-30	2,0668E-30
Backprop.	Parcial	22	1290	5	2,8136E-30	3,0923E-30

**Tabla 20:** Modelos parciales Backpropagation (mejores resultados)

Una vez obtenidos estos resultados testaremos todos los modelos de cada red. Cada modelo se testeará con los datos de todos los modelos de funcionamiento. De esta forma podremos expresar en forma de matriz los errores de test de cada modelo entrenado con los datos de cada modo de funcionamiento.

A continuación pueden verse dichas matrices:







Para facilitar la interpretación de estos resultados se procede a graficar para cada modo de funcionamiento con el que las redes han sido entrenadas el error de testeo con cada uno de los modos de funcionamiento. Estas gráficas se encuentran en los anexos de la memoria.

A diferencia que en los modelos globales, en estos podemos apreciar como la distinción entre un fallo y otro es más evidente. Por ejemplo, en los tres modelos entrenados con fallo1 el menor error de test se consigue cuando se testea frente a ese mismo fallo. Lo mismo pasa con el resto de modos de funcionamiento, de forma que si nos fijamos en las matrices de confusión, los valores de la diagonal (donde el fallo utilizado para el entrenamiento coincide con el de test) distan mucho de los demás.

Esto también puede apreciarse en las gráficas que presentan el error de test del modelo del fallo i al ser testeado con el conjunto de fallos:

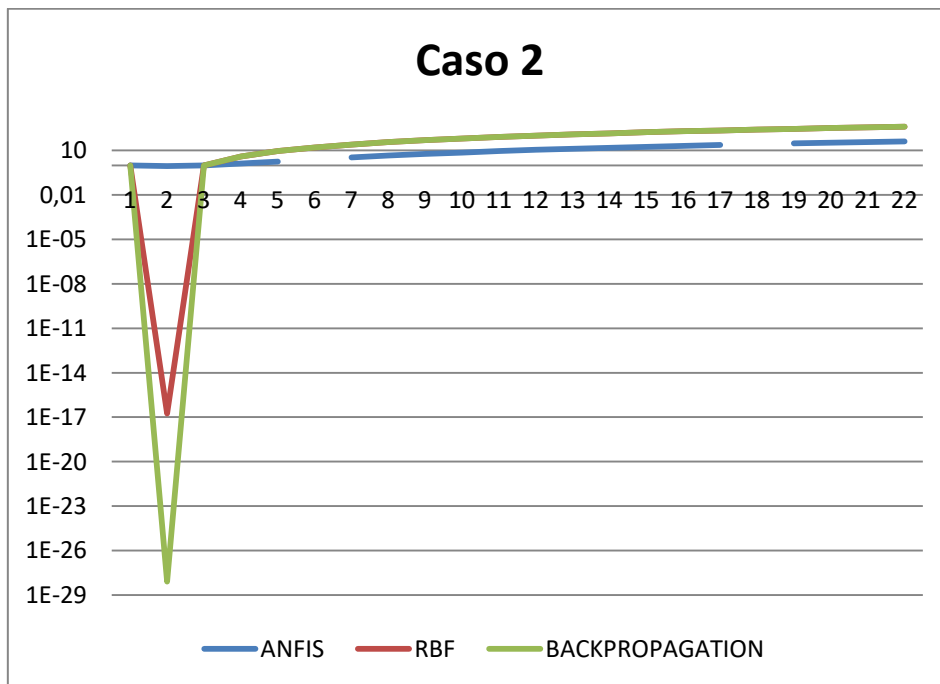
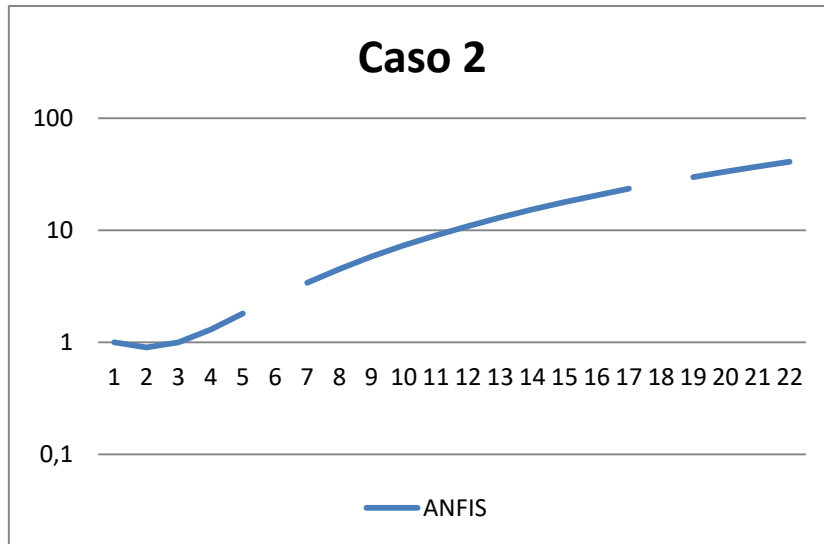


Fig 20: Error de test de modelo del caso 2 testeado con los 22 casos

Por ejemplo en esta figura podemos ver como el error de test conseguido con los modelos entrenados con el caso 2 en mucho inferior al ser testeados con el fallo 2. En gráfico esta diferencia es mucho más evidente en RBF y Backpropagation (por la escala) pero representando el modelo ANFIS sucede lo mismo.



**Fig 21:** Error de test de modelo del caso 2 testeado con los 22 casos (ANFIS)

Además es necesario indicar que para los fallos 6 y 18 con ANFIS no se han conseguido encontrar los parámetros que permitan entrenar la red.



#### 4.3.4. FDI: Toma de decisiones

Una vez se han generado los modelos, estos pasan a organizarse en 4 configuraciones distintas, tal y como se ha descrito en el Cap.4.2.2. Metodología utilizada.

En primer lugar, para el sistema formado por modelos **Anfis**, la tasa de éxito ha sido de un 90.90%, es decir ese porcentaje de los datos ha sido correctamente detectado.

Para la segunda configuración, formada por modelos **RBF** se ha medido una tasa de éxito del 100%.

La tercera configuración, formada por los 22 modelos **backpropagation** ha conseguido también una tasa de éxito del 100%, todos los datos han sido detectados de la forma correcta.

Por último la **configuración elegida**, como se ha indicado anteriormente se corresponde a la combinación de 22 modelos, uno por cada fallo. Estos modelos son aquellos que mejor detectan (proporcionando un menor error de test) el tipo de fallo para el que han sido entrenados, y que proporcionan un mayor de test ante una entrada que no corresponde a ese modo de funcionamiento. Como se ha explicado en la sección anterior este sistema coincide con el segundo, es el basado en los 22 modelos entrenados con **RBF**.

Como última fase de nuestra metodología experimental este modelo ha sido testeado con el conjunto de datos y se ha obtenido que su tasa de éxito para la detección y diagnóstico de fallos en nuestra planta del 100%.

A continuación se muestra dicho sistema:

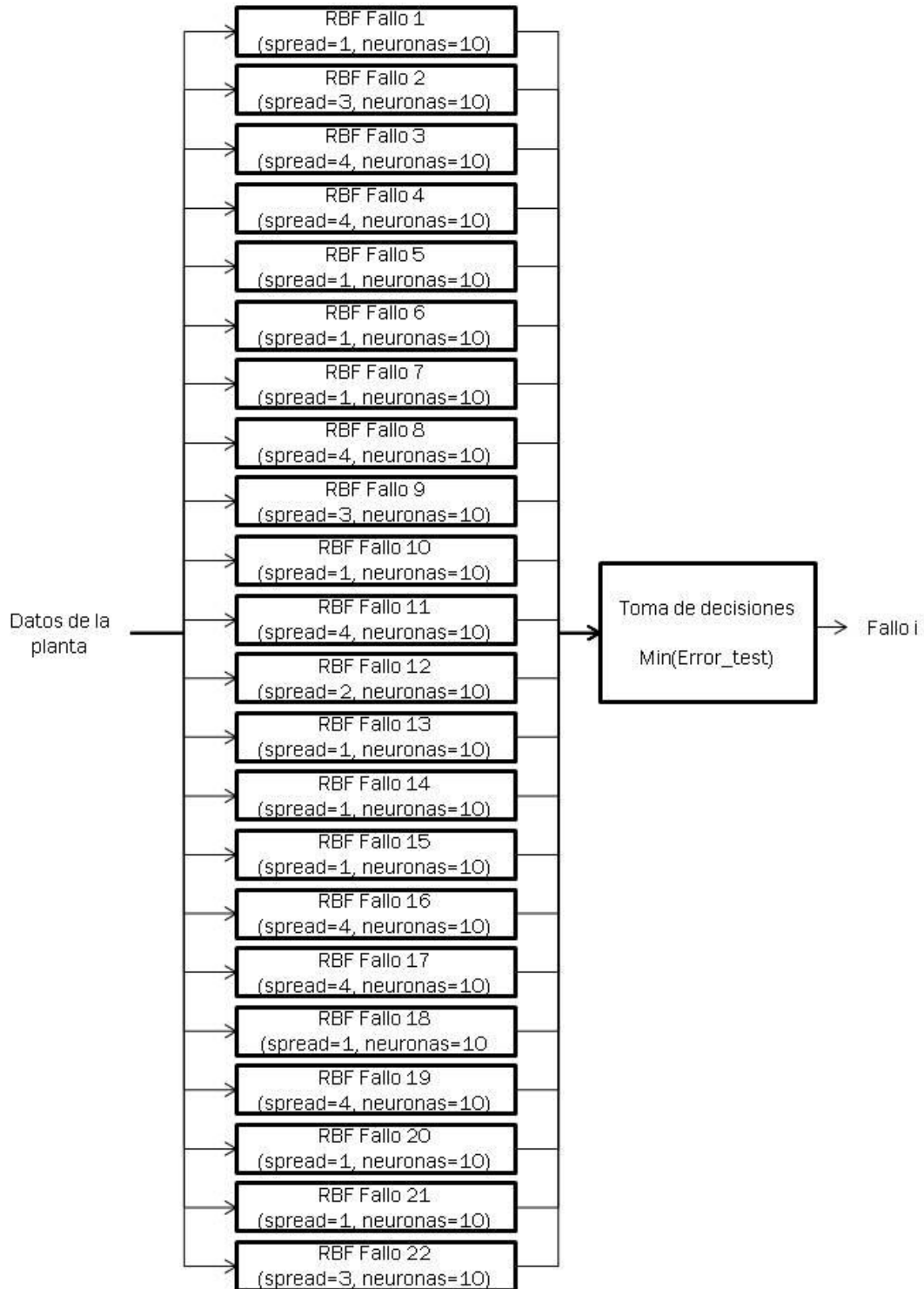


Fig 22: Sistema de detección y diagnóstico elegido

# Capítulo 5

## Conclusiones

En este capítulo se exponen cuales son las principales conclusiones extraídas en la realización del presente trabajo y los principales aprendizajes adquiridos en ella.

Para ello es necesario hacer un repaso por todos los procedimientos que se han utilizado y cuáles han sido sus resultados.

En primer lugar indicar la importancia del **procesamiento** de la muestra con el fin de poder aplicar las distintas técnicas de la misma forma a todos los datos de la misma. En un conjunto de datos pueden existir un gran número de errores que conviertan nuestro análisis en poco preciso. Para evitar esto es necesario analizar cuáles son los principales puntos débiles de nuestra muestra y utilizar las distintas técnicas existentes para mejorar los resultados.

Otro de los puntos más importantes de nuestra metodología ha sido la realización de una **selección de variables**, para facilitar el análisis de los datos y el entrenamiento de las distintas redes neuronales. El hecho de tener un conjunto de datos formado por tantas variables haría mucho más complejas las tareas posteriores, por lo que es importante una buena selección midiendo cual es la relevancia de cada una de las variables respecto del conjunto.

En cuanto a los **modelos generados**, es importante destacar el gran número de parámetros que deben variarse para encontrar el mejor modelo posible. En nuestra experimentación se han generado un gran cantidad de modelos basados en redes neuronales y neurodifusas, utilizando diferentes datos de entrada y modificando sus parámetros más relevantes. Esta es una ardua tarea que permite que cada uno de los modelos usados para el sistema se adapte lo mejor posible al problema a tratar.

En nuestro caso de estudio, se ha llegado a la conclusión de que han funcionado mejor los modelos entrenados con un solo fallo que los que han sido entrenados

con el conjunto de los modos de funcionamiento al mismo tiempo. Sin embargo esto no implica que sea así para otros casos.

Además los errores tanto de test como de entrenamiento conseguidos al utilizar las redes neuronales han sido menores que al entrenar las red neurodifusa ANFIS. En los modelos basados en ANFIS se han encontrado mayores problemas para encontrar los parámetros correctos para un entrenamiento adecuado.

En casos de estudio como este es muy importante la correcta interpretación de los resultados obtenidos. Específicamente las matrices de confusión utilizadas (mostradas en la metodología experimental) han sido muy importantes para llegar a las distintas conclusiones de este trabajo.

Respecto a las distintas redes utilizadas los mejores datos han sido los proporcionados por las redes RBF y Backpropagation (Tasa de éxito del 100%). Aunque los resultados obtenidos con los modelos entrenados con Anfis también han sido buenos, hay dos fallos (el número 6 y el 18) de los que no hemos conseguido parámetros con los que entrenar las redes. Además hay que destacar que el porcentaje de éxito con el sistema formado por modelos entrenados con RBF y Backpropagation (100%) ha sido bastante más alto que con Anfis (90%).

Por último, decir que las distintas configuraciones de modelos creadas en los últimos pasos de la metodología son capaces de hacer una detección bastante rápida que puede resultar muy útil en procesos de producción.

## Capítulo 6

### Bibliografía

Ballesteros, A. (2017). *Neural Networks Frameworks*. Recuperado el 20 de 10 de 2018, de <http://www.redes-neuronales.com.es/>

Caamano, J. (2018). *JuanPa Caamano*. Recuperado el 22 de octubre de 2018, de <http://blog.espol.edu.ec/jupacaam/2011/02/17/redes-neurodifusas/>

Colmenares, G. (s.f.). *Función de base radial*. Recuperado el 6 de 8 de 2018, de [http://www.webdelprofesor.ula.ve/economia/gcolmen/programa/redes\\_neuronales/capitulo4\\_funciones\\_bases\\_radiales.pdf](http://www.webdelprofesor.ula.ve/economia/gcolmen/programa/redes_neuronales/capitulo4_funciones_bases_radiales.pdf)

Correa-Henao, G. J., & Montoya-Suárez, L. M. (2013). Aplicación del moedlo ANFIS para predicción de series de tiempo. *Lámpsakos*, 15.

Delicado, P. (2008). *Curso de Modelos no Paramétricos*. Recuperado el 24 de octubre de 2018, de [http://www-eio.upc.es/~delicado/docencia/Apuntes\\_Models\\_No\\_Parametrics.pdf](http://www-eio.upc.es/~delicado/docencia/Apuntes_Models_No_Parametrics.pdf)

E.F.Vogel, J. (1993). *A Plant-Wide Industrial Process Control Problem, Computers and Chemical Engineering*. Recuperado el 4 de octubre de 2018, de <https://www.sciencedirect.com/science/article/pii/009813549380018I?via%3Dihub>

El Hamzaoui, Y. (2014). *Introducción a la Inteligencia Computacional y su Aplicación en la Ingeniería*. Universidad Autónoma de Cd Juárez.

Fuyuan Zhang, Z. G. (2015). Decision fusion systems for fault detection and identification in industrial processes. *Journal of Process Control* (págs. 45 -54). El sevier.

Garcia-Alvarez, D., & Fuente, M. (2011). Estudio comparativo de técnicas de detección de fallos basadas en el Análisis de Componentes Principales (PCA). *Revista Iberoamericana de Automática e Informática industrial*, 183.

- Haykin, S. (2009). *Neural Networks and Learning Machines*. Pearson.
- Hernández, C. L., & Rodríguez, J. E. (2009). Preprocesamiento de datos estructurados. *Vinculos*, 28-37.
- Herrera, F. (Mayo de 2014). *Preprocesamiento de datos*. Recuperado el 2018 de octubre de 4, de (<http://www.lsi.us.es/redmidas/Ilreunion/trans/prepro.pdf>).
- Izaurieta, F., & Saavedra, C. (s.f.). *Redes Neuronales Artificiales*. Concepción (Chile).
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer.
- Jolliffe, I. (1986). *Principal Component Analysis*. Springer.
- Jyh-Shing Roger Jang, C.-T. S. (1997). *Neuro-Fuzzy and Soft Computing: A computational approach to learning and machine intelligence*. Prentice Hall.
- Lorenzón, M. (2011). *Operadores de agregación*. Recuperado el 22 de noviembre de 2018, de [http://exa.unne.edu.ar/informatica/SO/Lorenzon\\_Matias-Operadores\\_de\\_Agregacion.pdf](http://exa.unne.edu.ar/informatica/SO/Lorenzon_Matias-Operadores_de_Agregacion.pdf)
- N.M. Al-Kandari, I. J. (2007). Variable selection and interpretation in correlation principal components., (págs. 339-354).
- Palomares, I., Liu, J., Xu, Y., & L. M. (2012). Uso de operadores OWA para modelar la actitud hacia el consenso en problemas de toma de decisión en grupo. Valladolid.
- Pueyo, R. P. (2005). *Procesado y Optimización de Espectros Raman mediante Técnicas de Lógica Difusa*. Barcelona: Universitat Politècnica de Catalunya.
- Puigjaner, L., Ollero, P., Prada, C., & Jiménez, L. (2006). *Estrategias de modelado, simulación y optimización de procesos químicos*. Síntesis.
- Rubin, D. (1987). *Inference and Missing Data*. Cambridge. Biometrika.
- Russell, E. L., Chiang, L. H., & Braatz, R. D. (2000). *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes*. Londres: Springer.
- Sánchez-Fernández, A., Sainz-Palmero, G., Benítez, J., & Fuente., M. (2018). Linguistic OWA and two time-windows based fault identification in wide plants. *Computers & Chemical Engineering*, 422-423.
- Smith, L. I. (2002). *A tutorial on Principal Components Analysis*. Recuperado el 21 de octubre de 2018, de [http://www.iro.umontreal.ca/~pift6080/H09/documents/papers/pca\\_tutorial.pdf](http://www.iro.umontreal.ca/~pift6080/H09/documents/papers/pca_tutorial.pdf)
- Tapia, D. E. (2017). *Localización Óptima de Sensores para Detectar Fugas en Sistemas de Distribución de Agua Potable*.

Valls, J. M. (s.f.). *Redes de Neuronas. Redes de Base Radial*. Recuperado el 2018 de agosto de 3, de <http://eva.evannai.inf.uc3m.es/et/docencia/rn-inf/documentacion/Tema4-RNBR.pdf>

Vasco, U. e. (2015). *Algoritmos genéticos*. Recuperado el 20 de 10 de 2018, de [www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/temageneticos.pdf](http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/temageneticos.pdf)

Villada, F., García, E., & Molina, J. D. (2011). *Pronóstico del Precio de la Energía Eléctrica usando Redes Neuro-Difusas*.

Xinwang Liu, S. H. (2008). Orness and parameterized RIM quantifier aggregation with OWA operators: A summary. *International Journal of Approximate Reasoning*, (págs. 77 -97).

Yager, R. R. (1996). Quantifier Guided Aggregation Using OWA Operators. *International journal of intelligent systems*. Nueva York.

Zapana, R. A. (2018). *Reducción de la dimensionalidad de series temporales climáticas usando Deep Multi-Layer Autoencoder*. Perú: Universidad Nacional de San Agustín.

# Índice de figuras

<i>Fig 1: Datos de partida</i>	17
<i>Fig 2: Datos referenciados a la primera componente principal</i>	17
<i>Fig 3: Estructura de red neurodifusa (1)</i>	19
<i>Fig 4: Estructura de red neurodifusa (2)</i>	19
<i>Fig 5: Estructura de una red neurodifusa ANFIS</i>	20
<i>Fig 6: Ejemplo de red monocapa</i>	22
<i>Fig 7: Ejemplo de red multicapa (Perceptrón)</i>	22
<i>Fig 8: Aprendizaje supervisado</i>	23
<i>Fig 9: Aprendizaje no supervisado</i>	23
<i>Fig 10: Estructura de una red neuronal de base radial (RBF)</i>	24
<i>Fig 11: Estructura de una red Backpropagation</i>	25
<i>Fig 12: Diagrama de flujo de la metodología propuesta</i>	30
<i>Fig 13: Esquema de modelos para la detección de fallos</i>	38
<i>Fig 14: Esquema del proceso Tennessee Eastman</i>	42
<i>Fig 15: Esquema de sistema de FDI con modelos ANFIS</i>	55
<i>Fig 16: Esquema de sistema de FDI con modelos RBF</i>	56
<i>Fig 17: Esquema de sistema de FDI con modelos Backpropagation</i>	57
<i>Fig 18: Ejemplo de datos sin normalizar (variable 4)</i>	61
<i>Fig 19: Ejemplo de datos normalizados (variable 4)</i>	61
<i>Fig 20: Error de test de modelo del caso 2 testeado con los 22 casos</i>	79
<i>Fig 21: Error de test de modelo del caso 2 testeado con los 22 casos (ANFIS)</i>	80
<i>Fig 22: Sistema de detección y diagnosis elegido</i>	82
<i>Fig 23: Porcentaje de veces que el modelo i detecta datos del caso 1</i>	100
<i>Fig 24: Porcentaje de veces que el modelo i detecta datos del caso 2</i>	100
<i>Fig 25: Porcentaje de veces que el modelo i detecta datos del caso 3</i>	101
<i>Fig 26: Porcentaje de veces que el modelo i detecta datos del caso 4</i>	101
<i>Fig 27: Porcentaje de veces que el modelo i detecta datos del caso 5</i>	101
<i>Fig 28: Porcentaje de veces que el modelo i detecta datos del caso 6</i>	102
<i>Fig 29: Porcentaje de veces que el modelo i detecta datos del caso 7</i>	102
<i>Fig 30: Porcentaje de veces que el modelo i detecta datos del caso 8</i>	102
<i>Fig 31: Porcentaje de veces que el modelo i detecta datos del caso 9</i>	103
<i>Fig 32: Porcentaje de veces que el modelo i detecta datos del caso 10</i>	103
<i>Fig 33: Porcentaje de veces que el modelo i detecta datos del caso 11</i>	103
<i>Fig 34: Porcentaje de veces que el modelo i detecta datos del caso 12</i>	104
<i>Fig 35: Porcentaje de veces que el modelo i detecta datos del caso 13</i>	104
<i>Fig 36: Porcentaje de veces que el modelo i detecta datos del caso 14</i>	104
<i>Fig 37: Porcentaje de veces que el modelo i detecta datos del caso 15</i>	105
<i>Fig 38: Porcentaje de veces que el modelo i detecta datos del caso 16</i>	105
<i>Fig 39: Porcentaje de veces que el modelo i detecta datos del caso 17</i>	105
<i>Fig 40: Porcentaje de veces que el modelo i detecta datos del caso 18</i>	106



<i>Fig 41: Porcentaje de veces que el modelo i detecta datos del caso 19 .....</i>	<i>106</i>
<i>Fig 42: Porcentaje de veces que el modelo i detecta datos del caso 20 .....</i>	<i>106</i>
<i>Fig 43: Porcentaje de veces que el modelo i detecta datos del caso 21 .....</i>	<i>107</i>
<i>Fig 44: Porcentaje de veces que el modelo i detecta datos del caso 22 .....</i>	<i>107</i>
<i>Fig 45: Error de test de modelo del caso 1 testeado con los 22 casos.....</i>	<i>108</i>
<i>Fig 46: Error de test de modelo del caso 2 testeado con los 22 casos.....</i>	<i>108</i>
<i>Fig 47: Error de test de modelo del caso 3 testeado con los 22 casos.....</i>	<i>109</i>
<i>Fig 48: Error de test de modelo del caso 4 testeado con los 22 casos.....</i>	<i>109</i>
<i>Fig 49: Error de test de modelo del caso 5 testeado con los 22 casos.....</i>	<i>109</i>
<i>Fig 50: Error de test de modelo del caso 6 testeado con los 22 casos.....</i>	<i>110</i>
<i>Fig 51: Error de test de modelo del caso 7 testeado con los 22 casos.....</i>	<i>110</i>
<i>Fig 52: Error de test de modelo del caso 8 testeado con los 22 casos.....</i>	<i>110</i>
<i>Fig 53: Error de test de modelo del caso 9 testeado con los 22 casos.....</i>	<i>111</i>
<i>Fig 54: Error de test de modelo del caso 10 testeado con los 22 casos.....</i>	<i>111</i>
<i>Fig 55: Error de test de modelo del caso 11 testeado con los 22 casos.....</i>	<i>111</i>
<i>Fig 56: Error de test de modelo del caso 12 testeado con los 22 casos.....</i>	<i>112</i>
<i>Fig 57: Error de test de modelo del caso 13 testeado con los 22 casos.....</i>	<i>112</i>
<i>Fig 58: Error de test de modelo del caso 14 testeado con los 22 casos.....</i>	<i>112</i>
<i>Fig 59: Error de test de modelo del caso 15 testeado con los 22 casos.....</i>	<i>113</i>
<i>Fig 60: Error de test de modelo del caso 16 testeado con los 22 casos.....</i>	<i>113</i>
<i>Fig 61: Error de test de modelo del caso 17 testeado con los 22 casos.....</i>	<i>113</i>
<i>Fig 62: Error de test de modelo del caso 18 testeado con los 22 casos.....</i>	<i>114</i>
<i>Fig 63: Error de test de modelo del caso 19 testeado con los 22 casos.....</i>	<i>114</i>
<i>Fig 64: Error de test de modelo del caso 20 testeado con los 22 casos.....</i>	<i>114</i>
<i>Fig 65: Error de test de modelo del caso 21 testeado con los 22 casos.....</i>	<i>115</i>
<i>Fig 66: Error de test de modelo del caso 22 testeado con los 22 casos.....</i>	<i>115</i>



# Índice de tablas

<b>Tabla 1:</b> Variables de la planta .....	43
<b>Tabla 2:</b> Fallos de la planta.....	44
<b>Tabla 3:</b> Parametros modelos parciales ANFIS (N° épocas=10:10:1000).....	51
<b>Tabla 4:</b> 4° sistema de modelos para FDI usando ANFIS, RBF y Backpropag.....	59
<b>Tabla 5:</b> Variabilidad aportada por cada PC.....	62
<b>Tabla 6:</b> PC según la variabilidad conservada (Datos entrenamiento).....	63
<b>Tabla 7:</b> Aumento de PC al aumentar la variabilidad (Datos entrenamiento).....	63
<b>Tabla 8:</b> PC según la variabilidad conservada (Datos test).....	64
<b>Tabla 9:</b> Aumento de PC al aumentar la variabilidad (Datos test) .....	65
<b>Tabla 10:</b> Variables relevantes modelos globales (orness=0.6) .....	66
<b>Tabla 11:</b> Variables relevantes para cada fallo (orness=0.6).....	66
<b>Tabla 12:</b> Mejor resultado para modelos globales Anfis.....	67
<b>Tabla 13:</b> Mejor resultado para modelos globales RBF .....	68
<b>Tabla 14:</b> Mejor resultado para modelos globales Backpropagation.....	68
<b>Tabla 15:</b> Mejor modelo global ANFIS fallo i testeado fallo j (R=0,56, ép.=10).....	70
<b>Tabla 16:</b> Mejor modelo global RBF fallo i testeado fallo j (Spr=3, neur.=460) .....	71
<b>Tabla 17:</b> Mejor modelo global Backp. fallo i testeado fallo j (lt.=5, neu.=370).....	72
<b>Tabla 18:</b> Modelos parciales ANFIS (mejores resultados) .....	73
<b>Tabla 19:</b> Modelos parciales RBF (mejores resultados).....	74
<b>Tabla 20:</b> Modelos parciales Backpropagation (mejores resultados).....	75
<b>Tabla 21:</b> Mejor modelo parcial ANFIS fallo i testeado con el fallo j.....	76
<b>Tabla 22:</b> Mejor modelo parcial RBF fallo i testeado con el fallo j.....	77
<b>Tabla 23:</b> Mejor modelo parcial Backpropagation fallo i testeado con el fallo j .....	78
<b>Tabla 27:</b> Importancia relativa de cada variable en los datos de entrenamiento ...	94
<b>Tabla 28:</b> Importancia relativa de cada variable en datos de test.....	96
<b>Tabla 29:</b> Ranking de variables para modelos globales.....	97
<b>Tabla 30:</b> Ranking de variables para modelos parciales .....	99

# Anexo A

## Tablas de relevancia de las variables

En Las siguientes tablas se pueden encontrar tanto para los ficheros de test como para los ficheros de entrenamiento la relevancia (en porcentaje) de cada una de las 52 variables para cada uno de los modos de funcionamiento.

Esta relevancia se ha calculado con la matriz de cargas obtenida en el análisis de componentes principales (PCA).

$$R_i = \frac{\sum_{j=1}^n \alpha_{ij}}{\sum_{k=1}^n (\sum_{j=1}^n \alpha_{kj})} \quad (\text{Ec. 28})$$

Para conocer más sobre cómo se ha calculado esta relevancia puede acudir al capítulo 3 de esta memoria.

Datos de entrenamiento (%)																						
	D01	D02	D03	D04	D05	D06	D07	D08	D09	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D00
1	3,215	2,330	1,147	1,493	2,984	2,761	3,185	2,815	1,362	1,408	1,613	2,991	2,663	0,849	1,502	1,446	1,208	2,830	0,802	1,500	1,535	1,554
2	3,024	2,216	1,279	1,554	2,904	2,535	2,653	2,923	1,404	1,491	1,338	2,573	2,520	1,504	1,645	1,197	1,553	2,428	1,523	1,443	1,315	1,523
3	1,756	1,436	2,219	1,424	1,262	2,306	1,193	2,426	1,293	2,506	1,603	2,307	2,947	1,556	1,581	1,379	1,598	2,012	1,406	1,941	1,682	1,899
4	1,587	1,479	1,760	1,573	1,863	2,233	1,476	2,918	1,798	2,187	1,462	2,721	2,564	1,249	1,764	1,760	1,519	1,985	1,569	2,120	1,791	1,706
5	2,840	1,875	2,320	2,233	2,113	2,660	2,613	1,363	1,713	1,732	1,675	2,048	1,298	1,628	1,638	1,758	2,383	2,069	1,793	2,682	1,913	1,839
6	2,100	1,767	2,200	1,967	1,680	2,081	1,995	2,359	1,660	1,830	1,647	2,571	1,730	1,880	1,686	1,523	2,175	1,668	1,442	1,896	1,992	1,663
7	1,709	1,766	1,694	1,844	2,281	1,795	1,911	2,766	1,775	1,741	2,157	2,644	2,603	1,840	1,612	1,753	2,017	2,216	1,838	1,403	2,550	1,715
8	2,030	1,888	1,897	1,793	2,327	2,006	1,562	2,328	2,016	1,953	1,811	2,230	2,296	1,924	1,947	1,654	1,766	1,885	1,840	2,119	2,001	2,008
9	2,210	1,598	1,953	1,943	2,097	1,776	1,947	2,197	2,184	2,415	1,950	1,682	2,086	2,132	1,613	2,062	2,292	1,657	1,671	1,689	2,243	2,152
10	2,478	2,066	2,012	1,824	2,438	2,365	1,920	1,844	1,936	1,991	1,857	1,646	2,149	1,812	1,923	2,190	1,823	2,167	1,886	2,042	2,019	1,661
11	2,888	2,179	1,972	1,820	1,854	2,017	2,476	2,200	2,181	1,779	2,119	1,574	2,076	2,124	1,594	1,928	1,710	2,202	2,005	2,243	2,488	1,707
12	2,038	2,151	1,304	1,477	1,737	2,394	1,942	1,548	1,649	2,459	1,848	1,853	2,010	2,261	1,922	2,291	1,984	2,171	1,923	2,046	1,887	1,818
13	2,211	2,151	1,784	1,507	2,344	2,323	1,909	1,712	2,132	1,954	1,478	2,437	2,332	2,111	1,849	1,483	1,874	2,269	1,768	2,489	1,778	1,824
14	2,074	1,864	2,544	1,541	2,660	2,409	2,137	2,864	3,204	2,120	1,644	1,672	2,140	1,727	2,580	3,043	1,978	2,008	1,983	2,372	1,996	1,821
15	1,762	2,156	2,270	2,054	1,731	2,157	1,932	1,709	1,927	2,701	3,078	2,431	2,276	1,757	2,000	2,067	1,624	1,944	2,232	2,760	1,451	2,041
16	2,442	3,124	2,575	2,217	2,248	2,373	1,983	2,045	2,031	1,879	2,352	1,793	1,727	3,012	2,470	2,109	3,150	2,026	2,426	1,836	1,754	2,243
17	2,188	2,741	2,664	2,237	2,736	2,347	2,284	2,465	2,180	2,686	2,389	2,425	1,944	2,317	1,462	2,151	2,567	2,246	2,538	2,554	1,679	2,034
18	2,128	2,474	1,976	2,142	1,772	2,173	2,682	2,052	1,980	2,218	2,688	1,944	2,098	1,995	2,704	2,202	2,220	2,059	2,570	2,543	2,143	2,170
19	2,267	2,096	2,865	2,381	2,019	2,172	2,511	1,816	2,722	1,856	1,912	2,500	2,729	2,574	2,568	2,765	2,418	2,289	2,128	1,944	2,235	2,697
20	1,985	2,649	2,601	2,638	1,950	2,027	2,042	2,547	1,835	2,313	1,990	1,959	1,891	2,485	2,128	2,503	1,759	1,966	2,360	1,938	2,072	2,793
21	2,069	1,836	1,926	2,755	2,384	1,845	2,380	1,342	2,291	2,556	2,536	2,144	1,865	2,409	3,032	2,362	2,349	2,172	2,392	2,343	2,617	2,278
22	2,388	2,264	2,221	2,568	2,090	1,897	1,907	1,955	2,357	2,713	2,570	1,816	2,541	2,466	2,206	2,391	2,716	2,241	2,606	2,529	2,711	2,583
23	2,193	2,412	2,548	2,368	2,051	2,438	2,679	2,658	2,187	1,803	2,408	2,142	1,942	2,261	2,588	2,400	2,348	2,102	2,482	2,371	2,294	2,599
24	2,130	2,230	2,147	2,799	1,764	2,397	2,499	2,220	2,580	2,320	2,445	2,012	2,246	2,604	2,652	2,250	2,182	2,303	2,071	2,033	1,959	2,158
25	2,056	2,787	2,124	2,583	1,894	2,246	2,389	2,139	2,869	2,440	2,358	2,234	2,470	2,916	2,484	2,542	2,198	2,393	2,507	2,409	2,648	2,278
26	2,122	1,868	2,579	1,997	1,985	2,120	2,171	2,203	2,039	2,521	2,660	2,444	2,293	2,337	2,400	2,642	2,188	2,008	2,567	2,265	2,175	2,194
27	2,268	2,590	2,182	2,677	2,466	1,976	1,859	2,185	2,421	2,430	1,848	2,004	2,250	2,561	1,987	1,865	2,771	2,327	2,253	2,386	2,150	2,368

28	2,145	2,465	2,497	2,006	2,149	2,036	2,337	1,992	2,157	2,027	2,180	2,200	2,245	2,151	2,118	2,493	2,138	1,947	2,353	2,179	2,336	2,050
29	2,106	2,087	2,045	2,131	2,625	1,572	2,040	2,317	2,238	2,433	2,452	2,375	2,020	2,238	2,716	2,281	2,292	2,230	1,934	2,253	2,534	2,331
30	1,838	1,883	2,618	1,928	2,193	1,844	2,463	2,294	1,869	2,489	2,040	2,059	2,249	2,348	2,314	2,380	1,718	2,221	1,896	2,283	1,978	2,301
31	2,393	2,214	2,395	2,354	1,570	2,274	1,982	1,994	2,528	2,367	2,040	2,323	1,966	2,013	1,880	2,190	2,363	1,616	2,267	2,062	2,218	2,156
32	1,771	2,148	2,119	2,407	1,845	1,958	1,823	1,876	1,968	2,248	2,410	1,917	1,959	2,084	2,395	1,693	2,461	2,113	2,503	2,317	2,577	2,106
33	1,962	2,174	2,045	2,052	1,768	2,034	1,947	2,193	1,938	2,140	2,177	2,006	1,732	2,207	2,190	2,008	1,980	2,174	2,191	2,009	2,313	2,160
34	2,148	2,502	2,539	2,887	2,606	1,982	2,252	2,379	1,937	2,132	2,171	1,674	1,708	2,140	2,460	2,207	2,211	2,123	2,307	2,011	2,455	2,669
35	1,811	1,840	1,713	2,371	2,292	2,008	2,274	1,840	2,512	2,170	2,047	2,292	2,234	2,299	2,119	2,333	2,565	2,180	2,508	2,334	2,543	2,112
36	1,875	2,078	2,029	1,995	2,147	1,909	2,125	1,892	2,714	2,179	1,972	2,105	1,964	2,145	2,268	2,181	2,001	2,216	2,186	2,461	2,156	2,516
37	1,655	1,792	1,894	2,042	1,716	2,060	2,247	1,854	1,973	1,783	2,466	2,185	2,012	2,190	1,860	2,730	1,624	1,933	2,339	1,889	2,459	1,801
38	2,082	1,984	2,246	2,513	2,051	1,953	2,202	1,634	2,328	2,193	2,506	1,792	1,618	2,033	2,515	2,049	2,203	2,033	2,061	1,738	1,397	2,424
39	2,147	1,519	1,730	1,118	2,139	1,754	1,824	2,111	1,912	1,715	1,833	1,327	1,965	2,308	1,454	2,295	2,428	1,912	1,923	1,887	2,182	1,579
40	1,812	1,971	1,937	2,302	2,025	1,974	1,808	1,783	1,678	1,639	2,366	1,933	1,865	1,355	1,897	1,834	1,618	1,989	2,354	1,849	2,428	1,981
41	1,576	2,188	2,212	1,833	1,697	1,565	1,793	1,938	2,091	1,921	1,455	1,923	1,248	2,268	2,273	1,783	1,793	1,444	1,805	1,277	1,380	2,213
42	1,021	1,159	1,169	1,645	1,555	1,489	1,192	1,314	1,145	2,121	1,950	1,283	1,681	1,220	1,381	1,964	2,210	1,648	2,123	1,840	1,517	1,106
43	1,577	1,439	1,273	1,516	1,250	1,707	1,712	1,493	1,656	0,887	1,905	1,727	1,743	1,541	2,010	2,271	1,987	1,761	2,001	1,104	1,915	1,578
44	1,429	1,909	1,789	2,014	1,738	1,484	1,170	1,596	2,084	1,285	1,697	1,720	1,579	2,076	1,467	1,039	1,212	1,746	1,018	1,703	1,744	1,911
45	1,189	1,601	1,756	1,685	1,641	1,626	1,710	1,374	1,761	1,387	1,528	1,621	1,563	1,743	1,826	1,294	1,728	1,708	1,531	1,555	1,454	1,678
46	1,566	1,148	1,161	1,227	1,590	1,274	1,149	1,336	1,219	1,205	1,219	1,198	1,152	1,264	1,186	1,356	1,206	1,595	1,774	1,124	1,037	1,395
47	0,888	0,913	1,071	1,076	1,118	0,974	0,837	0,876	1,085	0,903	1,038	0,928	0,833	1,005	0,989	0,931	1,021	1,072	1,536	1,450	1,394	1,013
48	1,137	1,218	1,350	1,257	0,888	1,091	1,137	1,099	1,319	1,148	1,237	0,812	0,951	1,360	1,291	1,189	1,256	1,085	1,093	0,890	1,076	1,352
49	0,869	0,949	0,976	1,020	1,075	0,805	0,914	0,843	1,027	0,947	0,992	1,036	1,141	0,983	1,084	0,969	1,038	1,017	1,027	1,150	0,865	1,006
50	0,802	0,845	0,781	0,807	0,884	0,806	0,812	0,875	0,842	0,810	0,777	0,776	0,838	0,864	0,857	0,791	0,850	0,745	0,815	0,852	1,152	0,844
51	1,020	0,991	0,946	1,200	0,902	0,995	1,007	0,747	1,148	0,932	1,054	0,996	1,024	0,938	0,956	1,012	0,864	0,925	0,936	0,942	1,152	1,199
52	1,023	0,991	0,948	1,203	0,900	0,996	1,003	0,747	1,148	0,934	1,057	0,994	1,023	0,936	0,955	1,012	0,865	0,924	0,936	0,945	0,660	1,194

**Tabla 24:** Importancia relativa de cada variable en los datos de entrenamiento

Datos de test (%)																						
	D01	D02	D03	D04	D05	D06	D07	D08	D09	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D00
1	3,270	2,241	1,485	1,173	3,265	2,647	3,276	2,941	1,271	1,374	1,169	3,058	2,935	0,860	1,173	1,514	1,167	3,027	0,850	1,434	1,923	1,460
2	3,202	1,432	1,469	1,177	1,895	2,799	2,810	2,974	1,302	1,546	1,272	2,617	2,830	1,374	1,242	1,368	1,281	2,338	1,521	1,220	2,469	1,430
3	1,322	1,281	1,680	1,628	2,379	2,545	1,394	2,385	1,917	3,198	1,458	2,224	2,564	1,371	1,621	2,863	1,549	2,669	1,436	1,698	2,197	1,740
4	2,021	2,612	1,438	1,608	2,763	1,702	1,394	3,027	1,901	2,125	1,820	2,829	2,687	1,371	1,736	1,830	1,475	1,560	1,348	1,507	1,440	1,705
5	2,806	1,963	1,670	1,582	1,944	1,945	2,472	1,332	1,568	2,492	1,795	2,271	2,639	1,756	1,581	2,258	1,921	2,267	1,498	2,046	1,374	1,558
6	1,499	1,624	1,473	1,728	1,924	2,042	1,853	2,141	1,591	1,844	1,577	2,051	1,482	1,627	1,695	1,627	2,171	1,626	1,533	1,462	1,663	1,954
7	1,927	1,813	1,886	2,059	1,836	2,045	1,741	2,839	1,813	2,017	1,732	2,309	1,521	1,760	1,940	2,149	1,709	2,151	1,950	2,176	1,553	1,976
8	1,995	1,804	2,258	2,031	1,810	2,647	2,450	2,998	1,827	2,756	2,403	1,335	2,367	2,082	2,048	2,422	1,831	2,008	1,732	2,317	1,920	1,816
9	1,432	2,077	1,826	1,978	2,097	1,797	1,984	1,752	2,375	1,722	2,140	1,872	1,545	2,156	2,044	2,629	1,856	2,472	1,958	2,701	1,801	1,943
10	1,894	1,958	1,795	2,012	2,046	2,344	1,786	2,010	1,882	2,234	2,303	1,691	1,703	1,708	1,838	1,995	2,318	1,956	1,786	2,309	2,028	1,628
11	2,110	2,228	2,048	1,645	1,929	2,130	2,084	1,964	2,301	2,054	1,950	2,547	1,873	1,737	1,760	1,993	2,122	1,932	1,970	2,248	1,669	1,882
12	2,180	1,941	2,049	1,978	1,936	2,770	2,174	1,687	2,224	2,648	2,254	2,489	1,286	1,996	2,542	2,522	2,371	1,796	1,659	2,769	1,750	1,492
13	1,999	2,363	1,441	1,748	2,278	1,834	1,946	2,557	2,013	2,152	1,732	2,017	2,135	2,166	1,475	2,052	2,015	1,796	1,184	2,620	1,737	1,807
14	2,199	2,189	3,235	2,724	2,318	2,411	1,953	2,396	2,956	1,837	1,506	2,181	2,452	1,950	2,744	2,116	2,802	2,436	2,111	2,109	1,811	2,917
15	2,277	1,882	2,627	2,329	2,100	2,609	2,210	1,715	2,955	2,321	3,179	2,343	2,135	1,850	2,411	2,583	2,065	1,858	2,268	2,432	1,302	2,476
16	2,189	2,826	2,088	2,494	2,527	1,948	1,984	1,968	1,888	2,475	2,742	2,161	1,774	1,862	2,354	2,431	3,023	1,914	2,925	2,943	1,599	2,695
17	2,841	2,792	2,451	2,067	2,317	2,085	2,620	2,242	2,362	1,575	2,085	2,051	2,114	2,246	2,694	1,720	3,093	2,802	2,468	2,000	1,878	2,077
18	2,276	2,997	1,950	2,929	2,383	1,597	2,312	2,251	1,507	1,913	2,368	2,115	2,623	3,163	2,215	2,355	2,071	2,051	2,607	2,086	2,237	2,051
19	2,309	1,647	2,331	2,579	2,061	1,808	2,389	2,124	2,506	2,201	2,104	2,348	2,470	2,531	2,605	1,697	2,173	2,261	2,552	1,793	2,851	2,196
20	1,774	2,066	2,106	2,266	2,093	2,504	1,958	2,514	2,096	2,703	1,977	2,108	2,065	2,702	1,789	2,685	1,909	1,975	1,938	1,793	2,160	2,450
21	2,451	1,909	1,921	2,438	2,144	2,218	2,179	1,808	2,278	2,165	2,508	1,693	2,239	2,291	2,244	2,429	1,833	1,938	2,571	2,307	2,348	2,370
22	2,378	2,157	2,574	2,146	2,050	2,106	2,076	2,410	2,170	2,305	2,134	2,048	2,434	2,910	2,460	2,366	2,449	2,079	2,294	2,442	2,637	2,542
23	1,725	2,294	2,357	2,589	2,300	2,000	2,329	2,425	2,636	2,433	2,240	2,083	2,679	2,512	2,408	2,448	2,117	1,965	2,284	2,141	2,527	2,410
24	2,028	2,379	2,286	1,936	2,029	2,376	2,776	2,387	2,157	2,763	2,240	2,432	2,540	2,715	2,467	2,293	2,288	2,318	2,336	2,239	2,055	2,464
25	2,306	2,266	2,488	2,217	2,373	1,298	2,126	2,370	2,534	2,342	2,180	2,483	2,272	2,669	2,501	2,171	2,284	2,053	2,588	1,930	2,727	2,488
26	2,181	2,311	2,153	2,572	1,582	2,304	2,454	2,274	1,592	2,081	2,514	2,301	2,280	2,199	1,921	2,065	2,511	2,278	2,784	2,375	2,486	2,014

27	1,994	2,455	2,417	2,630	1,988	1,946	1,938	1,975	1,841	1,966	2,318	2,440	2,549	2,739	2,514	2,426	1,937	2,368	2,219	2,067	2,731	2,013
28	2,233	2,671	2,181	2,368	2,610	1,904	2,396	2,335	1,994	2,242	2,253	2,858	2,399	2,204	2,320	2,098	1,818	1,825	2,236	2,405	2,555	2,431
29	2,013	2,123	2,400	2,329	2,081	2,032	2,183	2,137	1,751	2,635	2,401	2,138	2,086	2,406	2,555	2,479	2,086	1,716	2,298	2,069	1,716	2,417
30	1,965	2,070	2,630	2,266	2,078	2,120	2,095	2,129	1,854	1,826	2,248	2,192	1,913	2,044	1,950	2,377	2,447	2,024	1,815	2,402	2,347	2,074
31	2,210	2,026	2,189	1,818	1,662	2,039	2,260	1,447	2,512	2,166	2,340	1,719	2,063	2,515	2,192	1,610	2,351	1,896	2,156	2,010	1,618	1,644
32	1,777	2,147	1,742	2,324	2,076	2,388	1,815	1,947	2,382	2,134	1,991	1,919	2,108	1,917	1,814	1,891	2,058	1,959	2,487	1,790	2,638	2,436
33	1,992	1,768	2,411	2,178	1,703	1,878	1,533	1,910	2,214	1,441	2,048	2,385	1,950	1,613	2,466	2,323	2,047	2,151	1,778	2,053	2,847	2,386
34	2,194	2,327	2,754	2,370	2,149	2,329	2,417	2,093	2,778	1,914	1,951	1,633	2,013	2,163	2,422	2,033	1,976	1,945	2,156	2,282	2,161	2,692
35	2,500	2,270	2,068	2,017	2,245	1,731	2,203	2,037	2,268	2,076	2,180	1,745	1,940	2,369	2,322	2,236	2,555	1,904	2,553	2,140	2,181	2,167
36	1,752	1,706	2,031	2,551	2,184	1,762	2,224	1,893	2,117	2,348	2,720	1,939	1,776	2,629	2,208	2,108	2,195	2,306	2,667	2,565	1,979	1,935
37	1,709	2,194	1,967	1,925	1,815	2,260	1,759	1,752	2,094	2,004	2,141	1,489	2,164	2,025	1,906	1,593	1,716	1,895	2,280	2,112	2,454	1,728
38	2,159	1,990	2,390	2,222	2,045	1,413	2,023	1,641	2,280	1,913	1,861	1,975	1,394	1,853	2,364	1,862	2,138	1,953	2,158	1,915	2,360	2,202
39	2,170	2,126	1,833	1,424	1,907	2,178	2,136	1,941	1,748	1,767	2,394	1,550	1,632	2,228	1,892	1,842	2,355	2,104	1,816	1,849	2,380	2,041
40	1,160	1,713	1,944	1,947	1,980	2,262	1,904	1,761	2,028	1,572	1,799	1,643	1,801	1,334	1,984	1,423	1,927	2,248	2,422	1,843	1,981	1,882
41	1,873	2,000	2,354	2,302	2,055	1,685	1,935	1,539	2,224	1,910	1,578	1,873	1,964	2,288	2,259	1,850	1,484	1,674	1,772	1,697	1,259	2,312
42	1,426	0,852	1,123	0,979	1,764	1,267	1,661	1,472	1,105	2,114	2,102	1,604	0,981	1,039	0,999	1,877	2,247	1,530	2,213	1,759	1,782	1,111
43	1,689	1,638	1,590	1,311	1,116	1,875	1,464	1,567	1,820	0,970	1,885	1,348	1,466	1,434	1,928	2,201	1,974	1,880	1,712	1,157	2,125	1,944
44	1,351	1,885	1,998	1,893	1,019	1,826	1,088	1,183	1,962	1,475	1,246	1,577	1,616	2,025	1,435	0,972	1,170	1,611	1,229	1,652	1,754	1,851
45	1,687	1,630	1,845	1,701	1,430	1,713	1,237	0,952	1,678	1,507	1,677	1,629	1,274	1,791	1,613	1,044	1,665	1,789	1,663	1,183	1,425	1,724
46	1,792	1,173	1,141	1,251	1,484	1,318	1,543	1,484	1,158	1,116	1,194	0,854	1,374	1,293	1,200	1,268	1,024	1,366	1,288	1,556	1,053	1,289
47	0,849	0,911	0,995	1,029	1,147	1,014	0,858	0,831	1,032	0,932	0,931	1,077	0,799	1,050	0,964	0,868	0,962	1,336	1,540	1,221	1,318	1,016
48	1,175	1,208	1,284	1,288	0,949	1,178	1,128	1,170	1,294	1,210	1,215	0,844	1,250	1,414	1,299	1,125	1,274	1,503	1,065	0,941	1,077	1,268
49	0,951	0,945	0,984	0,984	1,115	0,622	0,916	0,899	1,042	0,889	1,009	1,080	1,231	1,062	0,996	0,921	0,966	0,991	1,032	1,115	0,888	1,003
50	0,859	0,837	0,851	0,857	0,868	0,878	0,757	0,841	0,857	0,801	0,851	0,814	0,851	0,864	0,824	0,757	0,787	0,755	0,811	0,871	1,286	0,827
51	0,965	1,141	0,899	1,202	1,077	0,933	0,897	0,785	1,171	0,897	1,141	1,009	0,882	1,068	1,033	1,120	1,220	0,873	1,238	1,123	1,282	1,031
52	0,965	1,141	0,897	1,199	1,075	0,936	0,899	0,785	1,172	0,899	1,143	1,008	0,882	1,072	1,034	1,119	1,216	0,875	1,243	1,122	0,655	1,032

*Tabla 25: Importancia relativa de cada variable en datos de test*



## Anexo B

### Rankings de variables

En las siguientes tablas se puede encontrar el ranking de variables con mayor relevancia. La primera tabla tiene en cuenta los datos referentes a todos los modos de funcionamiento, y la segunda cada uno de los fallos de forma individual.

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

Tabla 26: Ranking de variables para modelos globales





# Anexo C

## Gráficas de detección de fallos en modelos globales

En las siguientes graficas se presenta el porcentaje de veces para el que un modelo entrenado con cada fallo detecta el fallo indicado.

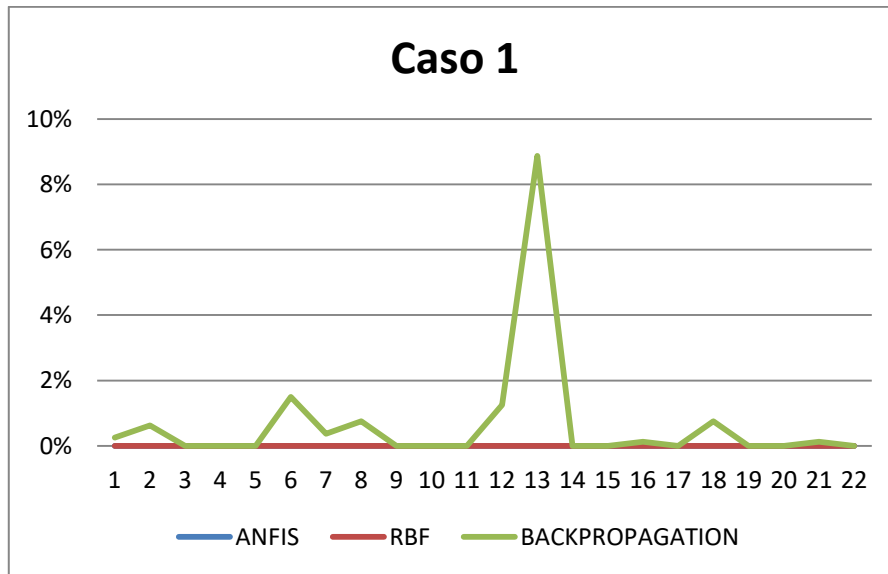


Fig 23: Porcentaje de veces que el modelo i detecta datos del caso 1

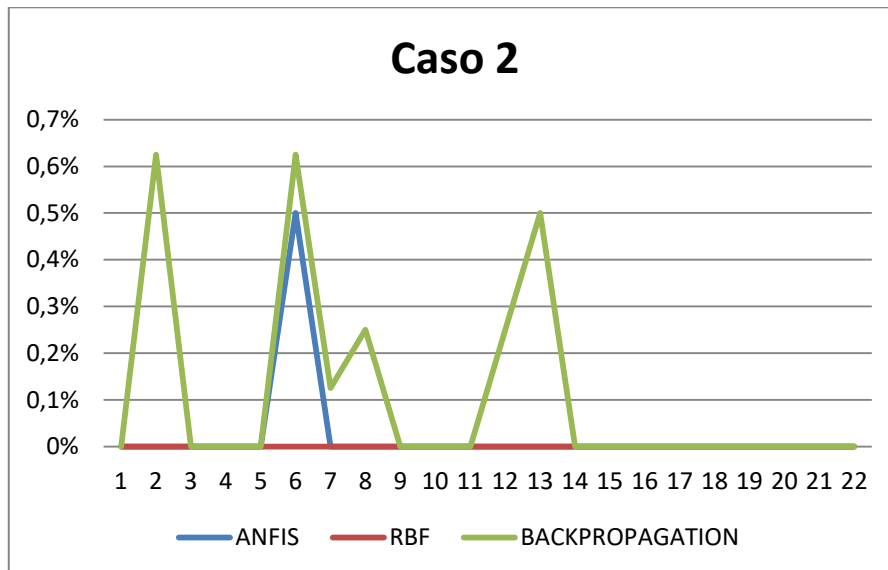
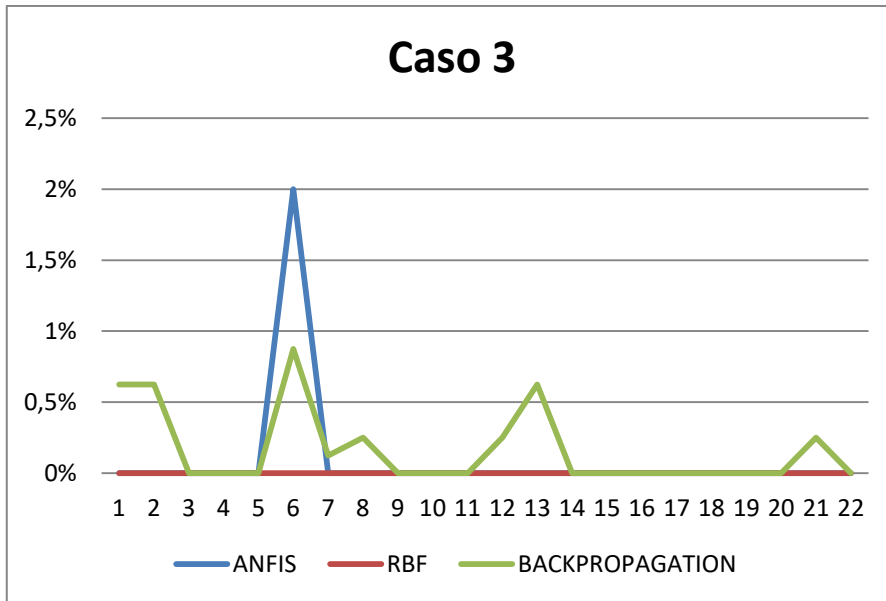
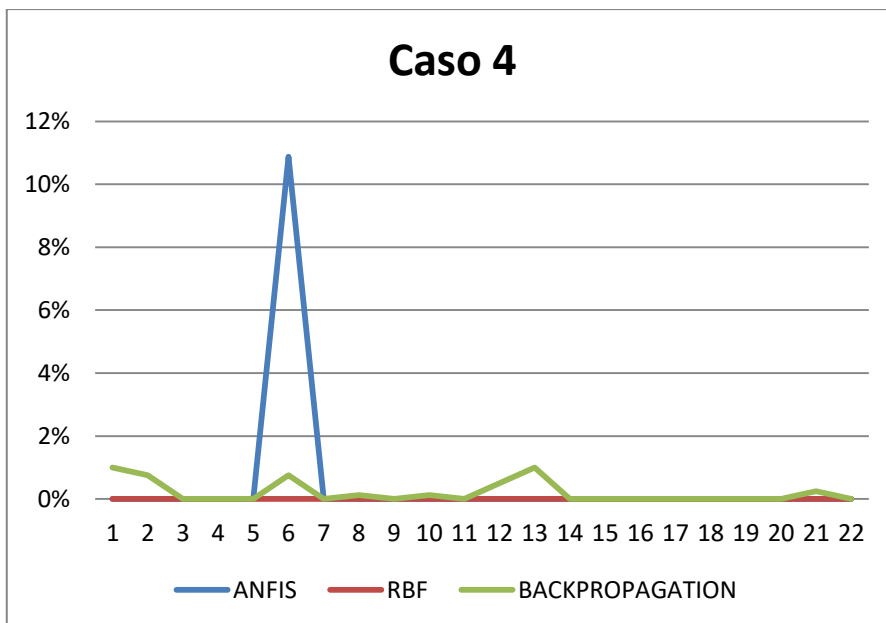


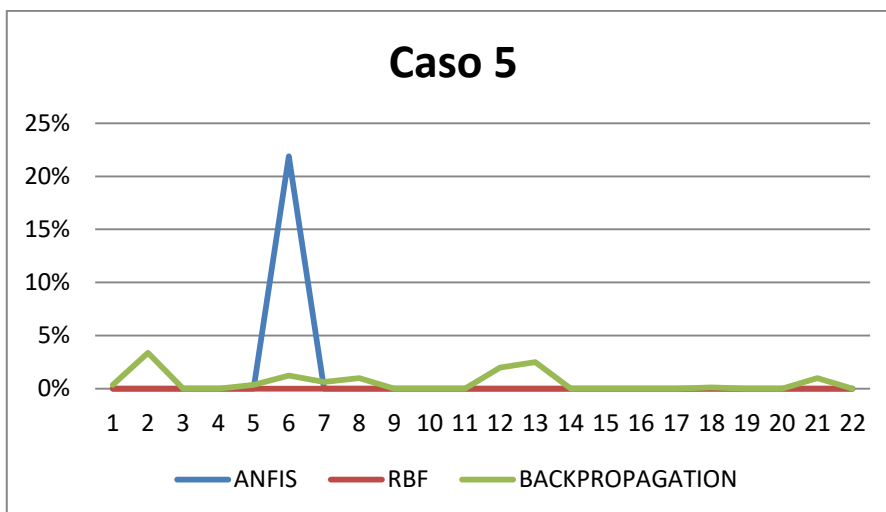
Fig 24: Porcentaje de veces que el modelo i detecta datos del caso 2



**Fig 25:** Porcentaje de veces que el modelo *i* detecta datos del caso 3



**Fig 26:** Porcentaje de veces que el modelo *i* detecta datos del caso 4



**Fig 27:** Porcentaje de veces que el modelo *i* detecta datos del caso 5

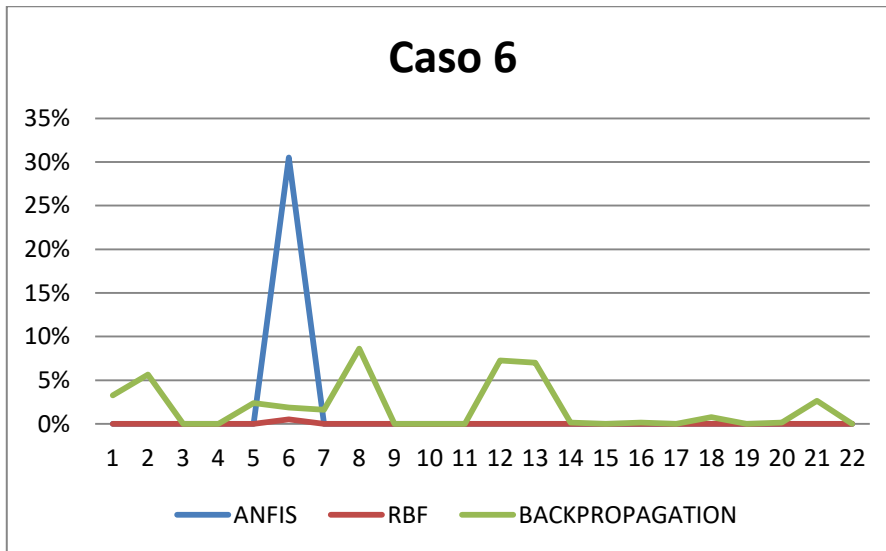


Fig 28: Porcentaje de veces que el modelo i detecta datos del caso 6

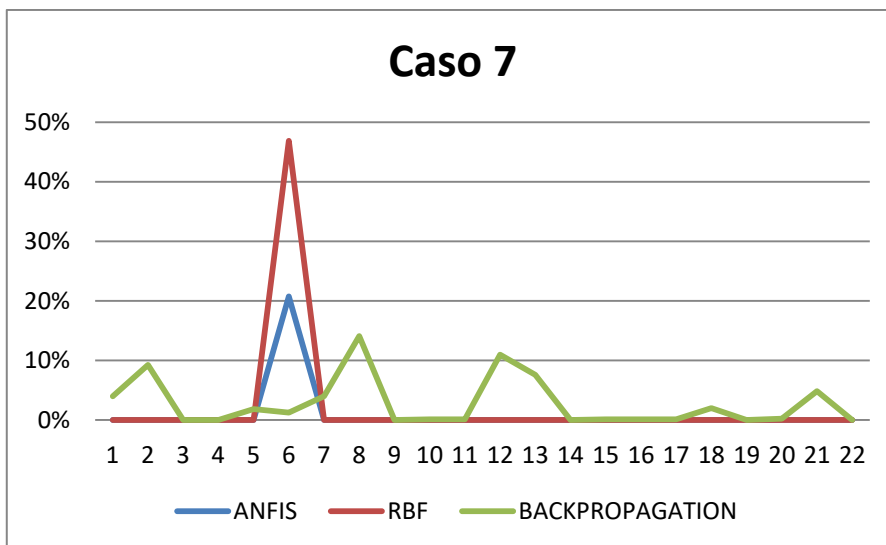


Fig 29: Porcentaje de veces que el modelo i detecta datos del caso 7

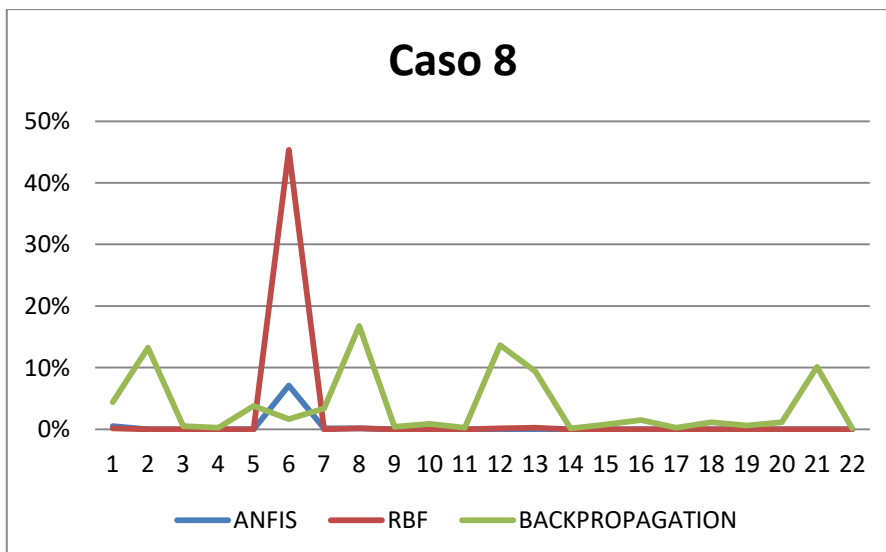
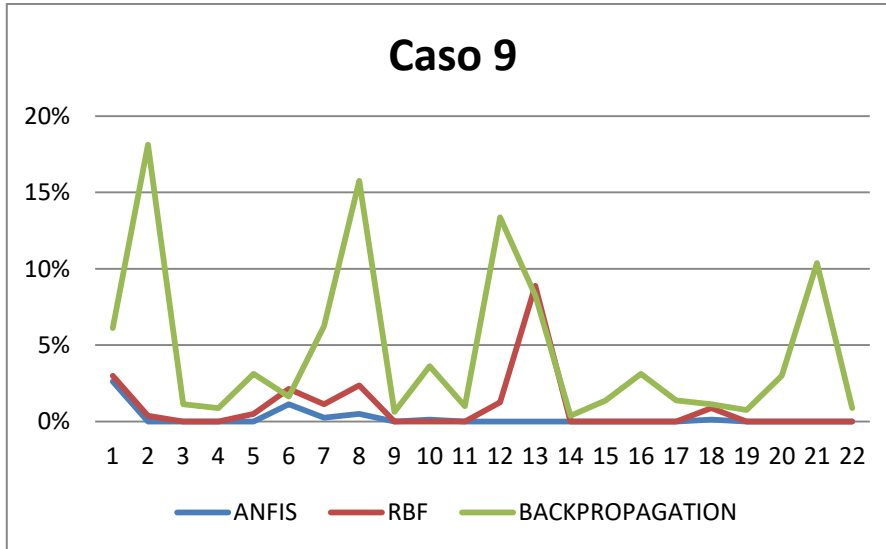
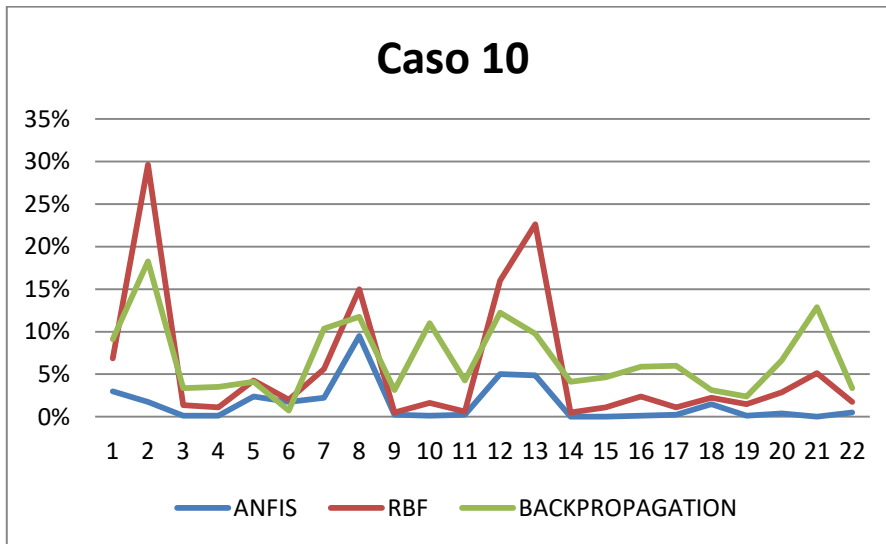


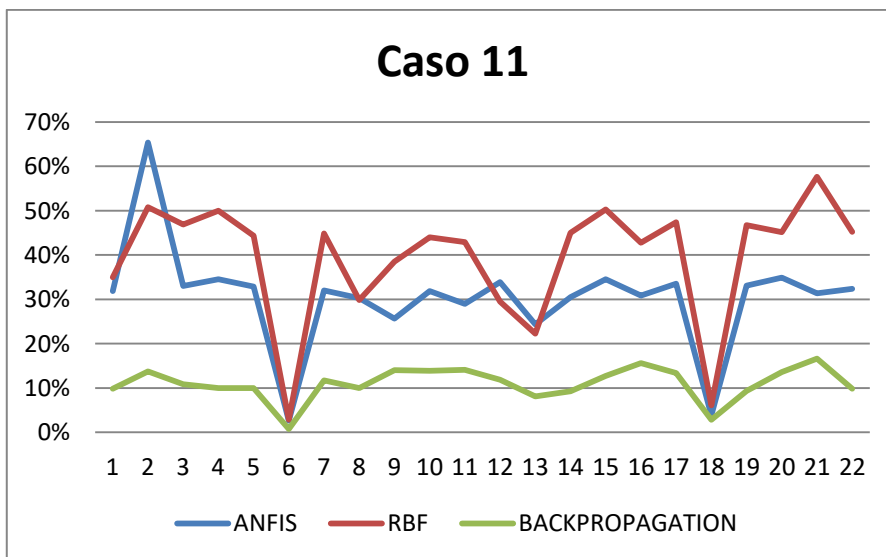
Fig 30: Porcentaje de veces que el modelo i detecta datos del caso 8



**Fig 31:** Porcentaje de veces que el modelo *i* detecta datos del caso 9



**Fig 32:** Porcentaje de veces que el modelo *i* detecta datos del caso 10



**Fig 33:** Porcentaje de veces que el modelo *i* detecta datos del caso 11

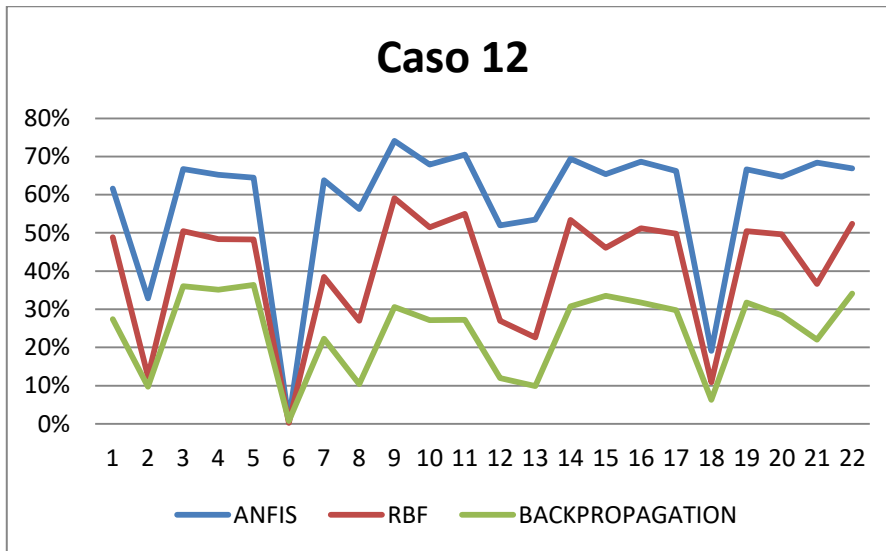


Fig 34: Porcentaje de veces que el modelo i detecta datos del caso 12

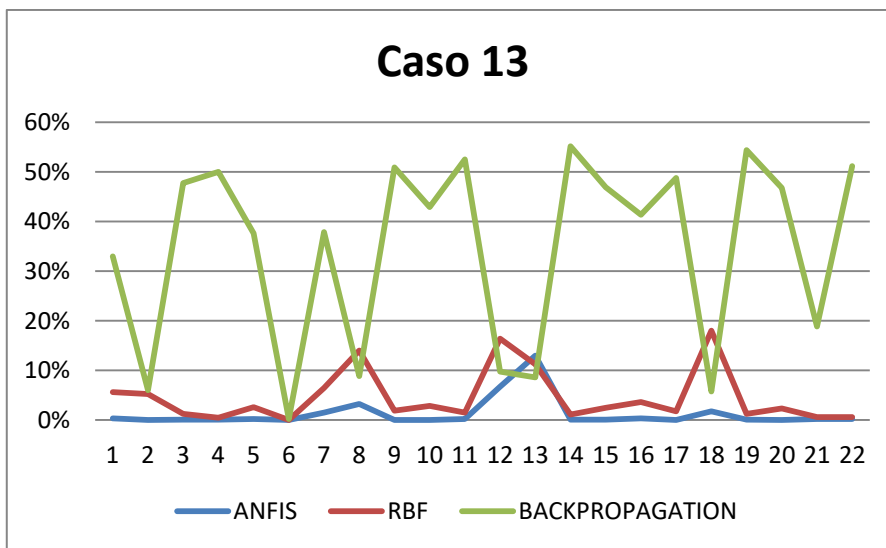


Fig 35: Porcentaje de veces que el modelo i detecta datos del caso 13

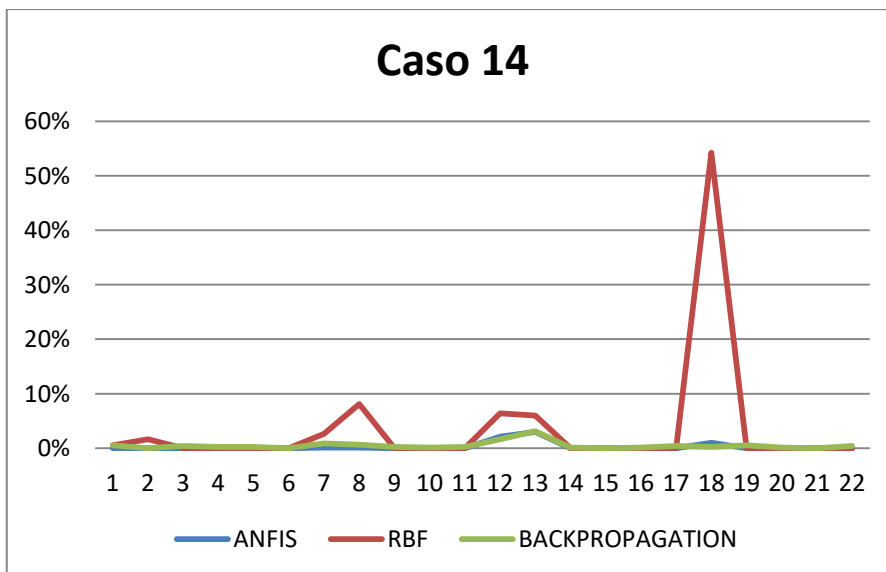
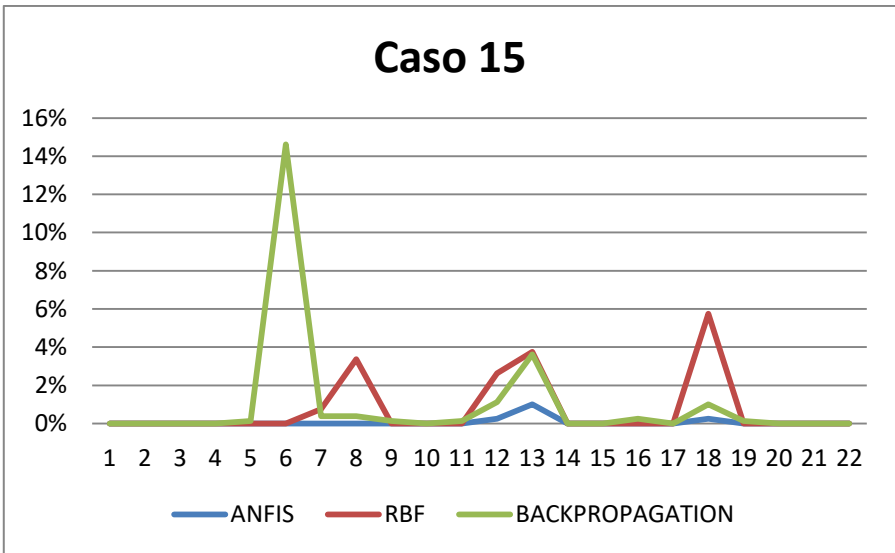
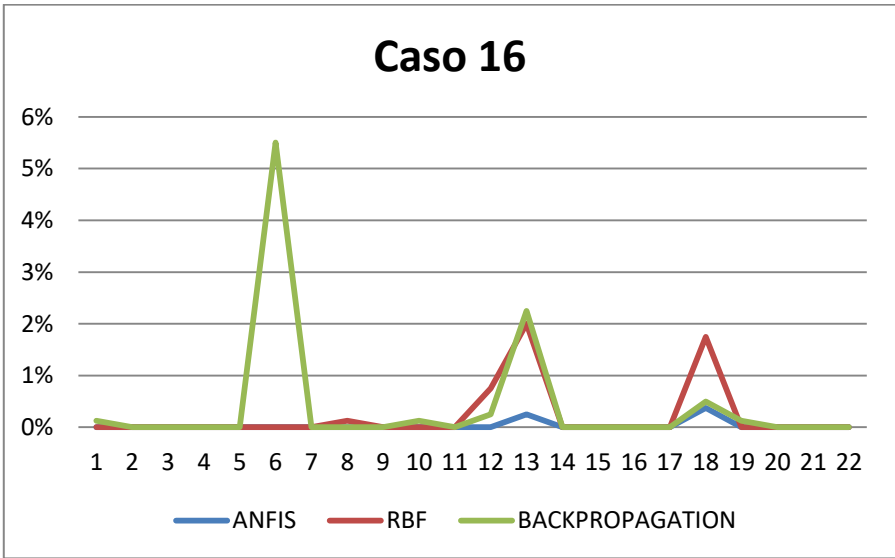


Fig 36: Porcentaje de veces que el modelo i detecta datos del caso 14

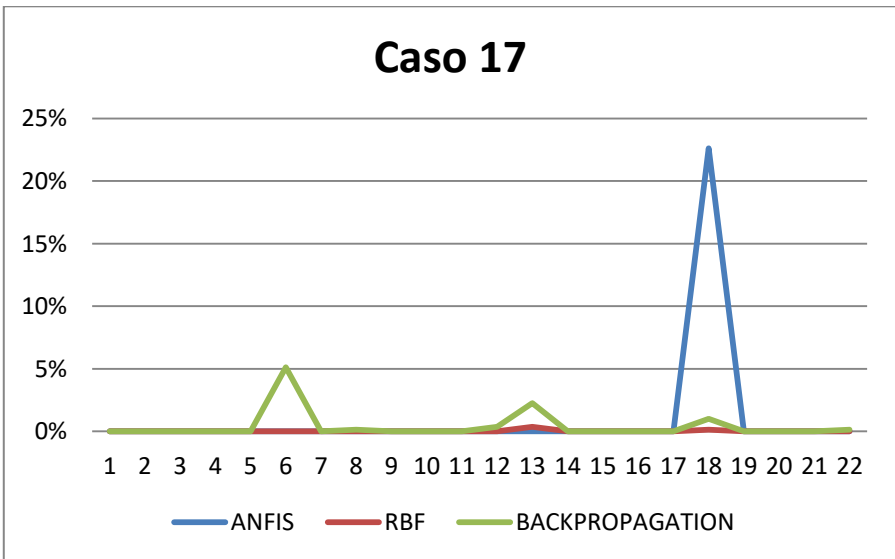




**Fig 37:** Porcentaje de veces que el modelo *i* detecta datos del caso 15



**Fig 38:** Porcentaje de veces que el modelo *i* detecta datos del caso 16



**Fig 39:** Porcentaje de veces que el modelo *i* detecta datos del caso 17

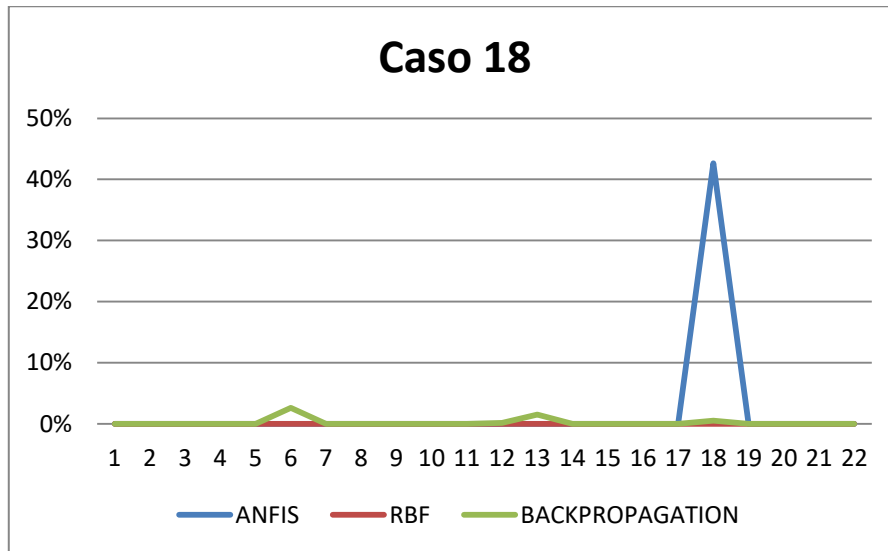


Fig 40: Porcentaje de veces que el modelo *i* detecta datos del caso 18

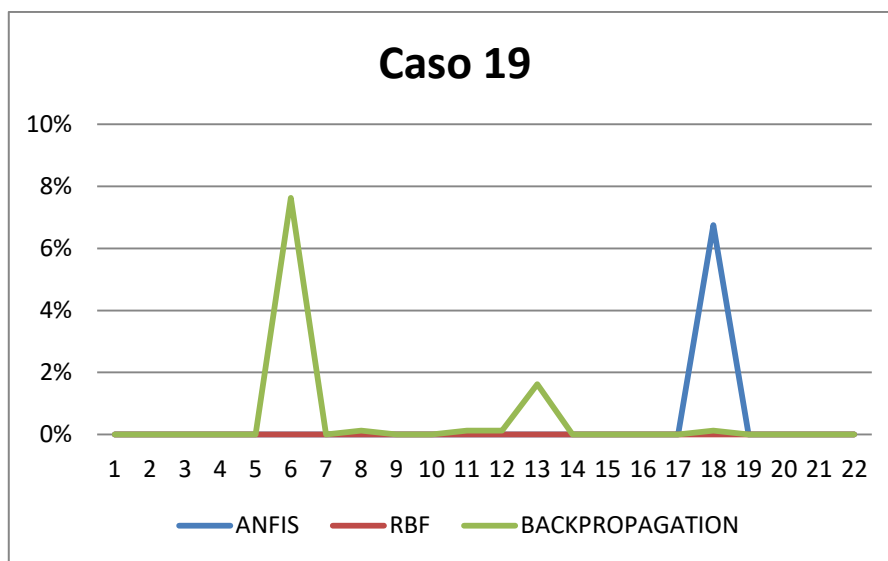


Fig 41: Porcentaje de veces que el modelo *i* detecta datos del caso 19

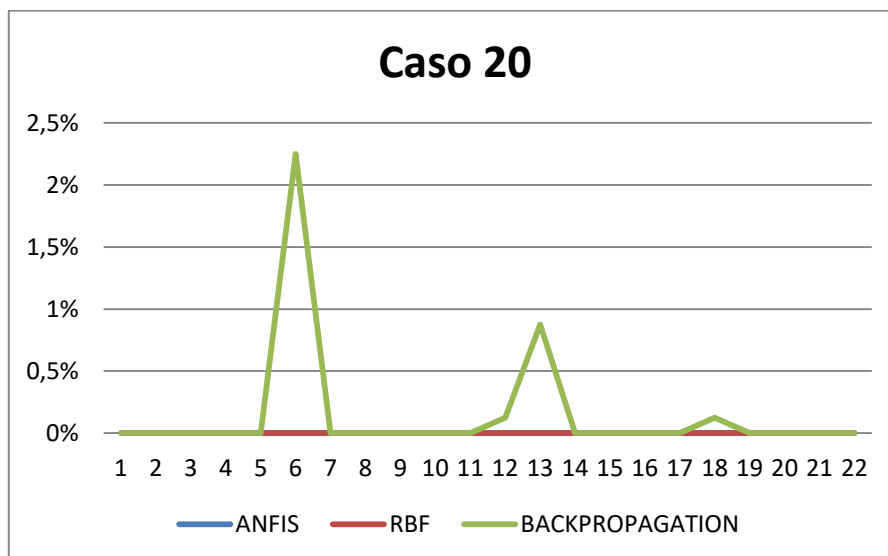


Fig 42: Porcentaje de veces que el modelo *i* detecta datos del caso 20

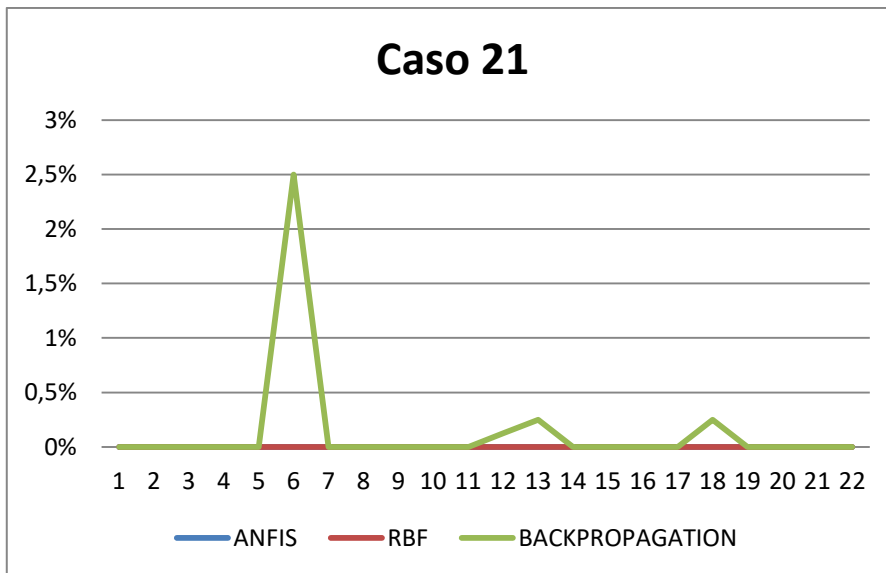


Fig 43: Porcentaje de veces que el modelo i detecta datos del caso 21

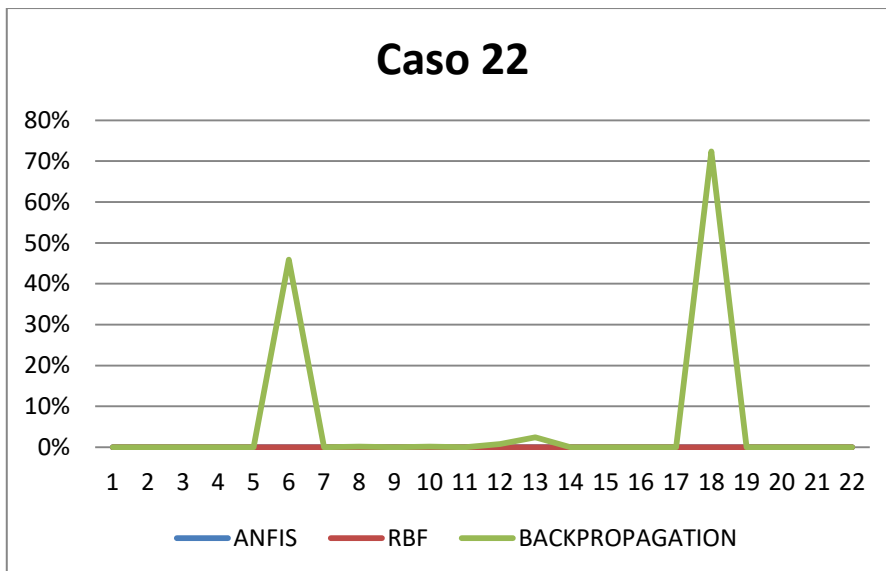


Fig 44: Porcentaje de veces que el modelo i detecta datos del caso 22

## Anexo D

# Gráficas de errores de test en modelos parciales

En las siguientes graficas reflejan el error de test obtenido al testear la red entrenada con el caso i con cada uno de los 22 fallos. Además reflejan los datos resultantes de los tres tipos de redes neuronales.

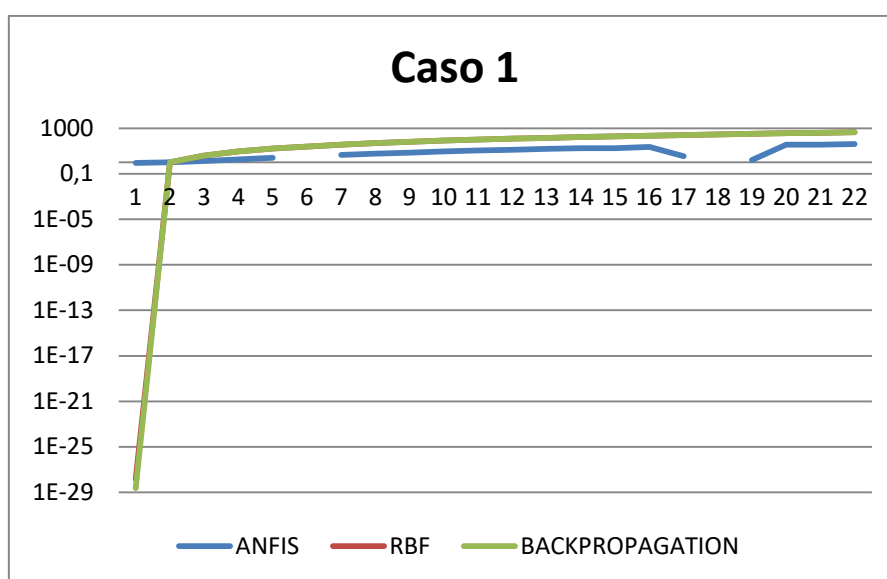


Fig 45: Error de test de modelo del caso 1 testeado con los 22 casos

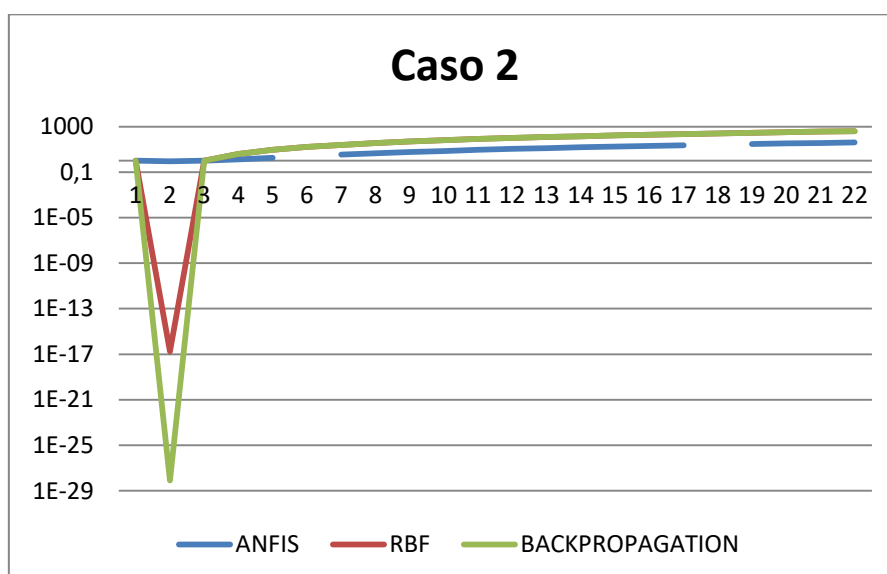


Fig 46: Error de test de modelo del caso 2 testeado con los 22 casos

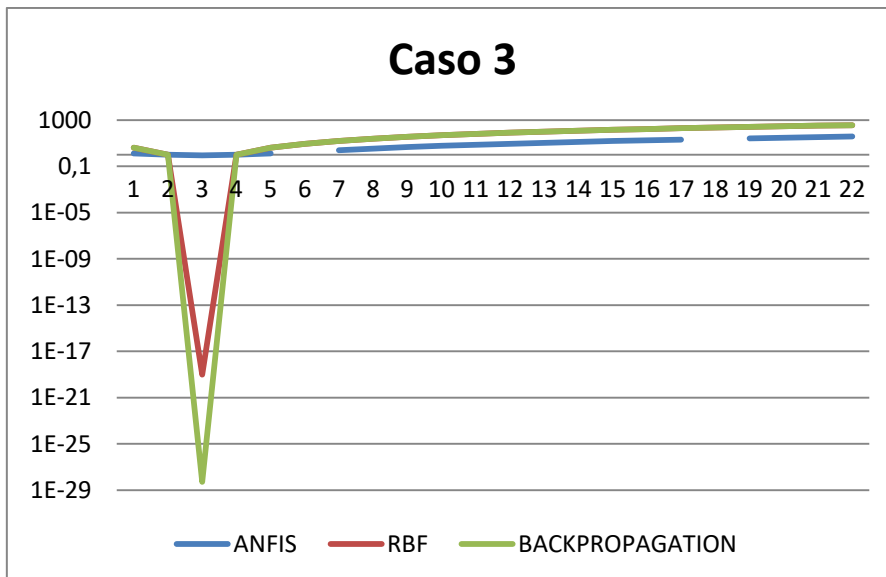


Fig 47: Error de test de modelo del caso 3 testeado con los 22 casos

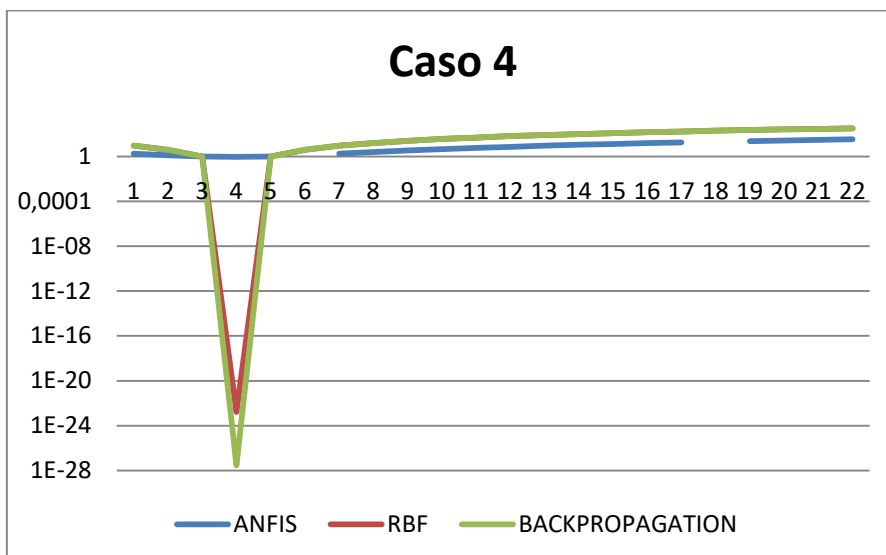


Fig 48: Error de test de modelo del caso 4 testeado con los 22 casos

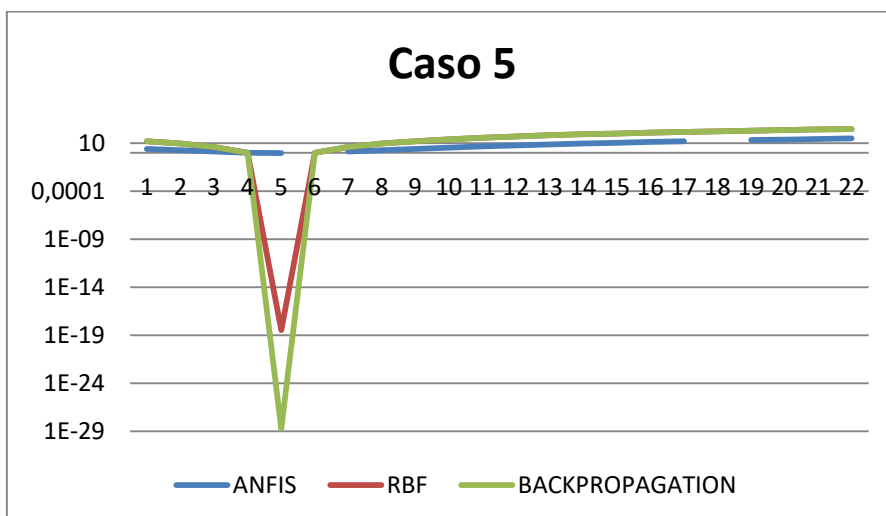


Fig 49: Error de test de modelo del caso 5 testeado con los 22 casos

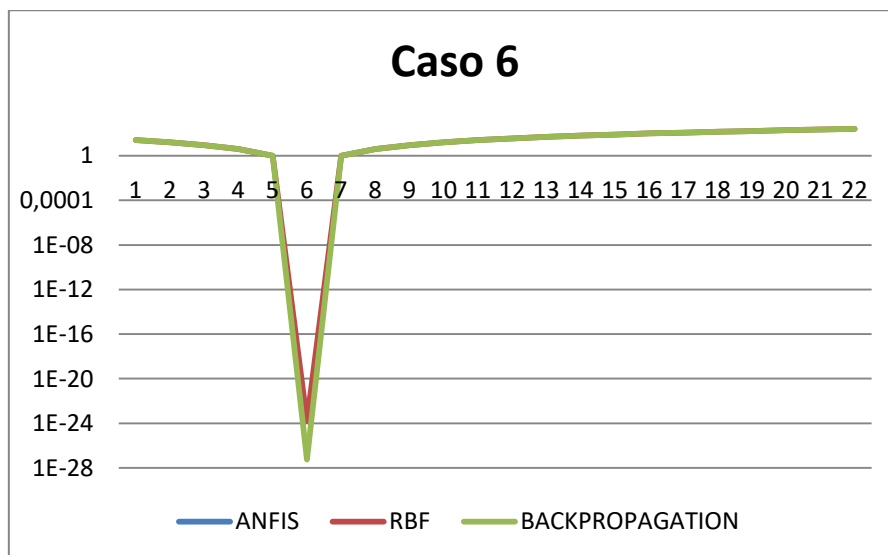


Fig 50: Error de test de modelo del caso 6 testeado con los 22 casos

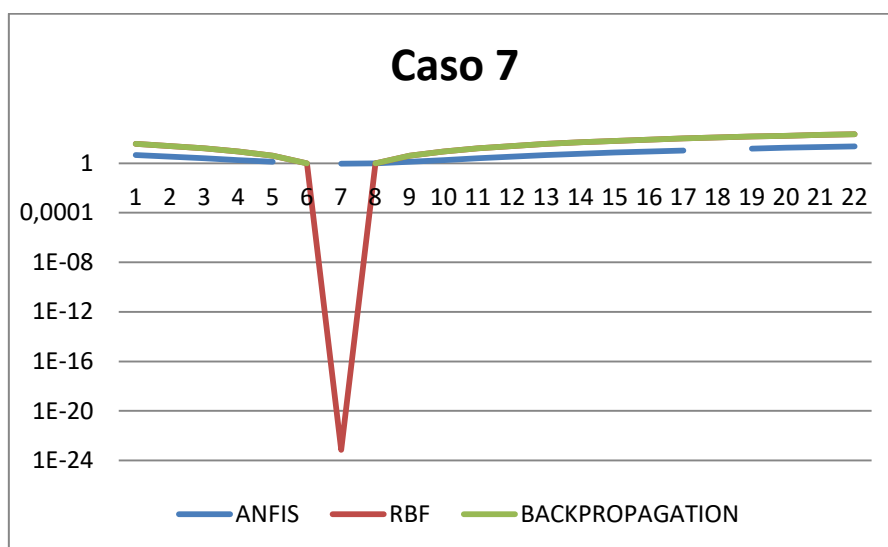


Fig 51: Error de test de modelo del caso 7 testeado con los 22 casos

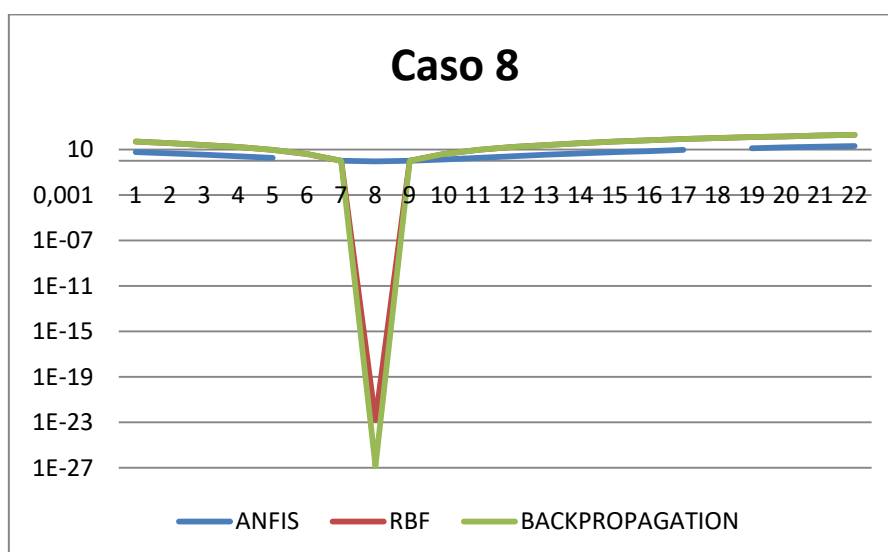
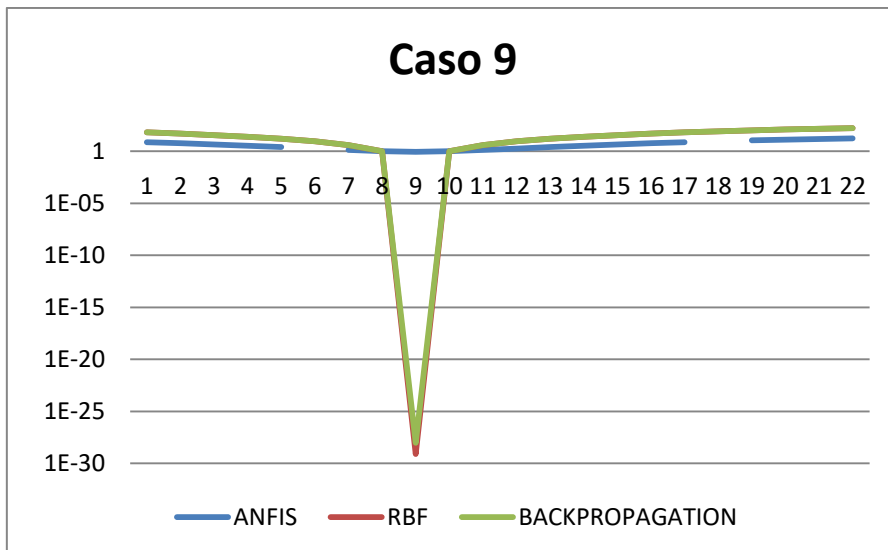
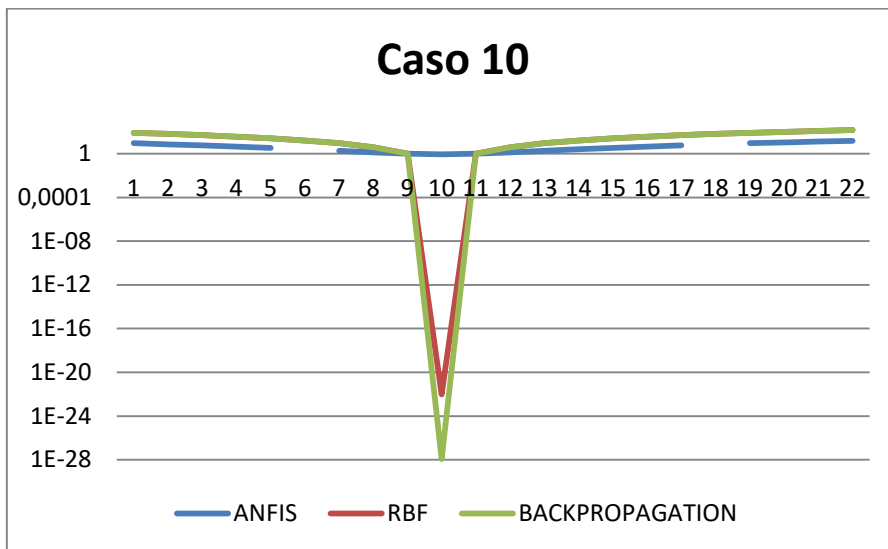


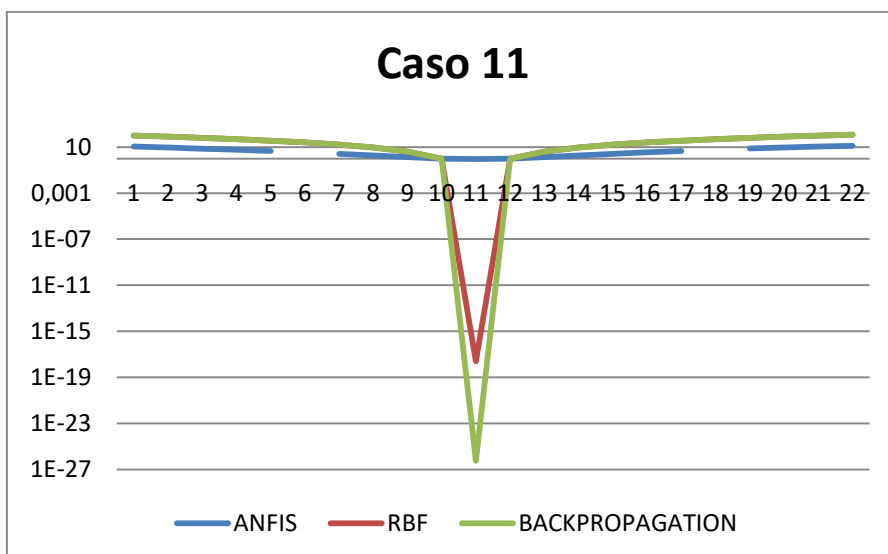
Fig 52: Error de test de modelo del caso 8 testeado con los 22 casos



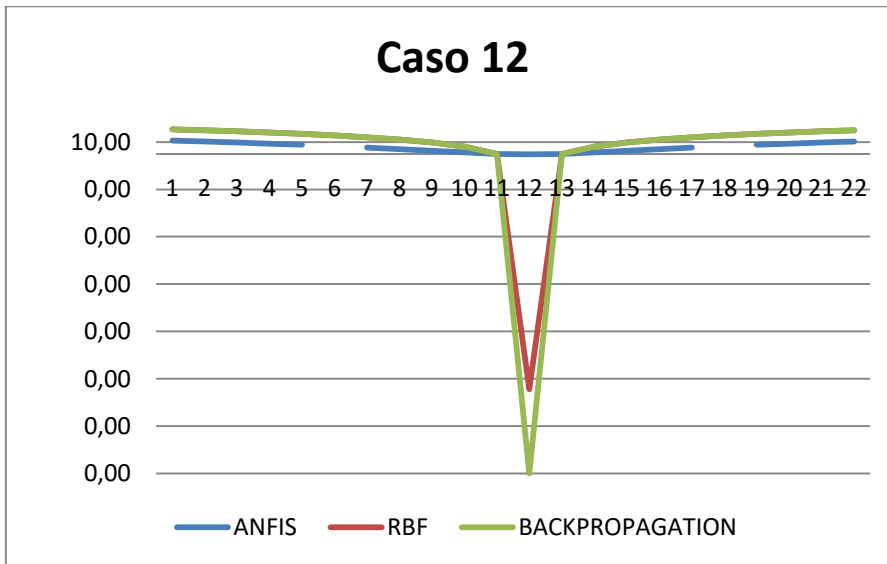
*Fig 53: Error de test de modelo del caso 9 testeado con los 22 casos*



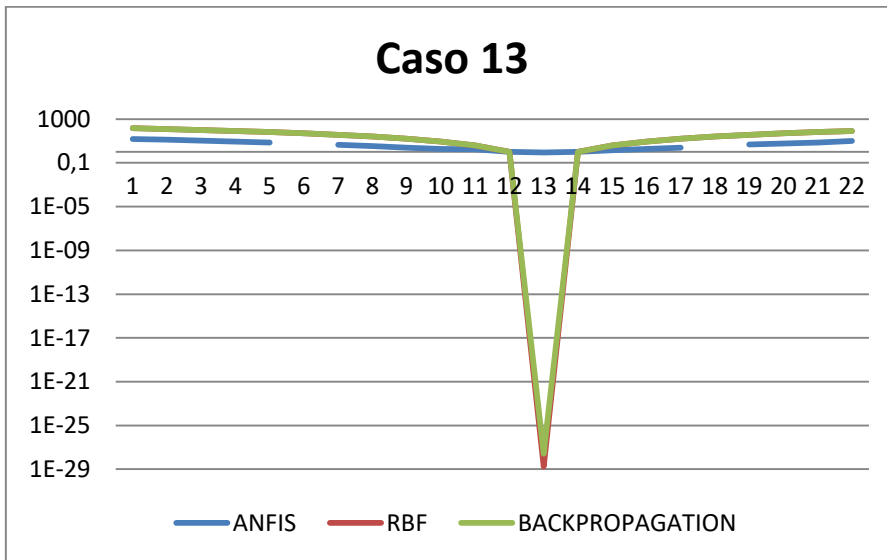
*Fig 54: Error de test de modelo del caso 10 testeado con los 22 casos*



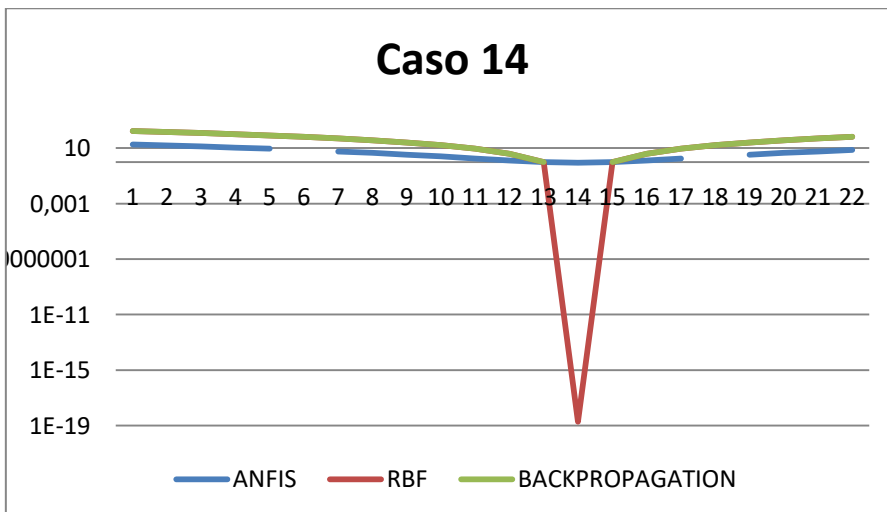
*Fig 55: Error de test de modelo del caso 11 testeado con los 22 casos*



*Fig 56: Error de test de modelo del caso 12 testeado con los 22 casos*



*Fig 57: Error de test de modelo del caso 13 testeado con los 22 casos*



*Fig 58: Error de test de modelo del caso 14 testeado con los 22 casos*



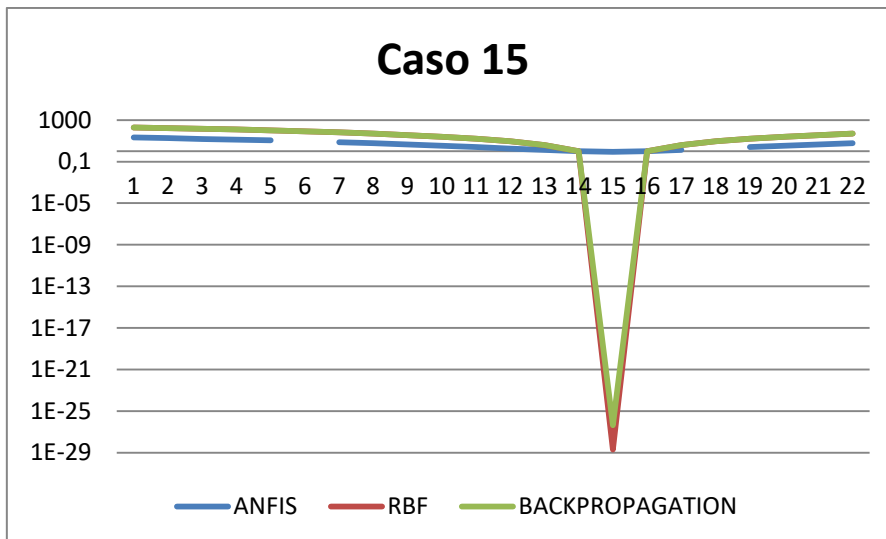


Fig 59: Error de test de modelo del caso 15 testeado con los 22 casos

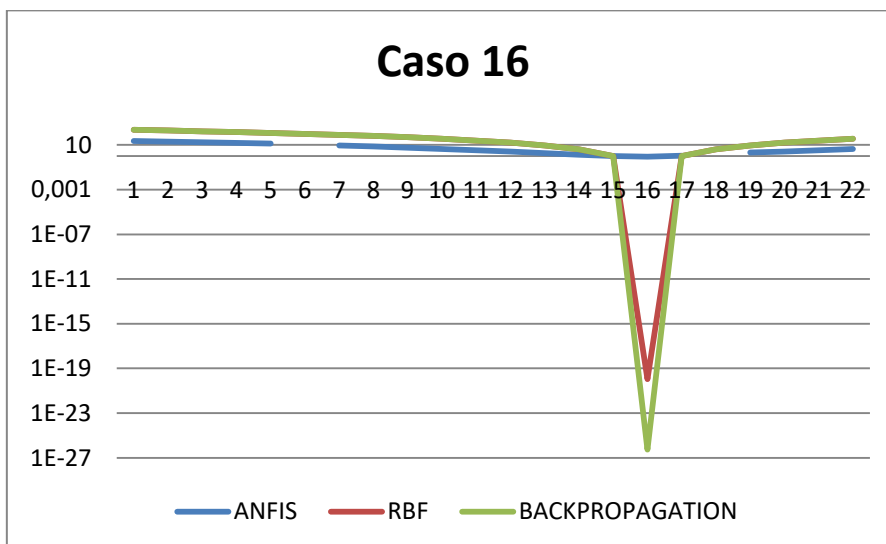


Fig 60: Error de test de modelo del caso 16 testeado con los 22 casos

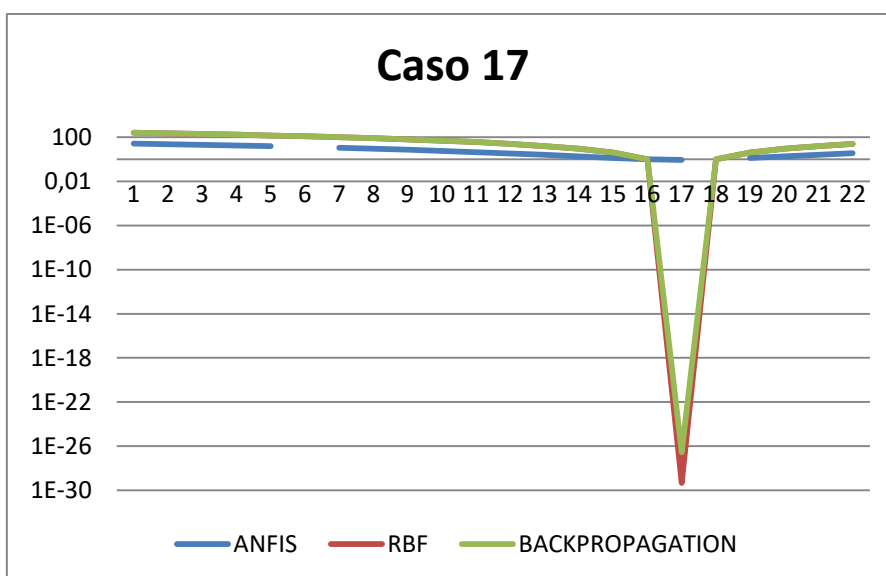


Fig 61: Error de test de modelo del caso 17 testeado con los 22 casos

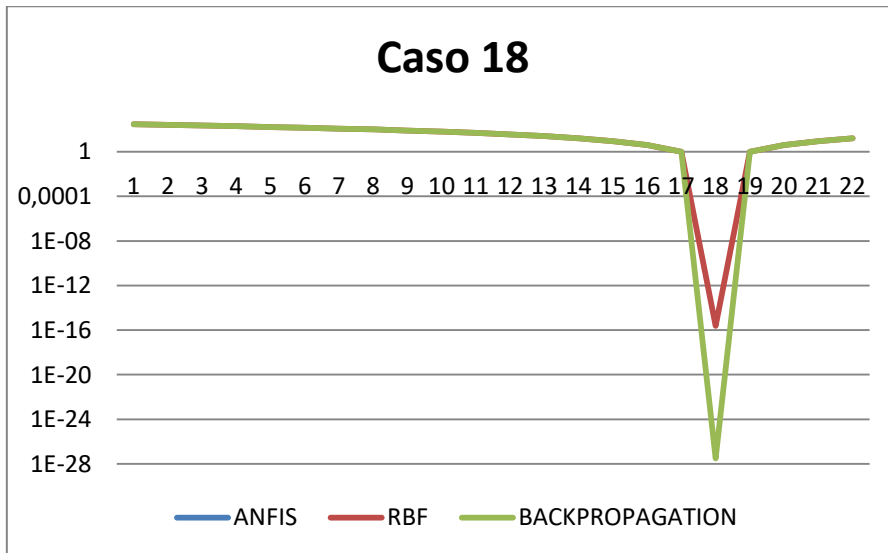


Fig 62: Error de test de modelo del caso 18 testeado con los 22 casos

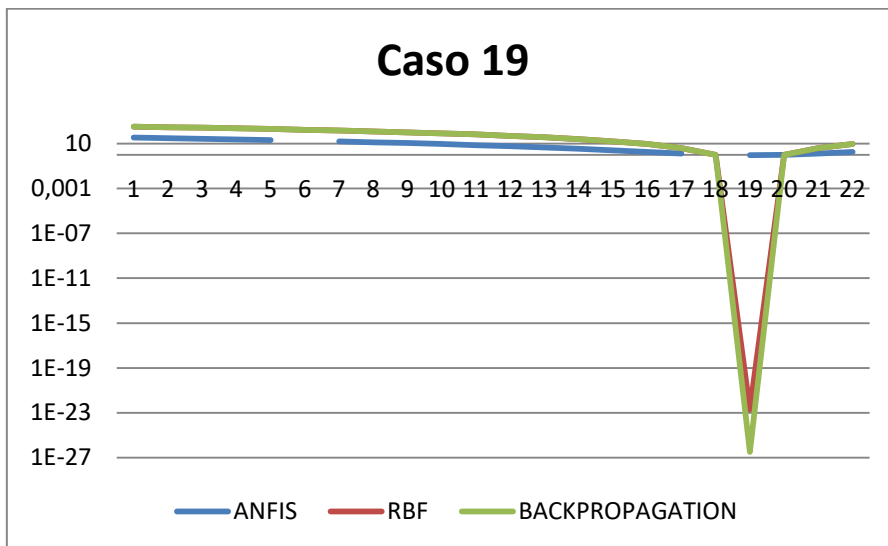


Fig 63: Error de test de modelo del caso 19 testeado con los 22 casos

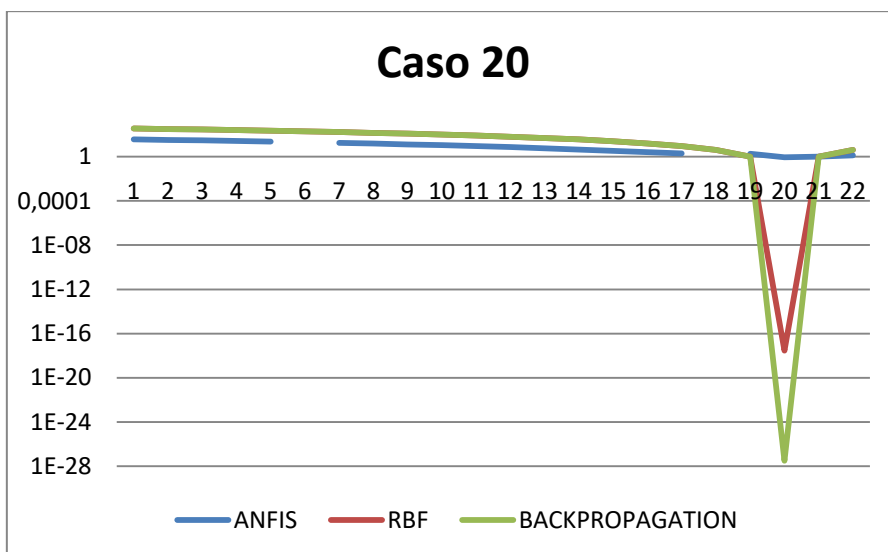
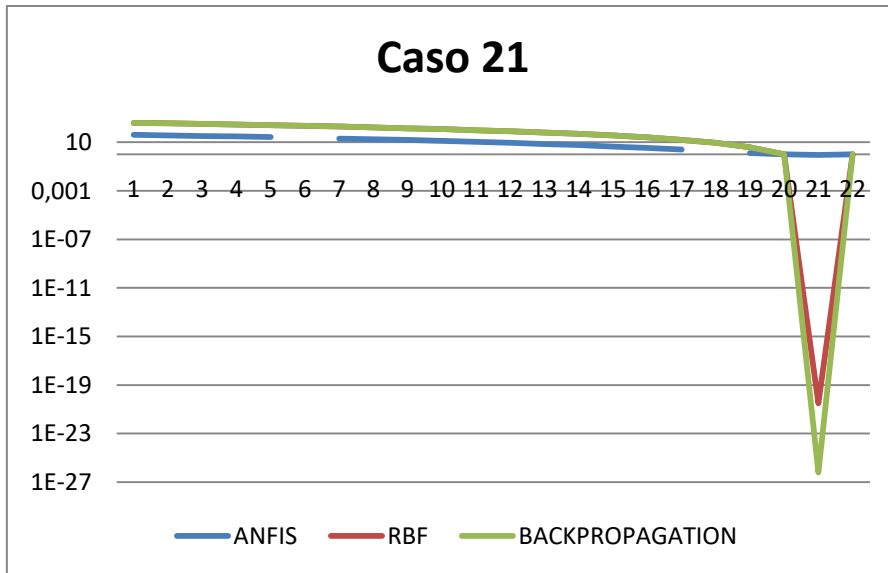
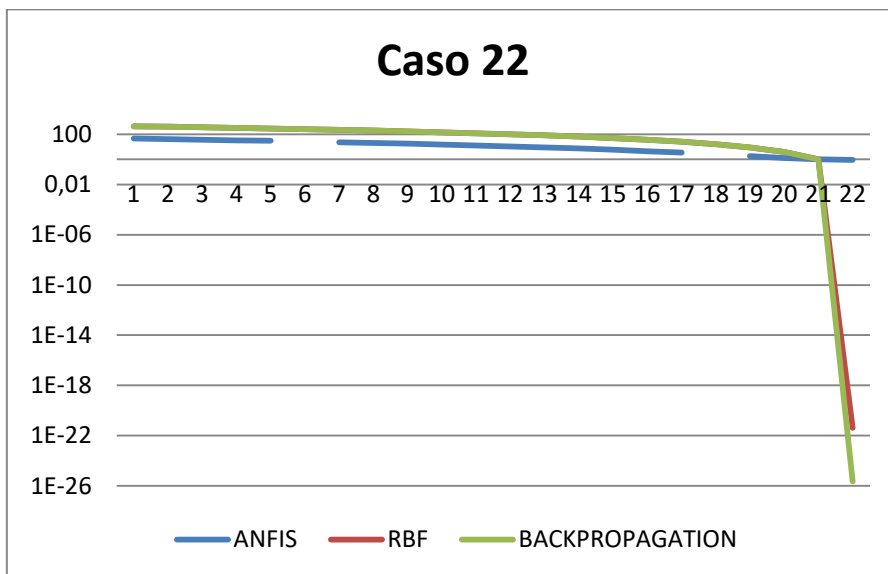


Fig 64: Error de test de modelo del caso 20 testeado con los 22 casos



**Fig 65:** Error de test de modelo del caso 21 testeado con los 22 casos



**Fig 66:** Error de test de modelo del caso 22 testeado con los 22 casos