



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería Química

**Estudio de técnicas de clasificación para
la detección y diagnóstico de fallos.**

Autor:

Medrano Sanz, Julia Isabel

Tutor:

**De la Fuente Aparicio, María Jesús
Departamento de Ingeniería de
Sistemas y Automática.**

Valladolid, julio 2019.

AGRADECIMIENTOS.

En primer lugar, quiero expresar mi más sincero agradecimiento a mi tutora María Jesús de la Fuente, por darme la oportunidad de realizar este trabajo, con el que tanto he aprendido y gracias al cual seré graduada en Ingeniería Química. Gracias por todo lo que me has enseñado y me has sabido transmitir, por estar siempre disponible y por tu paciencia.

Gracias a mi familia por haber estado siempre a mi lado durante esta etapa, sin vosotros, vuestros consejos y paciencia no habría sido posible.

Por último, gracias a todos mis amigos por haber estado ahí para celebrar todos los logros y también para escucharme cuando parecía que no había luz al final del túnel.

ÍNDICE DE CONTENIDOS

1.	RESUMEN.	7
2.	INTRODUCCIÓN Y OBJETIVOS.	11
2.1.	MOTIVACIÓN.	11
2.2.	OBJETIVOS DEL TRABAJO.	12
2.3.	ORGANIZACIÓN DE LA MEMORIA.	13
3.	ESTADO DEL ARTE.	15
3.1.	REDES NEURONALES.	16
3.1.1.	RED NEURONAL PERCEPTRÓN.	18
3.2.	REDES NEURONALES PROBABILÍSTICAS.	20
3.3.	ANÁLISIS DEL DISCRIMINANTE DE FISCHER (FDA).	22
3.4.	ÁRBOLES DE DECISIÓN.	26
3.4.1.	RANDOM FOREST	28
3.5.	MÁQUINAS DE VECTORES SOPORTE O SVM.	29
4.	PLANTA TENNESSE EASTMAN.	35
5.	SIMULACIÓN.	39
5.1.	DATOS DE LA PLANTA.	39
5.2.	REDES NEURONALES.	40
5.3.	REDES NEURONALES PROBABILÍSTICAS.	48
5.4.	DISCRIMINANTE DE FISCHER.	55
5.5.	ARBOLES DE DECISIÓN.	66
5.6.	MÁQUINAS DE VECTORES SOPORTE (SVM).	69
5.7.	COMPARACIÓN.	77
6.	CONCLUSIONES. TRABAJO FUTURO.	79
6.1.	CONCLUSIONES.	79
6.2.	TRABAJO FUTURO.	81
7.	BIBLIOGRAFÍA.	83

1. RESUMEN.

En el presente trabajo se estudian diferentes técnicas de clasificación para diagnosticar fallos o malfunciones que puede haber en diversos procesos industriales con el objetivo de tener un seguimiento de la calidad, tanto del propio proceso como de los distintos productos fabricados. Para poder llevar un estudio profundo y poder llegar a unas conclusiones solidas sobre los distintos tipos de clasificadores primero se hace un estudio teórico sobre el estado del arte de los clasificadores. En él se describe de manera teórica los clasificadores y los algoritmos de clasificación de estos. En este trabajo se van a estudiar cinco tipos de clasificadores, redes neuronales multicapa, redes neuronales probabilísticas, discriminante de Fischer, árboles de decisión y por último máquinas de vectores soporte. Posteriormente, se realiza una simulación de estos clasificadores con datos de fallo. Los datos de fallo son extraídos del conocido proceso Tennessee Eastman (TEP), se analizan únicamente cuatro tipos de fallo (fallo 1, fallo 2, fallo 4 y fallo 5). Una vez se ha realizado la simulación se comparan los resultados obtenidos con los diferentes clasificadores y se hace un breve estudio sobre el trabajo futuro que se podría llevar a cabo para en el futuro mejorar este trabajo.

Palabras clave: clasificación, diagnosis de fallos, Tennessee Eastman, redes neuronales, árboles de decisión, máquinas de vectores soporte, análisis del discriminante de Fisher.

ABSTRACT.

This work studies different techniques of classification faults or malfunctions that nowadays apply to diverse industrial processes with the aim of having a follow-up of the quality, of the own process as of the different manufactured products. In order to be able to carry out a deep study and to reach some solid conclusions about the different types of classifiers, a theoretical study is first made. This study is about the classifiers, its theoretical formalism and the classification algorithms of them. In this paper we will study five types of classifiers, multilayer neural networks, probabilistic neural networks, Fischer discriminant analysis, decision trees and finally support vector machines. Subsequently, a simulation of these classifiers is carried out with fault data. Fault data are extracted from the well-known Tennessee Eastman process (TEP), only four types of failure are analysed (failure 1, failure 2, failure 4 and failure 5). Once the simulation has been carried out, the results obtained are compared with the different classifiers and a brief study is made about the future work that could be carried out to improve this work in the future.

Keywords: classification, fault diagnosis, Tennessee Eastman. Neural networks, decision trees, SVM, FDA

2. INTRODUCCIÓN Y OBJETIVOS.

2.1. MOTIVACIÓN.

Las industrias están constantemente tras la búsqueda de herramientas, elementos, guías, métodos, estrategias y estándares dirigidos a facilitar uno de sus principales objetivos, proteger sus instalaciones, procesos, personal e inversión de otros tantos parámetros influyentes. Para lograr estos objetivos, se llevan a cabo ciertas actividades de planificación e investigación, dirigidas a encontrar cual es la combinación clave entre instrumentos, técnicas, tecnologías y métodos que permitan controlar de forma óptima sus procesos, para lograr una correcta supervisión y monitorización de sus instalaciones y actividades. ^[1]

El diseño de sistemas de detección y diagnóstico automático de fallos presenta una significativa importancia dentro de las medidas a llevar a cabo para el correcto funcionamiento de las industrias, debido a la creciente demanda internacional de sistemas de control de alta fiabilidad y seguridad, no solo para objetivos de seguridad crítica, tales como reactores nucleares, plantas químicas, etcétera, sino también para otros sistemas avanzados empleados en diversas industrias, así como en automóviles, alimentación, hospitales, y en la fabricación de diversos productos. ^[2]

Las consecuencias de fallos en los equipos e instrumentos de plantas industriales automatizadas incrementan, como mínimo, los costos de operación de estas. Si, además, se consideran aspectos más serios, el funcionamiento anormal de algún proceso puede causar accidentes catastróficos de importantes consecuencias en términos de mortalidad humana, impacto medioambiental y pérdidas económicas. Las indicaciones tempranas sobre los fallos que se puedan originar en dichas plantas ayudan a evitar averías en las mismas, paradas, así como catástrofes. Por consiguiente, la fiabilidad y seguridad de las plantas industriales debe considerarse como un problema más a tratar, por su gran importancia. ^[2]

La implementación de este tipo de sistemas plantea beneficios tanto a los fabricantes como a los consumidores finales. Los fabricantes logran conseguir la mejor calidad de sus productos, además de poder optimizar su proceso, tanto económicamente como en materia de producción. Los consumidores obtendrán dichos productos de máxima calidad pudiendo obtener precios más competitivos. ^[2] Por tanto, es necesario implementar sistemas de detección y diagnóstico de fallos capaces de detectar cualquier fallo que se presente en la industria.

La palabra fallo, representa una desviación no permitida en al menos, una propiedad característica o un parámetro del sistema, que lo aparta de la situación aceptable, habitual o estándar. Según esta definición, el significado del término fallo se puede extender al término cambio. Este cambio puede ocurrir debido a una variación en el modo de operación o debido a una mala función que puede tener lugar en los sensores, actuadores o en la propia planta. [1]

Existen diferentes puntos de vista para clasificar los fallos, en correspondencia con la forma en que el sistema se vea afectado por los mismos, dependiendo de la forma en que aparezcan, dependiendo de los elementos que fallen, etc. Los fallos se pueden clasificar como fallos de aditivos o fallos multiplicativos según la primera clasificación, en fallos abruptos, fallos incipientes o intermitentes siguiendo el segundo tipo de clasificación o fallos en los sensores, en los actuadores o en el propio proceso según el tercer tipo de clasificación. [2]

El problema de la detección y diagnóstico de fallos ha sido abordado desde distintos puntos de vista a lo largo de los años. Se pueden distinguir tres métodos: los basados en modelos o método analíticos, los métodos basados en datos y los métodos basados en conocimiento.

Los métodos de detección de fallos, en su mayoría se basan en el seguimiento de variables presentes en los procesos, con el objetivo de detectar anomalías a través de la comparación de las distintas variables o señales medidas con las variables admitidas como óptimas.

Cabe destacar la detección basada en métodos basados en datos porque es el tipo de detección y clasificación en la que se basa este trabajo. La detección basada en datos utiliza únicamente herramientas matemáticas o analíticas (procesado de señales, modelos matemáticos precisos basados en datos, etc.) para realizar sus funciones. El funcionamiento de este tipo de detectores se basa en la comparación de las salidas de los modelos basados en datos con las señales medidas de la planta.

2.2. OBJETIVOS DEL TRABAJO.

Puesto que en cualquier industria se pueden producir fallos que afectan a la fiabilidad y seguridad de la planta y a la calidad de los productos fabricados, en este trabajo planteamos técnicas de diagnosis de fallos que pretenden detectar cualquier anomalía o malfuncionamiento en la planta. Es decir, tomando datos de la planta y aplicando las técnicas desarrolladas en este trabajo decidiremos el modo de funcionamiento actual de la planta, normal o con algún tipo de fallos.

Para realizar este objetivo general, este trabajo se centra en el estudio de diferentes algoritmos de clasificación, utilizados para detectar y diagnosticar los fallos que se pueden producir en una planta. Y en concreto estudiaremos varias posibilidades unas basadas en la inteligencia computacional, como son las redes neuronales y en concreto redes neuronales perceptrón múltiple (MLP), y redes probabilísticas (PN), las máquinas de vectores soporte (SVM) y los árboles de decisión. Todos estos métodos hacen una clasificación no-lineal. Además, utilizaremos otro tipo de clasificadores basados en el control estadístico de procesos, como es el Análisis discriminante de Fisher (FDA), donde la clasificación realizada es lineal.

Aplicaremos todos estos métodos sobre un benchmark sacado de la literatura científica, una planta química no lineal, una planta de proceso Tennessee (TEP), usada para probar técnicas de control de procesos y de monitorización y detección de fallos. Después haremos una comparativa de todos los métodos de clasificación para ver cuál es el mejor, en este campo de la detección y diagnóstico de fallos.

2.3. ORGANIZACIÓN DE LA MEMORIA.

Por cuestiones de organización, el presente trabajo lo dividimos en las siguientes partes:

En el primer capítulo, se hace una introducción al tema tratado en donde explica su importancia y la motivación para la realización de este mismo. También se explican los objetivos y la organización de la memoria.

En el segundo capítulo, se desarrolla una introducción teórica sobre la detección de fallos. Se explica el fundamento teórico y el modo de implementación de todos los tipos de clasificación que se han usado durante el trabajo. En conclusión, en este capítulo se desarrolla la teoría necesaria para la detección y clasificación de fallos.

En el tercer capítulo, se hace una descripción detallada de la planta, historia, funcionamiento y equipos entre otros. También se explica porque se usan los datos generados por esta planta para la clasificación y el diagnóstico de fallos.

En el cuarto capítulo, se explicarán los tipos de datos a tratar y la parte práctica del proyecto. Constará de la parte de simulación, indicando los pasos realizados para cada tipo de clasificación. Se expondrán los resultados obtenidos según los distintos tipos de datos tratados.

En el capítulo quinto se hace una vista atrás al proyecto para analizar tanto la evolución como los resultados obtenidos. Se expondrá el trabajo futuro que se podría realizar para conseguir métodos más eficaces e innovadores.

En el sexto capítulo se encuentra la bibliografía utilizada para la realización de este trabajo.

3. ESTADO DEL ARTE.

La gran demanda por producir productos de alta calidad ha provocado que características como fiabilidad, disponibilidad, seguridad y protección medioambiental adquieran cada vez más importancia. Los controladores estándar están diseñados para mantener los procesos operando de forma satisfactoria, pero a veces aparecen causas especiales que se traducen en un incorrecto funcionamiento del proceso. Estas causas especiales son causadas por errores en el proceso o en los componentes del mismo. Para evitar esta situación o prever la posible degradación se presenta necesario dotar a los sistemas de control de un esquema de monitorización capaz de detectar fallos incipientes o el inicio de la degradación de un proceso para evitar el daño o futuras pérdidas. [3]

Cabe destacar la diferencia entre los términos fallo y avería. Fallo es un cambio inesperado del sistema que lleva una degradación del comportamiento global del mismo, pero que no tiene por qué representar un defecto de ninguno de sus componentes. En cambio, avería se refiere a un comportamiento catastrófico del sistema lo que conlleva una interrupción permanente de la capacidad para mantener una función requerida por el sistema. [3]

Un método de detección y diagnóstico de fallos (FDD) debe poseer las siguientes características, con las que se puede evaluar su rendimiento: [3]

- Tiempo bajo de detección (t_d): tiempo que tarda el esquema de monitorización en detectar el fallo desde que se produjo.
- Un alto grado de detección correctas (P_d): todos los fallos detectados ocurren.
- Un índice bajo de falsas alarmas (P_f): una falsa alarma ocurre cuando el método detecta un fallo, pero el sistema está en condiciones normales de operación.
- Buena capacidad de aislamiento: es la capacidad para distinguir entre los distintos fallos.
- Sensibilidad es una medida que determinará la magnitud del fallo necesaria para poder aislar dicho fallo bajo condiciones preestablecidas.
- Robustez: es la habilidad de aislar fallos en presencia de errores de modelado y perturbaciones desconocidas.

Existen tres métodos diferentes para transformar los datos tomados de un proceso en medidas útiles, y en caso de ser necesario, diagnosticar los fallos encontrados. Estos métodos son: métodos basados en señal, métodos analíticos y métodos basados en conocimiento. [3]

En este trabajo, casi todos los métodos que se utilizan para clasificar los fallos serán métodos basados en datos. Este tipo de clasificadores son las redes neuronales, las redes neuronales probabilísticas, los árboles de decisión y las máquinas de vector soporte. Aunque también se utiliza un tipo de método basado en señal, el discriminante de Fischer.

Una vez se ha realizado esta introducción para entender el estudio del arte del diagnóstico y clasificación de fallos se va a describir de manera breve los conceptos básicos de cada tipo de clasificador que se utilizará posteriormente en la simulación. Se verán esquemas de cada uno de ellos y la manera en la que se tienen que diseñar o que datos tienen que tratar para que clasifiquen de manera adecuada.

3.1. REDES NEURONALES.

Las redes de neuronas artificiales (RNA) son dispositivos programados de tal manera que funcionen como un proceso automático inspirado en la forma en la que trabaja el sistema nervioso de los seres vivos. Se trata de un sistema de interconexión de neuronas en una red que, operando de forma paralela, consiguen resolver problemas relacionados con el reconocimiento de formas o patrones, predicción, codificación, control y optimización, para producir un estímulo de salida. [4]

La neurona es el elemento fundamental para que el sistema nervioso entre en funcionamiento. Generalmente, una neurona consta de un cuerpo celular más o menos esférico del que salen una rama principal, el axón, y varias ramas más cortas llamadas dendritas. Una de las características de las neuronas es su capacidad para comunicarse. Las dendritas y el cuerpo celular reciben señales de entrada; el cuerpo celular combina e integra y emite señales de salida. El axón transmite dichas señales a los terminales axónicos, que distribuyen información a un nuevo conjunto de neuronas.

Para construir una similitud directa entre la actividad sináptica y la analogía con las redes neuronales artificiales, se pueden considerar que las señales que llegan a la sinapsis son las entradas a la neurona. Estas son ponderadas (atenuadas o simplificadas) a través de un parámetro denominado peso asociado a la sinapsis correspondiente. Estas señales de entrada pueden excitar a la neurona (sinapsis con peso positivo) o inhibirla (peso negativo). El efecto es la suma de las entradas ponderadas, siendo esta señal del tipo todo/nada, es decir, la neurona se activa o no se activa. [4]

Una neurona artificial es un procesador elemental en el que se procesa una o varias entradas, donde puede verse aumentada su intensidad debido al parámetro característico de conexión denominado peso. Las conexiones

determinan la influencia entre neuronas. Finalmente, se produce una respuesta o salida única.

Los componentes principales de una neurona artificial son las entradas, los pesos, las salidas y las funciones de activación. En la figura 1 se encuentra un esquema de una neurona artificial.

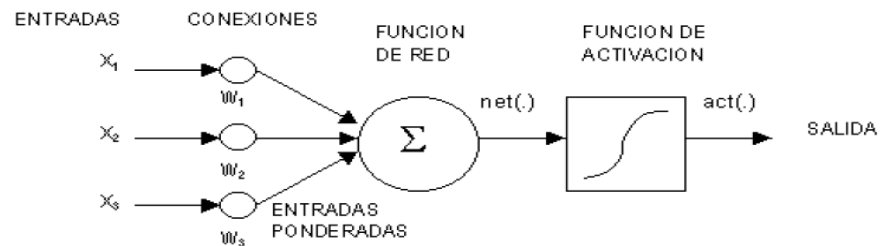


Figura 1. Esquema de una neurona artificial.

Las entradas son la información que reciben del exterior. Cada neurona suele recibir varias entradas. Los pesos se pueden asemejar a la importancia que tienen las distintas entradas que llegan a la neurona. Los pesos pueden adaptarse dentro de la red y constituyen la medida de la fuerza de una conexión de entrada. Las salidas vienen definidas por los valores que dan las distintas funciones de activación. Las funciones de activación transforman la entrada neta que la neurona presenta al valor de salida. La función escalón o la Gaussiana son ejemplos de funciones de activación. [4]

La combinación de dos o más neuronas forma una capa, y una o más capas forman una red. Generalmente, los elementos de una misma capa tienen la misma función de activación. Cada capa puede recibir información de una o más capas, y después de procesarla puede enviar su salida a una o más capas.

El caso más general a tener en cuenta para analizar una capa de neuronas es considerar que las neuronas de la capa poseen múltiples entradas, su esquema sería similar al de la figura 1. Pero cuando la red neuronal está formada por una o más capas de neuronas, su esquema se puede ver en la figura 2. En esta figura se ven las distintas capas que componen la red neuronal multicapa. La primera capa o capa de entrada, recibe la información de fuentes externas a la red. La capa oculta, no tiene contacto con el exterior y es donde se encuentran las distintas neuronas que forman la capa. La capa de salida es la encargada de transferir la información al exterior.

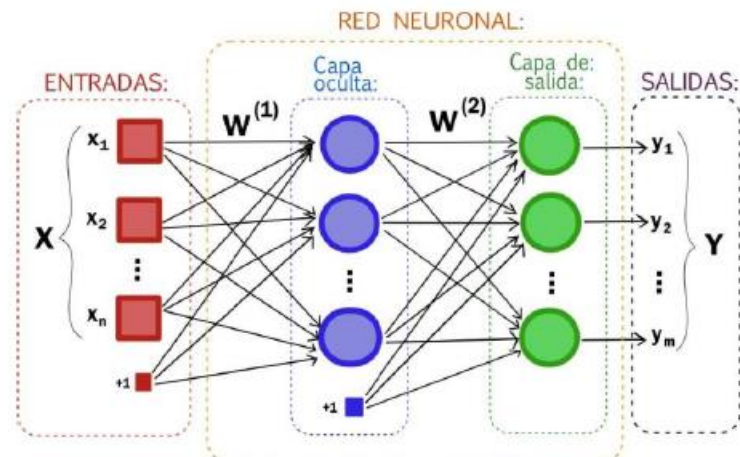


Figura 2. Esquema red neuronal multicapa. (Fuente: ceres.urg.es)

Las redes neuronales pueden actuar de dos maneras, si las entradas y señales de las capas se propagan siempre hacia adelante, o si las entradas y salidas de las capas tienen retroalimentación. La primera forma de actuación de las redes se conoce como red estática o red feedforward, mientras que la segunda se conoce como red dinámica o red recurrente. En la simulación se usarán redes neuronales multicapa que clasifiquen de manera estática. Tienen un número determinado de neuronas que se encuentran en la capa oculta. [4]

Para poder utilizar las redes neuronales como clasificadores es necesario llevar a cabo un aprendizaje previo de las mismas. El aprendizaje se define como el mecanismo que hace que los pesos tomen los valores necesarios para desempeñar la tarea deseada. Los mecanismos de aprendizaje y determinación de pesos constituyen uno de los puntos críticos en la investigación con redes neuronales. Hay varios tipos de aprendizaje como puede ser el aprendizaje supervisado, el aprendizaje no supervisado o autoorganizado, el aprendizaje híbrido y el aprendizaje reforzado.

3.1.1. RED NEURONAL PERCEPTRÓN.

Fue el primer modelo de red neuronal artificial, desarrollado en 1958 por Rosenblatt. Despertó un gran interés en los años 60 debido a su capacidad para aprender a reconocer patrones sencillos. Está formado por varias neuronas para recibir las entradas a la red, y una neurona de salida que es capaz de decidir cuándo una entrada a la red pertenece a una de las dos clases que es capaz de reconocer. La neurona de salida del Perceptrón realiza la suma ponderada de las entradas, resta el umbral y pasa el resultado a una función de transferencia de tipo escalón. La regla de decisión es responder +1 si el patrón presentado pertenece a la clase A, o -1 si el patrón pertenece a la clase

B. Al constar solamente de una capa de entrada y otra de salida con una única neurona, tiene una capacidad de representación bastante limitada. Solo es capaz de discriminar patrones muy sencillos y linealmente separables en el plano. El aprendizaje de este tipo de estructuras es de tipo supervisado, es necesario que sus resultados sean evaluados. [4]

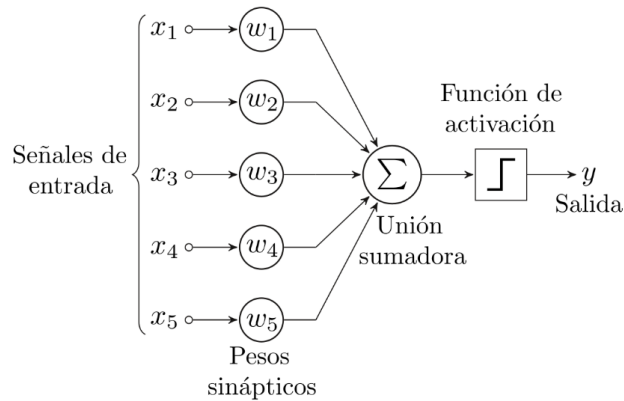


Figura 3. Esquema Perceptrón (Fuente: Wikipedia.com)

Como consecuencia de las limitaciones de la arquitectura del perceptrón simple, surge el perceptrón multicapa. En 1969, Minsky y Papert (Minsky and Papert, 1969) mostraron que la combinación de varios perceptrones simples podía resultar una solución adecuada para tratar ciertos problemas no lineales. Sin embargo, los autores no presentaron una solución al problema de cómo adaptar los pesos de la capa de entrada a la capa oculta, pues la regla de aprendizaje del Perceptrón simple no puede aplicarse en este escenario. [4]

A pesar de ello, la idea de combinar varios perceptrones sirvió de base para estudios posteriores, que dieron lugar al desarrollo de un método de retropropagación de errores medidos en la salida de la red hacia neuronas ocultas. Se ha demostrado que el Perceptrón multicapa es un aproximador universal, en el sentido de que cualquier función continua sobre un compacto de \mathbb{R}^n puede aproximarse con un Perceptrón multicapa, con al menos una capa oculta de neuronas. Un esquema general de este tipo de red es el mostrado en la figura 2, donde además la capa oculta siempre tendrá una función de activación no lineal y diferenciable, como la tangente hiperbólica.

La habilidad del Perceptrón multicapa para aprender a partir de un conjunto de ejemplos, aproximar relaciones no lineales, filtrar ruido en los datos, etc. hace que sea un modelo adecuado para abordar problemas reales. En la actualidad, es una de las arquitecturas más utilizadas en la resolución de problemas. [4]

3.2. REDES NEURONALES PROBABILÍSTICAS.

Las redes neuronales probabilísticas constituyen, junto con las redes neuronales perceptrón multicapa, una de las estructuras de aprendizaje más extendidas en la actualidad. Estas redes se enmarcan dentro de los denominados sistemas expertos y su uso se ha extendido fundamentalmente a problemas de clasificación. Las redes probabilísticas se enmarcan en los tipos de sistemas expertos que se enfrentan a problemas de tipo probabilístico pues se enfrentan a problemas con elementos de incertidumbre relevantes. [5]

Una red neuronal probabilística (PNN) es una red neuronal unidireccional, la cual se deriva de la red bayesiana y un algoritmo estadístico llamado "Kernel Fisher discriminant análisis" que fue introducido por D.F. Specht en los principios de 1990s. Pertenecen a la familia de redes neuronales con función en base radial. A diferencia de las redes neuronales definidas anteriormente, en este tipo de redes no hay que realizar el ajuste de pesos y bias. Los patrones de salida son determinados mediante la comparación y el cálculo de distancias. [5]

La regla de clasificación óptima bayesiana puede ser definida de la siguiente manera, dada una colección de muestras aleatorias de n poblaciones, la probabilidad a priori de que la muestra y_i pertenezca a la población k , es denotada como h_k . El costo asociado con una clasificación errónea de que una muestra pertenezca a la población k es denotado por l_k . La probabilidad condicional de que una muestra específica pertenezca a la población k , está dada por la función de densidad de probabilidad $f_k(y)$. Por tanto, una muestra y_i es clasificada dentro de la población k si cumple la condición establecida en la Ecuación 1, en el caso de dos poblaciones. [6]

$$h_k l_k f_k(y_i) > h_l l_l f_l(y_i) \quad (1)$$

En un PNN, las operaciones están organizadas en una red unidireccional multicapa con cuatro capas: Capa de entrada, capa oculta, capa de patrón o capa de suma y capa de salida, como se muestra en la figura 4.

La PNN es a menudo utilizada en problemas de clasificación. Cuando una entrada está presente, la primera capa calcula la distancia del vector de entradas a los vectores de entrada del entrenamiento. Esto produce un vector donde sus elementos indican cuán cercanos está la entrada de la entrada de entrenamiento. La segunda capa suma la contribución para cada clase de entradas y da su salida como un vector de probabilidades. Finalmente, una función de transferencia competitiva en la salida de la segunda capa elige el máximo de estas probabilidades, y produce un 1 (identificación positiva) para esta clase y un 0 (identificación negativa) para las clases no identificadas.

En la capa de entrada, cada neurona representa un predictor de variables. En variables categóricas, se utilizan $N-1$ neuronas cuando hay un número N de categorías. Estandariza el rango de los valores restando la mediana y dividiendo por el rango inter-cuartiles. Entonces las neuronas de la capa de entrada alimentan los valores a cada una de las neuronas en la capa oculta. [6]

La capa patrón contiene una neurona para cada posible patrón (o caso) en el conjunto de datos de entrenamiento. Almacena los valores del predictor de variables para cada patrón o caso junto con el valor objetivo. Una neurona de la capa oculta calcula la distancia Euclidiana del caso de prueba al centro de la neurona y entonces aplica una función de transferencia de tipo base radial (es decir, una función de transferencia gaussiana).

Para las redes probabilísticas hay una neurona en la capa patrón para cada categoría de la variable objetivo. La verdadera categoría objetivo de cada caso de prueba está almacenada con cada neurona oculta; el valor del peso que sale de una neurona de la capa oculta está alimentado sólo con la neurona patrón que corresponde a la categoría de la neurona oculta. Las neuronas patrón añaden los valores para la clase que representan, y así es como queda explicada la capa de suma. [7]

La capa de salida compara el peso de votos para cada categoría objetivo-acumulada en la capa patrón y utiliza el voto más grande para pronosticar la categoría objetivo.

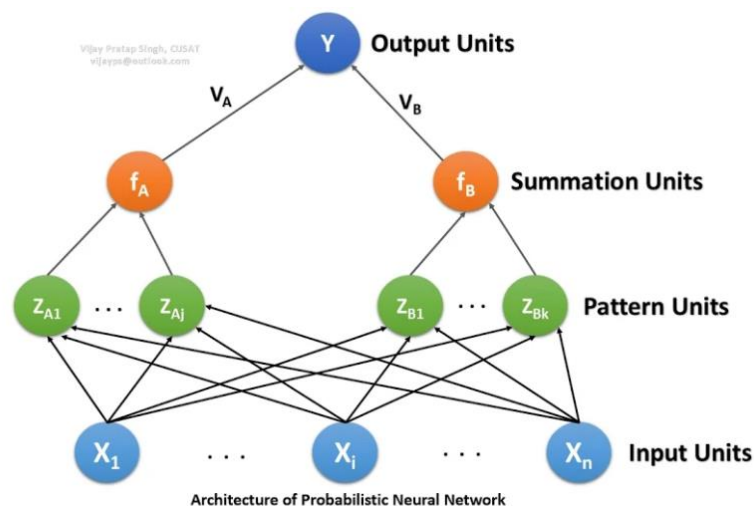


Figura 4. Esquema Red Probabilística. (Fuente: Wikipedia.com).

Este tipo de clasificador necesita aprendizaje para poder llevar a cabo después de este, su función de clasificador. Para este tipo de redes se dan dos modos de aprendizaje, el aprendizaje paramétrico y el aprendizaje estructural. Los primeros se centran en la búsqueda e identificación de la estructura de las relaciones de dependencia y correlación a través de la estructura grafo que define la red. Por su parte, los de tipo paramétrico están ligados a la estimación de valores de los parámetros asociados a los nodos y aristas del grafo que representa la red. Cabe señalar que varias estructuras de grafos pueden representar las mismas estructuras de independencia y/o las mismas distribuciones conjuntas para el conjunto de variables. [7]

3.3. ANÁLISIS DEL DISCRIMINANTE DE FISCHER (FDA).

El análisis del discriminante de Fischer o FDA, puede utilizarse para detectar fallos definiendo una clase adicional que es la de condiciones normales de operación, y utilizar una función discriminante que nos diga si los datos actuales son más similares a los datos de operación normal o a los datos de alguna las otras clases (posibles fallos). Esto se podrá utilizar siempre y cuando los datos de entrenamiento usados para calcular los FDA de la operación normal sean muy distintos a los datos recogidos de la planta cuando hay fallos.

Los datos tomados de la planta en diferentes situaciones (operación normal y/o operación con distintos tipos de fallos) se clasifican en clases, donde cada clase representa un determinado tipo de fallo. La técnica FDA es una técnica de reducción de la dimensionalidad que es óptima en términos de maximizar la separación entre clases. Es decir, determina un conjunto de vectores de transformación lineales, ordenados en términos de maximizar la distancia entre clases y minimizar la distancia dentro de la propia clase. [8]

Se definen n como el número de observaciones (datos) y m como el número de variables medibles, p es el número de clases, y n_j es el número de datos de la clase j . Si todos los datos tomados de la planta se representan en la matriz X , se obtendría la matriz representada en (2):

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{pmatrix} \quad (2)$$

Primero se definirán algunos conceptos, como la matriz de dispersión total:

$$S_t = \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})^T \quad (3)$$

donde \bar{x} es el vector media total, es decir, $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$.

X_j es el conjunto de vectores x_i que pertenecen a la clase j . Se define la matriz de dispersión de la clase j como:

$$S_j = \sum_{x_i \in X_j} (x_i - \bar{x}_j)(x_i - \bar{x}_j)^T \quad (4)$$

donde \bar{x}_j es el valor medio de la clase j , es decir: $\bar{x}_j = \frac{1}{n_j} \sum_{x_i \in X_j} x_i$

Se define también la matriz de dispersión dentro de la clase como:

$$S_w = \sum_{j=1}^p S_j \quad (5)$$

y la matriz de dispersión entre clases como:

$$S_b = \sum_{j=1}^p n_j (\bar{x}_j - \bar{x})(\bar{x}_j - \bar{x})^T \quad (6)$$

La matriz de dispersión total será la suma de las dos anteriores, es decir, $S_t = S_b + S_w$.

El objetivo del primer vector del análisis discriminante de Fisher es maximizar la dispersión entre clases y minimizar la dispersión dentro de la misma clase, es decir:

$$\max_{v \neq 0} \frac{v^T S_b v}{v^T S_w v} \quad (7)$$

Suponiendo S_w sea invertible y $v \in \mathbb{R}^m$. El objetivo del segundo vector FDA es resolver el mismo problema, pero considerando los vectores perpendiculares al primer vector FDA, y así sucesivamente. Los vectores FDA así calculados son iguales a los vectores propios w_k del problema de valores propios generalizado:

$$S_b w_k = \lambda_k S_w w_k \quad (8)$$

Donde los valores propios λ_k indican el grado de separabilidad entre las clases al proyectar los datos en w_k .

Esto se puede resolver siempre que S_w sea invertible, lo cual ocurre siempre que el número de datos (observaciones) n , sea mucho mayor que el número de variables medidas, m .

El primer vector FDA es el que está asociado al valor propio más grande, el segundo vector FDA está asociado al segundo valor propio más grande y así sucesivamente. Un valor grande de λ_k indica que cuando los datos de la clase se proyectan en los vectores propios asociados w_k hay una gran separación de la media de la clase relativa a las varianzas de las clases y en consecuencia, una gran separación de las clases a lo largo de la dirección w_k . [8]

Si se define la matriz $W_p \in R^{m \times (p-1)}$ que contiene los $(p-1)$ vectores FDA como columnas, entonces la transformación lineal de los datos del espacio de dimensión m (con todas las variables) a este espacio de dimensión menor $(p-1)$ se describe como.

$$z_i = W_p^T x_i \quad (9)$$

donde $z_i \in R^{(p-1)}$.

El problema ahora es decidir el orden de reducción, es decir el número, p , más adecuado de vectores FDA. Esto se puede realizar de varias formas, cuando se tienen suficientes datos de todas las posibles clases se puede utilizar la correlación cruzada, que consiste en dividir los datos en varios conjuntos, uno llamado de entrenamiento y los otros de test o validación. La técnica de reducción de dimensionalidad FDA se aplica a los datos de entrenamiento, y después el comportamiento del modelo obtenido se valida con los datos de test, y esto se hace para cada orden de reducción, es decir, desde, $a = 1, \dots, p$. Se elige el orden que optimice el comportamiento del modelo para los datos de validación. [8]

Otra forma de calcular el orden cuando tenemos pocos datos es calcular el orden a , que minimice el siguiente criterio:

$$f_m(a) + \frac{a}{\tilde{n}} \quad (10)$$

Donde $f_m(a)$ es la proporción de datos mal clasificados, es decir pertenecientes a una clase y clasificados como de otra clase, para los datos de entrenamiento proyectados sobre los “ a ” vectores FDA, este término tiene en cuenta el error de predicción, mientras que el otro término de la ecuación, a/\tilde{n} se añade para penalizar el incremento de dimensionalidad, es decir, tiene en

cuenta la complejidad del modelo, siendo \bar{n} el número medio de observaciones por clase. [8]

Una vez se ha descrito la forma teórica en la que se basa la clasificación mediante el discriminante de Fischer, se va a pasar a describir la manera de aplicar lo explicado anteriormente a una herramienta de simulación. Se asignará una observación a la clase i , si se cumple que:

$$g_i(x) > g_j(x) \quad \forall j \neq i, \quad (11)$$

Donde a la función $g_j(x)$ se le llama función discriminante para la clase j dado un vector de datos x . La función, $g_j(x)$ que minimiza el error, cuando ocurre el evento w_i (por ejemplo, un fallo) es:

$$g_i(x) = P(w_i | x) \quad (12)$$

Donde $P(w_i | x)$ es la probabilidad a posterior de que x pertenezca a la clase i . Si se usa la regla de Bayes:

$$P(w_i | x) = \frac{p(x | w_i) P(w_i)}{p(x)} \quad (13)$$

Donde $P(w_i)$ es la probabilidad a priori para la clase w_i , $p(x)$ es la función densidad de probabilidad para x , y $p(x | w_i)$ es la función densidad de probabilidad de x condicionada al evento w_i . Si se supone que los datos están normalmente distribuidos:

$$P(x | w_i) = \frac{1}{(2\pi)^{m/2} [\det(\Sigma_i)]^{1/2}} \exp \left[-\frac{1}{2} (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) \right] \quad (14)$$

Con m el número de variables medidas, y μ_i y Σ_i son la media y la matriz de covarianza para la clase i . La función $g_i(x)$ de la ecuación (12) puede ser sustituida por la ecuación (14), y se puede demostrar que la clasificación obtenida con ambas funciones es idéntica:

$$g_i(x) = \ln p(x | w_i) + \ln P(w_i) \quad (15)$$

Si se sustituye la ecuación (14) en la ecuación (15) se obtiene finalmente que la función discriminante es:

$$g_i(x) = -\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) - \frac{m}{2} \ln 2\pi - \frac{1}{2} \ln[\det(\Sigma_i)] + \ln P(w_i) \quad (16)$$

Si se caracteriza esta función para este caso, es decir, sea $W_a \in \mathbb{R}^{m \times a}$ la matriz formada por los vectores propios w_1, w_2, \dots, w_a calculados con la ecuación (8), la función discriminante puede calcularse como:

$$g_j(x) = -\frac{1}{2}(x - \bar{x}_j)W_a \left(\frac{1}{n_j - 1} W_a^T S_j W_a \right)^{-1} W_a^T (x - \bar{x}_j) + \ln(p_j) - \frac{1}{2} \ln \left[\det \left(\frac{1}{n_j - 1} W_a^T S_j W_a \right) \right] \quad (17)$$

donde S_j , \bar{x}_j y n_j ya han sido anteriormente definidas.

3.4. ÁRBOLES DE DECISIÓN.

Los árboles de decisión (Figura 5) son uno de los paradigmas más clásicos y ampliamente usados del mundo del aprendizaje automático. Un árbol de clasificación está formado por nodos, ramas y hojas. Cada nodo representa un test univariado o decisión sobre los valores de un atributo concreto. El primer nodo del árbol es conocido como el nodo raíz. Finalmente están los nodos terminales u hojas en los que se toma una decisión acerca de la clase a asignar. Así, a la hora de clasificar un nuevo caso, tendrán que compararse los valores de los atributos con las decisiones o test que se toman en los nodos, siguiendo la rama que coincida con dichos valores en cada test. Finalmente se llega a un nodo terminal u hoja que predice la clase para el caso tratado. Un árbol de decisión también se puede ver como un conjunto de reglas si-entonces, si bien la diferencia más obvia entre los dos paradigmas es que las reglas de decisión son independientes entre sí, mientras que las reglas extraídas del árbol de decisión no lo son. [9]

Los algoritmos de árbol de decisión desglosan el conjunto de datos mediante la formulación de preguntas hasta conseguir el fragmento de datos adecuado para hacer una predicción. Basado en las características de los datos de entrenamiento, el árbol de decisión “aprende” una serie de factores para inferir las etiquetas de clase de los ejemplos. El nodo de comienzo es la raíz del árbol, y el algoritmo dividirá de forma iterativa el conjunto de datos en la característica que contenga la máxima ganancia de información, hasta que los nodos finales (hojas) sean puros. [9]

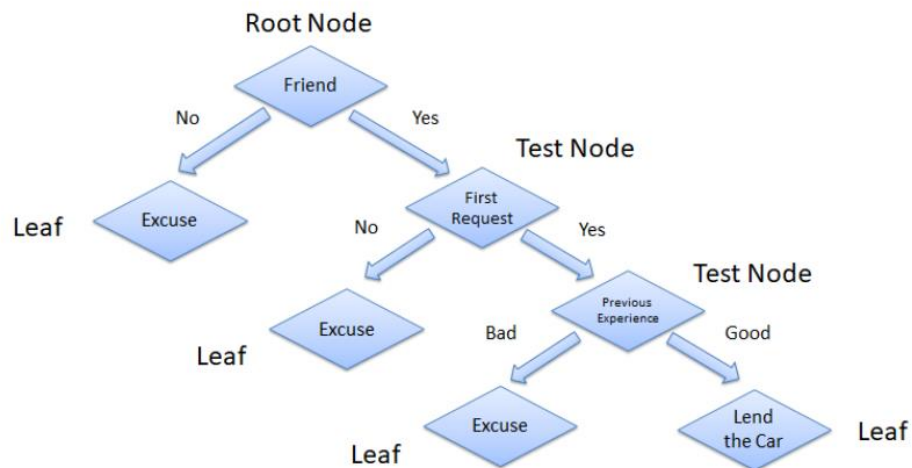


Figura 5. Esquema árbol de decisión. (Fuente: medium.com/datos-y-ciencia)

Los parámetros característicos en este tipo de clasificación son la máxima profundidad, el máximo número de muestras, el mínimo número de muestras y el máximo número de características.

La máxima profundidad es la mayor longitud desde la raíz a las hojas. Una gran profundidad puede causar sobreajuste, y pequeña profundidad puede causar sub-ajuste. Para evitar sobreajuste, 'podaremos' el árbol de decisión estableciendo un hiper-parámetro con la máxima longitud.

El máximo número de muestras está relacionado con la cantidad de nodos. Cuando cortamos un nodo, se podría tener el problema de conseguir 99 muestras en uno de los cortes y 1 muestra en el otro, lo que sería un mal uso de los recursos, para evitarlo, podemos establecer un máximo para el número de muestras que permitimos para cada hoja. Esto se puede especificar como un entero o como un número flotante. Un pequeño número de muestras caerá en sobreajuste, mientras que un gran número de muestras caerá en sub-ajuste. El mínimo número de muestras se define de manera análoga al máximo solo que en este caso se toman los valores mínimos. [9]

El máximo número de características es un parámetro clave dado que muy frecuentemente tendremos muchas características (columnas) para construir un árbol. En cada corte, tendremos que hacer revisar todo el conjunto de datos en cada una de las características, lo que puede ser muy costoso. Una solución a este problema es limitar el número de características que se buscan en cada corte. Si este número es suficientemente alto, es probable que encontremos una buena característica entre aquellas que buscamos (aunque

pueda no ser la perfecta). Sin embargo, si no es tan alto como el número total de características, la velocidad de los cálculos se elevará de manera significativa. [9]

3.4.1. RANDOM FOREST

Dentro de los árboles de decisión hay un tipo de ellos denominados Random Forest. Este tipo de clasificación se utiliza cuando tenemos un conjunto de datos con muchas características (columnas), el algoritmo del árbol de decisión tiende a sobre-ajustar, añadiendo complejidad al modelo y el proceso de aprendizaje [10]

Este clasificador es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. Es una modificación sustancial de la técnica de bagging que construye una larga colección de árboles no correlacionados y luego los promedia. El algoritmo para inducir un random forest fue desarrollado por Leo Breiman y Adele Cutler. El término aparece en la primera propuesta de Random decision forests, hecha por Tin Kam Ho de Bell Labs en 1995. El método combina la idea de bagging de Breiman y la selección aleatoria de atributos, introducida independientemente por Ho, Amit y Geman, para construir una colección de árboles de decisión con variación controlada. La selección de un subconjunto aleatorio de atributos es un ejemplo del método random subspace, el que, según la formulación de Ho, es una manera de llevar a cabo la discriminación estocástica propuesta por Eugenio Kleinberg. En muchos problemas el rendimiento del algoritmo random forest es muy similar a la del boosting, y es más simple de entrenar y ajustar. Como consecuencia, el Random forest es popular y ampliamente utilizado. [10]

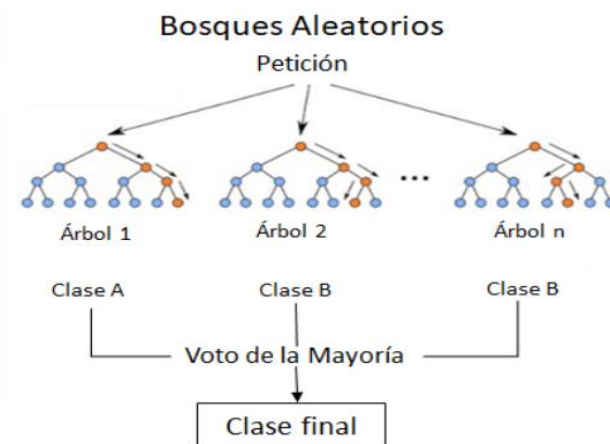


Figura 6. Esquema Bosques Aleatorios. (Fuente: medium.com/datos-y-ciencia)

Se desarrolla un algoritmo de agrupación de aprendizaje que combinará una serie de modelos más débiles para crear otro más robusto.

El algoritmo realizará los siguientes pasos:

- Diseñar una muestra de arranque de tamaño n .
- Desarrollar un árbol de decisión desde la muestra de arranque. En cada nodo habrá características seleccionadas aleatoriamente sin reemplazamiento y el nodo se cortará maximizando la ganancia de información.

El proceso previo se repetirá K veces. Agregar la predicción hecha para cada árbol, asignando la etiqueta de clase por votación mayoritaria.

La principal ventaja de este método es que normalmente no necesitaremos podar el bosque aleatorio (ya que el modelo es muy resistente al ruido). Sin embargo, es mucho menos interpretable que los árboles de decisión. El único hiperparámetro que necesitaremos ajustar es el número de árboles K . Normalmente, cuanto más grande es K , mejor se comportará el modelo, pero esto incrementará drásticamente el esfuerzo de computación y por tanto, el coste. [9]

3.5. MÁQUINAS DE VECTORES SOPORTE O SVM.

Se denomina SVM a SUPPORT VECTOR MACHINE, es decir a una máquina de vector soporte. Son un conjunto de algoritmos supervisados desarrollados por Vladimir Vapnik y los laboratorios AT&T. Estos métodos están propiamente relacionados con problemas de clasificación y regresión. Dado un conjunto de ejemplos de entrenamiento (de muestras) podemos etiquetar las clases y entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra. Intuitivamente, una SVM es un modelo que representa a los puntos de muestra en el espacio, separando las clases a 2 espacios lo más amplios posibles mediante un hiperplano de separación definido como el vector entre los 2 puntos, de las 2 clases, más cercanos al que se llama vector soporte. Cuando las nuevas muestras se ponen en correspondencia con dicho modelo, en función de los espacios a los que pertenezcan, pueden ser clasificadas a una o la otra clase. Una SVM construye un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alta (o incluso infinita) que puede ser utilizado en problemas de clasificación o regresión. Una buena separación entre las clases permitirá una clasificación correcta. [9]

Este algoritmo puede ser considerado una extensión del algoritmo "perceptron". En SVM (Figura 7) el objetivo de la optimización es establecer una línea de decisión que separe las clases maximizando el margen entre esta línea

y los puntos de muestra cercanos a este hiperplano, estos puntos son los denominados vectores soporte.

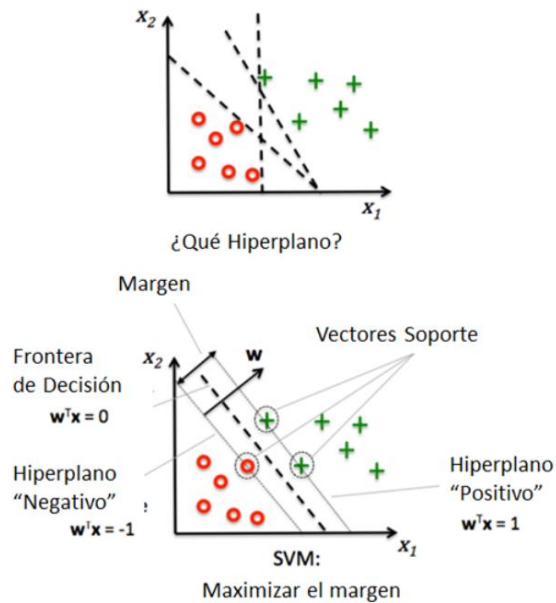


Figura 7. Esquema SVM. (Fuente: medium.com/datos-y-ciencia)

Para establecer los márgenes máximos, se añaden dos rectas paralelas (márgenes) e intentamos maximizar sus distancias a la línea de decisión original. Tendremos en cuenta los puntos sin clasificar (errores) y los que quedan entre los márgenes de la línea. Normalmente, las líneas de decisión con márgenes grandes tienden a tener un error de generalización menor. Por otro lado, los modelos con márgenes pequeños tienen menor tendencia al "sobreajuste" (overfitting).^[9]

La función de maximización (margen de error) se calcula de la siguiente forma:

Los hiperplanos positivos y negativos se pueden expresar como:

$$w_0 + w^T \cdot x_{positivas} = 1 \quad (18)$$

$$w_0 + w^T \cdot x_{negativas} = -1 \quad (19)$$

Restando 18 y 19 entre sí:

$$w^T \cdot (x_{positivas} - x_{negativas}) = 2 \quad (20)$$

Normalizando la ecuación 20, con la longitud del vector w , se obtiene la ecuación del margen de error.

$$\frac{w^T \cdot (x_{positivas} - x_{negativas})}{\|w\|} = \frac{2}{\|w\|} \quad (21)$$

La parte izquierda de la ecuación se interpreta como la distancia entre los hiperplanos positivos y negativos, en otras palabras, el margen que estamos tratando de maximizar. En la práctica, es más fácil minimizar el término recíproco, que se puede resolver con programación cuadrática, que se encuentra expresado en la ecuación 22. ^[11]

$$margen = \frac{1}{2} \|w\|^2 \quad (22)$$

Se define la variable de holgura ξ , que se usa para una clasificación menos restrictiva, es decir, con márgenes más suaves. Su propósito era la necesidad de suavizar las restricciones lineales, con el propósito de permitir la convergencia de los procesos de optimización cuando se trata con datos separables no lineales. Se añade a las restricciones lineales, y la nueva función de minimización se convierte en:

$$error = \frac{1}{2} \|w\|^2 + C(\sum_i \xi(i)) \quad (23)$$

Modificando la variable C de la función, podemos controlar la penalización por fallos en la clasificación, y controlar la anchura del margen, ajustando de esta manera la compensación de parcialidad. La mejor línea de frontera dependerá del problema que estamos tratando de resolver. Si nos enfrentamos a un problema médico, no queremos ningún error, sin embargo, podremos estar dispuestos a aceptar cierto nivel de error cuando trabajemos con otra clase de problemas. Valores C grandes corresponden a penalizaciones altas de error. Valores C pequeños implican que somos poco estrictos respecto a los fallos en la clasificación. Se puede observar en la figura 8. ^[11]

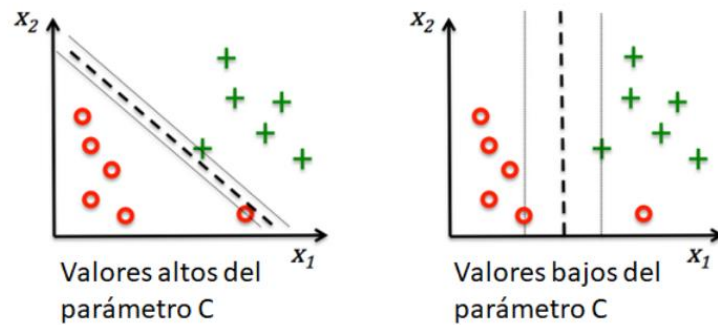


Figura 8. Explicación parámetro C. (Fuente: medium.com/datos-y-ciencia)

Esta es la manera más simple de realizar la separación mediante una línea recta, un plano recto o un hiperplano N-dimensional. Desafortunadamente los universos a estudiar no se suelen presentar en casos idílicos de dos dimensiones, figura 8, sino que un algoritmo SVM debe tratar con más de dos variables predictoras, curvas no lineales de separación, casos donde los conjuntos de datos no pueden ser completamente separados, clasificaciones en más de dos categorías, etc. Debido a las limitaciones computacionales de las máquinas de aprendizaje lineal, éstas no pueden ser utilizadas en la mayoría de las aplicaciones del mundo real. La representación por medio de funciones Kernel ofrece una solución a este problema, proyectando la información a un espacio de características de mayor dimensión el cual aumenta la capacidad computacional de las máquinas de aprendizaje lineal. [11]

Por tanto, las SVM son una rama popular de algoritmos, ya que se pueden utilizar para resolver problemas de clasificación no lineales. Esto se realiza con un método denominado “kernelización”. La idea básica es el uso de un kernel, cuando tratamos con combinaciones no lineales de las características originales, para proyectarlas en un espacio con más dimensiones vía una función de correspondencia ϕ , de forma que los datos sean linealmente separables en este nuevo espacio, normalmente de mayor dimensión. Intuitivamente, el conjunto de datos original se transforma en otro de más dimensiones, y después se aplica una proyección para hacer las clases separables. Por último, se aplica el algoritmo, se separan las clases y se aplica la función inversa a la que provoca la proyección para volver a la distribución original de los datos (Figura 9). [9]

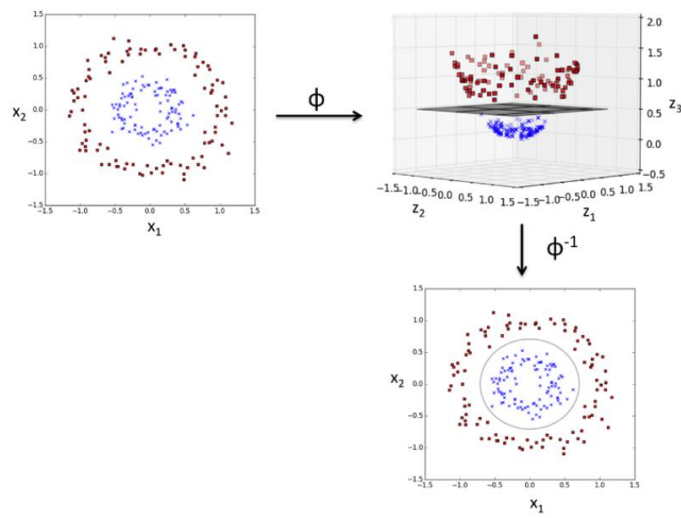


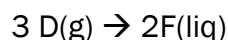
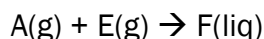
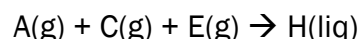
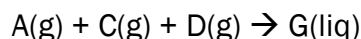
Figura 9. Kernel SVM. (Fuente: medium.com/datos-y-ciencia)

4. PLANTA TENNESSE EASTMAN.

Los datos que se van a utilizar para este trabajo son extraídos del proceso Tennessee Eastman (TEP) el cual ha sido ampliamente utilizado para numerosas pruebas de métodos de detección y diagnóstico de fallos.

El proceso Tennessee Eastman, que constituye un problema de control, fue propuesto por Downs y Vogel en 1993. Se vieron en la necesidad de crear una planta química hipotética basada en que la compañía Eastman Chemical desarrolló con el objetivo de evaluar técnicas de control de procesos y métodos de supervisión y diagnóstico y poder extrapolarlo a otras plantas o a otro tipo de procesos. [12]

El proceso ficticio produce dos compuestos a partir de cuatro reactivos. También presenta un inerte y una purga, por lo que en total se tienen ocho componentes, A-H. Las reacciones que se llevan a cabo son las siguientes:



Todas las reacciones son reversibles y exotérmicas y su equilibrio depende de la temperatura. Se pueden asemejar todas las cinéticas de reacción de todas las reacciones del proceso a cinéticas de orden uno. Además, esto hace que se respeten las concentraciones obtenidas de producto. [13]

El proceso tiene cinco operaciones clave, el reactor, el condensador, el compresor, el separador y el destilador. Se puede observar en la figura 10 todas las operaciones citadas y los componentes mencionados anteriormente.

Los reactivos gaseosos se alimentan al reactor donde se convierten en productos líquidos, por lo que la reacción ocurre en fase gaseosa. La reacción esta catalizada por un disolvente no volátil que se encuentra en la fase líquida. El reactor contiene un intercambiador de calor para su refrigeración, dado que la reacción genera gran cantidad de calor. [13]

Los productos que abandonan el reactor se encuentran en estado vapor. También sale como productos parte de los productos alimentados que no han reaccionado. Estos productos pasan a través de un enfriador para condensarse. Después de la condensación pasan a un separador de líquido y vapor. Los productos no condensados se recirculan mediante un compresor centrífugo y vuelven a alimentar al reactor. Los productos condensados se van

a la columna de destilación, la cual elimina las pequeñas cantidades de producto que pudieran quedar. La columna es alimentada también por una corriente de vapor pura.^[13]

Los productos G y H salen por la parte baja de la columna. Se llevará a cabo una separación de estos productos, pero esta operación no se encuentra dentro de los procesos de estudio.

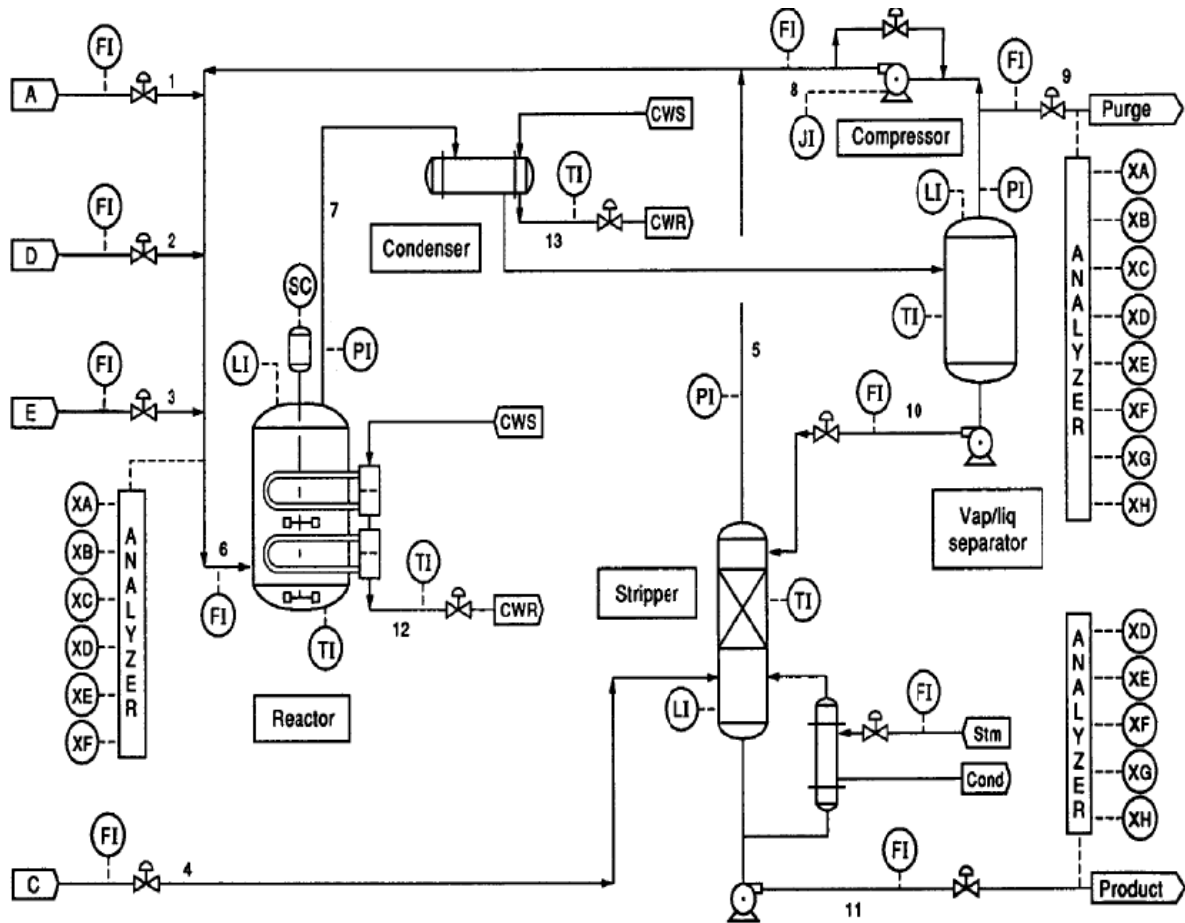


Figura 10. Esquema Planta TEP.

El proceso tiene 12 variables que pueden ser manipuladas y 41 instrumentos de medición, en los cuales se pueden ver los distintos cambios que se producen al cambiar las variables. En la figura 11 y en la figura 12 se encuentran las variables manipuladas y las medidas de los instrumentos respectivamente.^[13]

Manipulated (input) variables of the TE process.

XMV	Description
XMV(1)	Valve position feed component D (stream 2)
XMV(2)	Valve position feed component E (stream 3)
XMV(3)	Valve position feed component A (stream 1)
XMV(4)	Valve position feed component A & C stream4
XMV(5)	Valve position compressor re-cycle
XMV(6)	Valve position purge (stream 9)
XMV(7)	Valve position underflow separator stream10
XMV(8)	Valve position underflow stripper (stream 11)
XMV(9)	Valve position stripper steam
XMV(10)	Valve position cooling water outlet of reactor
XMV(11)	Valve position cooling water outlet of separator
XMV(12)	Rotation of agitator of reactor

Figura 11. Variables manipuladas TEP.

Measured (output) variables of the TE process.

XMEAS	Description
1	Feed flow component A (stream 1)
2	Feed flow component D (stream 2)
3	Feed flow component E (stream 3)
4	Feed flow components A & C (stream 4)
5	Recycle flow to reactor from separator (stream 8)
6	Reactor feed (stream 6)
7	Reactor pressure
8	Reactor level
9	Reactor temperature
10	Purge flow (stream 9)
11	Separator temperature
12	Separator level
13	Separator pressure
14	Separator underflow (liquid phase)
15	Stripper level
16	Stripper pressure
17	Stripper underflow (stream 11)
18	Stripper temperature
19	Stripper steam flow
20	Compressor work
21	Reactor cooling water outlet temperature
22	Condenser cooling water outlet temperature
23–28	Concentration in Reactor feed (stream 6), Components A through F
29–36	Concentration in Purge (stream 9), Components A through H
37–41	Concentration in stripper underflow (stream 11), Components D through H

Figura 12. Variables Medidas TEP.

Para obtener los datos, que más adelante usaremos para la simulación, se toman datos de las variables en intervalos de 3 minutos. Se almacenan los datos en dos conjuntos de datos, el de entrenamiento y el de prueba.

Cada simulación consta de 22 ensayos. El primer ensayo se realiza para un comportamiento normal de la planta, es decir sin que haya fallo. Los ensayos restantes, 21, se realizan cada uno para un tipo de fallo. Se tienen dos conjuntos de datos para cada situación, uno para entrenamiento y otro para validación. La simulación de los datos entrenamiento se realizan durante 25 horas y la de los datos de prueba se realiza durante 48 horas. Estos datos fueron generados por y son públicos pudiéndose bajar de la página web: <http://web.mit.edu/braatzgroup/links.html>. En la figura 13 se encuentran los 21 tipos de fallos que se pueden dar en la planta. ^[13]

Process faults for the Tennessee Eastman process.

Fault	Type	Disturbed value
IDV(1)	Step	A/C-ratio of stream 4, B composition constant
IDV(2)	Step	B composition of stream 4, A/C-ratio constant
IDV(3)	Step	D feed (stream 2) temperature
IDV(4)	Step	Cooling water inlet temperature of reactor
IDV(5)	Step	Cooling water inlet temperature of separator
IDV(6)	Step	A feed loss (stream 1)
IDV(7)	Step	C header pressure loss (stream 4)
IDV(8)	Random	A/B/C composition of stream 4
IDV(9)	Random	D feed (stream 2) temperature
IDV(10)	Random	C feed (stream 4) temperature
IDV(11)	Random	Cooling water inlet temperature of reactor
IDV(12)	Random	Cooling water inlet temperature of separator
IDV(13)	Drift	Reaction kinetics
IDV(14)	Sticking	Cooling water outlet valve of reactor
IDV(15)	Sticking	Cooling water outlet valve of separator
IDV(16)	Random	(unknown); deviations of heat transfer within stripper (heat exchanger)
IDV(17)	Random	(unknown); deviations of heat transfer within reactor
IDV(18)	Random	(unknown); deviations of heat transfer within condenser
IDV(19)	Sticking	(unknown); recycle valve of compressor, underflow separator (stream 10), underflow stripper (stream 11) and steam valve stripper
IDV(20)	Random	(unknown)
IDV(21)	Constant	The valve for Stream 4 was fixed at the steady state position

Figura 13. Fallos provocados TEP.

5. SIMULACIÓN.

En este capítulo describiremos el proceso que se ha seguido para desarrollar el sistema de clasificación de fallos realizado en este trabajo, se presentarán los resultados obtenidos en las simulaciones, se analizarán, se compararán y se discutirá sobre los mismos.

Para clasificar los fallos se plantearon diferentes enfoques. Primero se realizó una clasificación usando redes neuronales de dos tipos, red perceptron multicapa y una red probabilística. Después se utilizó un clasificador lineal estadístico, el análisis discriminante de Fisher (FDA), después se usaron nuevos enfoques sacados de la inteligencia computacional como son las máquinas de vectores soporte (SVM) y los árboles de decisión, en concreto "ramdon forest" (RF). Se analizará el resultado de clasificación de todos ellos y se hará una comparativa para saber cuál es el mejor para esta tarea de clasificación.

5.1. DATOS DE LA PLANTA.

Para la realización de la parte experimental o de simulación de este trabajo se han cogido ficheros de datos de la planta descrita en el tercer capítulo de este trabajo. La planta cuenta con 21 tipos de fallos, como se ha comentado anteriormente, pero en este trabajo solo se usarán 4 de ellos: fallo1 que consiste en un cambio en la relación entre los componentes A/C del flujo 4, fallo2 que consiste en un cambio en la composición de B en el flujo 4, fallo4 que consiste en un salto en la temperatura del agua de refrigeración del reactor y fallo5 que es un salto en la temperatura del agua de refrigeración del separador.

Se trabajará con dos tipos de ficheros de datos, uno para diseñar y testear los clasificadores que consta de 960 observaciones de las 52 variables de la planta y otro para probar la eficacia de los clasificadores. Habrá dos ficheros de datos por cada tipo de fallo, por lo que en total se cuenta con ocho ficheros de datos.

En los datos usados para el diseño y el entrenamiento de los clasificadores se conoce el tipo de fallo y de las 960 observaciones que se tienen, las primeras 160 son datos de comportamiento normal de la planta, es decir no son datos de fallo. A partir de la fila 161 son datos de fallo. Para diseñar los clasificadores se usarán únicamente datos de fallo, más en concreto los datos de fallo que van desde la fila 161 hasta la fila 750, es decir 590 observaciones se usan para entrenar los clasificadores. El resto de los datos, de la fila 750 hasta la 960, se utilizará para testear los clasificadores.

Un segundo conjunto de datos de cada fallo se usará para testear los clasificadores, en este caso se tienen únicamente 480 observaciones, todas de fallo, de las mismas 52 variables. De estas observaciones se eliminarán las diez primeras, que son los datos de puesta en marcha de la planta y con estos datos el clasificador no funcionaría correctamente.

5.2. REDES NEURONALES.

Como se ha descrito anteriormente, se utilizan redes neuronales perceptron multicapa. Lo primero que se realiza para poder llevar a cabo la clasificación, es la creación de la red neuronal, para posteriormente poder utilizarla.

Para crear la red neuronal, primero tenemos que decidir cuál son las entradas a la red. En este caso con todas las variables (52) y todos los datos de entrenamiento de cada fallo puestos unos detrás de otros. Es decir, se crea una matriz de entrada formada por 52 variables y (590*4) observaciones, donde las primeras 590 son de fallo 1, las siguientes 590 son de fallos 2 y así sucesivamente.

Después se deben definir la salida de la red, y cuántas neuronas hay en esta capa de salida. En este caso como hay 4 fallos se definen 4 neuronas en la capa de salida. También se necesita la salida deseada de la red neuronal para poder entrenarla. Esta salida deseada será una matriz formada por 4 variables y 590*4 observaciones. Esta matriz se diseña de la manera que el usuario quiera que la red clasifique los datos. En las redes neuronales multicapa se quiere tener una salida que valga 1 para el tipo de fallo que se quiere detectar y cero para el resto. Por tanto, tendremos una salida en la que la primera neurona detecte el fallo 1, es decir, vale 1 siempre que los datos de entrada sean de fallo 1 y cero en cualquier otro caso. La segunda neurona detectará el fallo 2 y su salida deseada será 1 cuando en la entrada los datos sean de fallo 2 y cero en todos los demás y así todas las demás. Por tanto, en la primera fila, los 590 primeros datos, tendrán un valor de 1 y cero el resto. En la segunda fila, a partir del 591 y hasta el 1181 valdrán 1 y cero todos los demás y así sucesivamente.

La matriz de datos de salida es de vital importancia, dado que es la que dirá qué tipo de fallo es. Según se ha diseñado para la red neuronal, si es fallo tipo 1, la primera neurona de la capa de salida nos da un 1 y cero el resto de las neuronas, si es fallo tipo 2, la segunda neurona tendrá valor uno y el resto cero, si es fallo tipo 4, la tercera neurona valdrá uno y el resto cero y para fallo tipo 5 la cuarta neurona es la que valdrá uno. Esto se observará en la salida de la red, mostrada en la figura 2 con los datos de entrenamiento, y en las

simulaciones que se realizarán para los datos de test y para los datos no conocidos que permitirán validar la red.

Un segundo paso en todo método basado en datos es el pre-procesamiento de la información, como filtrar el ruido, eliminar datos espurios, etc., en este caso sólo es necesario normalizar los datos de entrada entre $[-1, 1]$. Después ya se crea la red, definiendo el número de neuronas en la capa oculta, que será el único parámetro de diseño para crear diferentes redes y quedarnos con la mejor. Después de varias pruebas la mejor selección fue con 10 neuronas en la capa oculta. Y el último paso es el entrenamiento de la red, que consiste en modificar los pesos y las bias de la red para obtener la salida deseada. En figura 14, se observa un esquema y el resultado de la red entrenada.

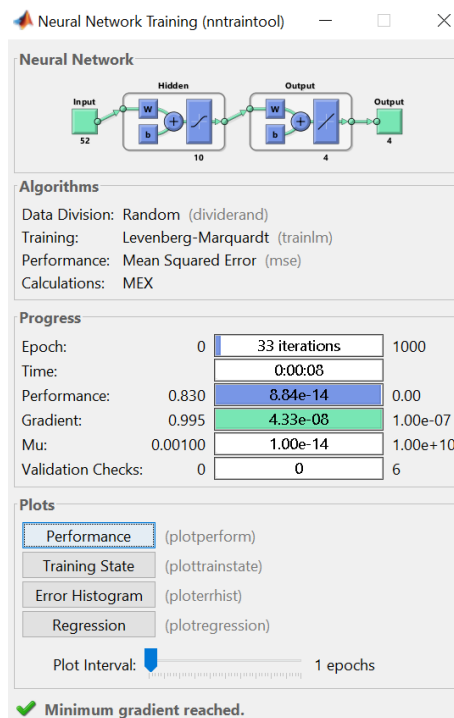


Figura 14. Red neuronal.

En la figura 15 se observa cómo es la salida de la red con los datos de entrenamiento. Se ve, que clasifica los datos de una manera muy apropiada. Salvo pequeñas oscilaciones, para el segundo y cuatro gráficos, en las que no se tiene un valor de cero exacto cuando no hay fallo. Las oscilaciones son muy pequeñas, dado que en ningún momento llegan a aproximarse siquiera a valores de 0,5.

Una vez que hemos creado, la red hay que probarla con los datos de test para ver si es capaz de generalizar, es decir, que de un buen resultado con datos que no haya visto en el entrenamiento, Para ello, se crea nuevamente la matriz de entrada a la red con los datos de test. Estos datos se tienen que normalizar entre $[-1 \ 1]$ con la normalización usada en el entrenamiento, Ahora, se va a probar la red con los distintos tipos de fallo por separado para ver su comportamiento.

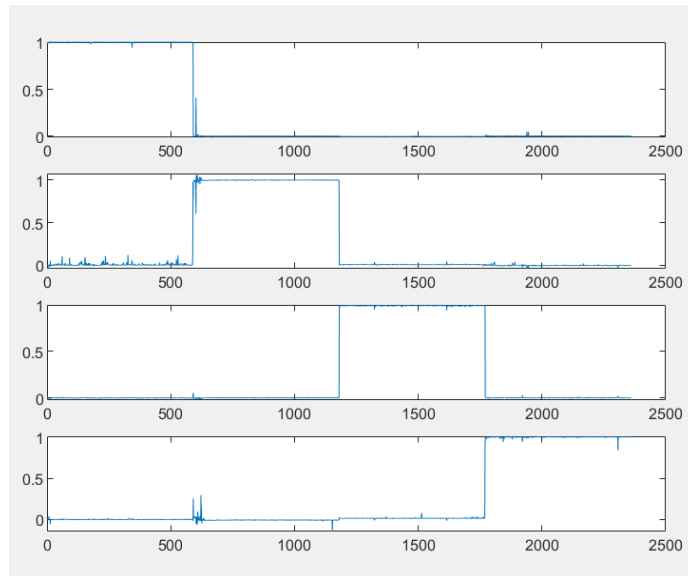


Figura 15. Gráficos de salida de la red con los datos de entrenamiento.

La salida para el fallo 1 se encuentra en la figura 16. Como se puede observar, es fallo 1, dado que vale 1 en la primera gráfica, y en las restantes vale 0. En este caso, al ser datos de test, sabíamos el tipo de fallo que iba a ser. La clasificación es extremadamente buena, porque el valor de la primera neurona o salida de la red se encuentra en su mayor parte en uno o en un intervalo muy próximo a este y las otras neuronas o salidas de la red toman valores muy cercanos a cero,

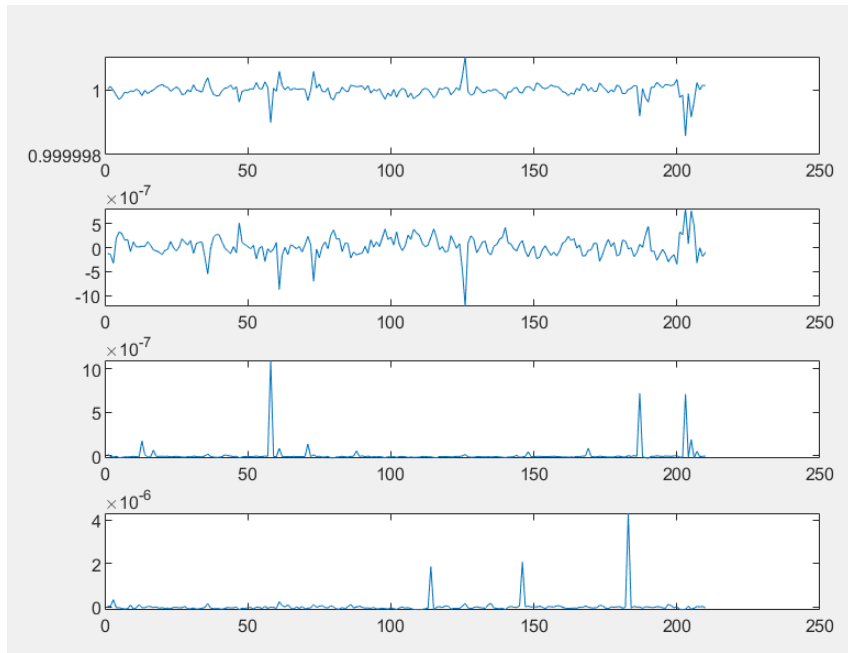


Figura 16. Gráficas salida clasificación fallo 1 con datos de test.

La salida para el fallo 2, se puede observar en la figura 17.

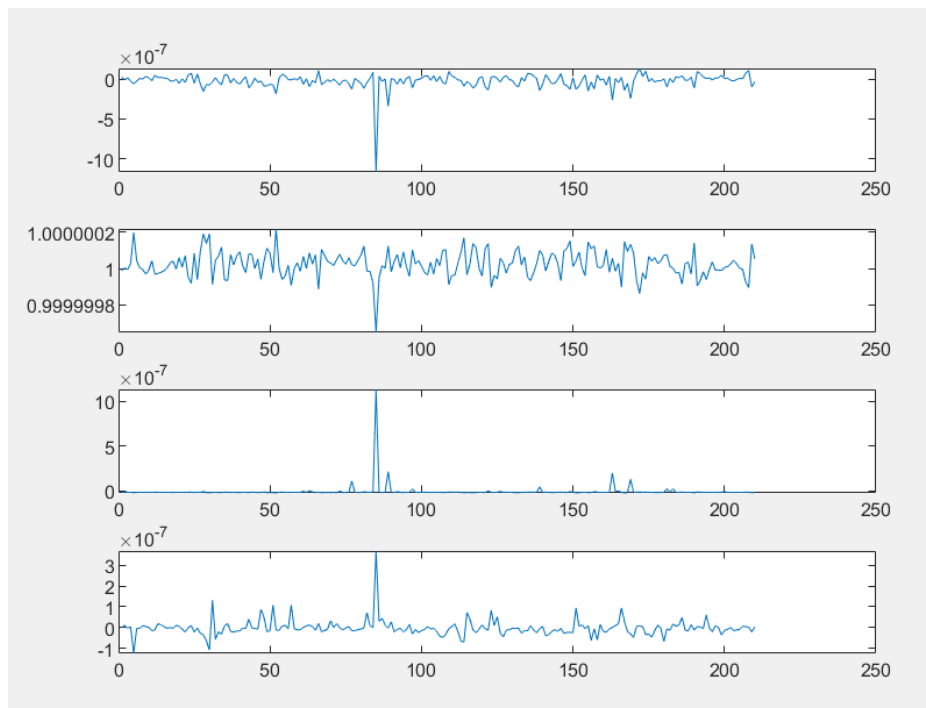


Figura 17. Gráficas salida clasificación fallo 2 con los datos de test.

La salida para el fallo 4, se puede observar en la figura 18.

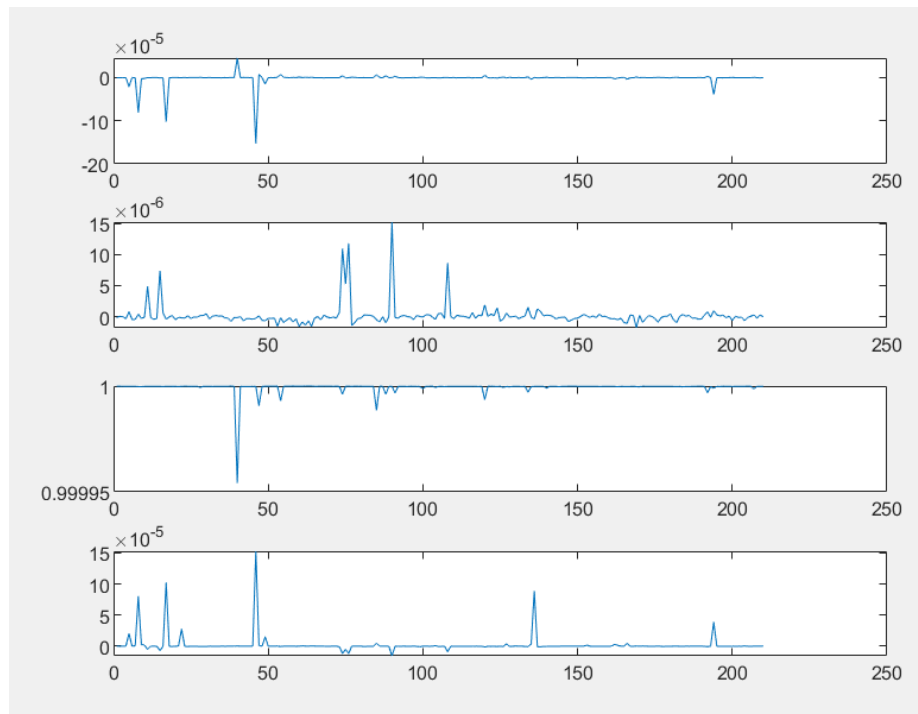


Figura 18. Gráficas salida clasificación fallo 4 con datos de test.

La salida para el fallo 5, se puede observar en la figura 19.

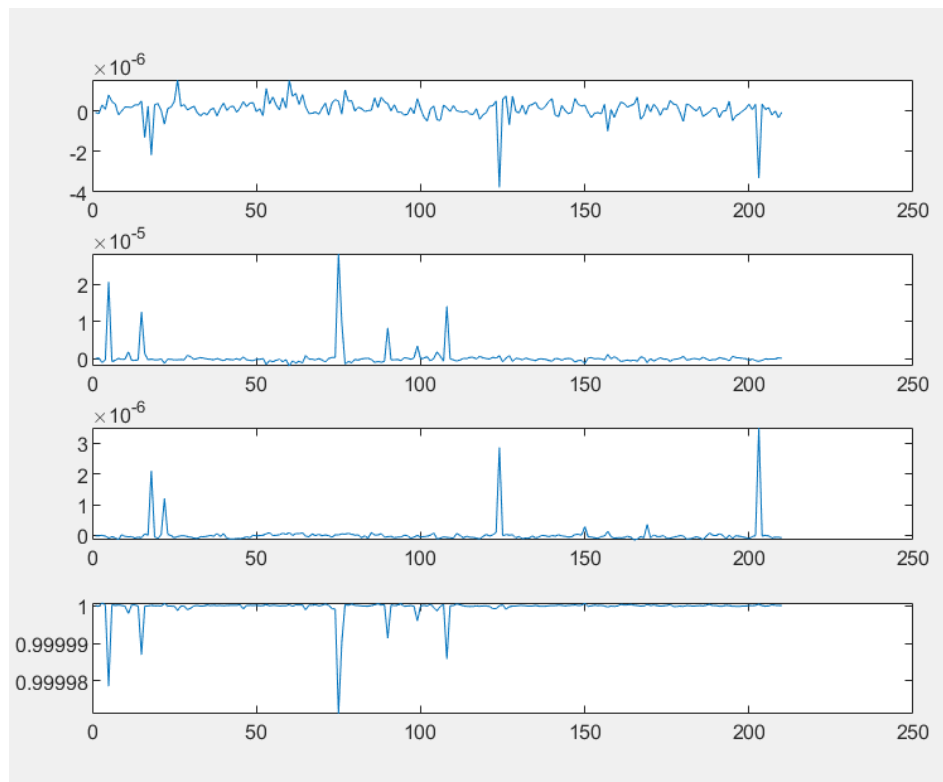


Figura 19. Gráficas salida clasificación fallo 5 con datos de test.

Observando las gráficas de las figuras 17,18 Y 19 se puede llegar a las mismas conclusiones que se han llegado para el fallo 1. La clasificación es muy buena usando los datos de test.

A continuación, se realiza una última simulación utilizando la red neuronal multicapa diseñada. Esta simulación se realiza para datos no conocidos, cogidos de la planta real para ver si somos capaces de clasificar los fallos que ocurren en la planta de forma adecuada. La salida de la red neuronal con el primer fichero de datos desconocidos se observa en la figura 20.

Los datos que se encontraban en este fichero son datos de tipo 1. El valor 1 aparece en la primera neurona o salida de la red, mientras que en el resto vale cero. Fijándose de manera detallada, se observa que aparecen varios datos de fallo tipo 1 en la cuarta gráfica. Pero esta gráfica nos indica que el fallo que ocurre es el fallo 1, por lo que está bien clasificado,

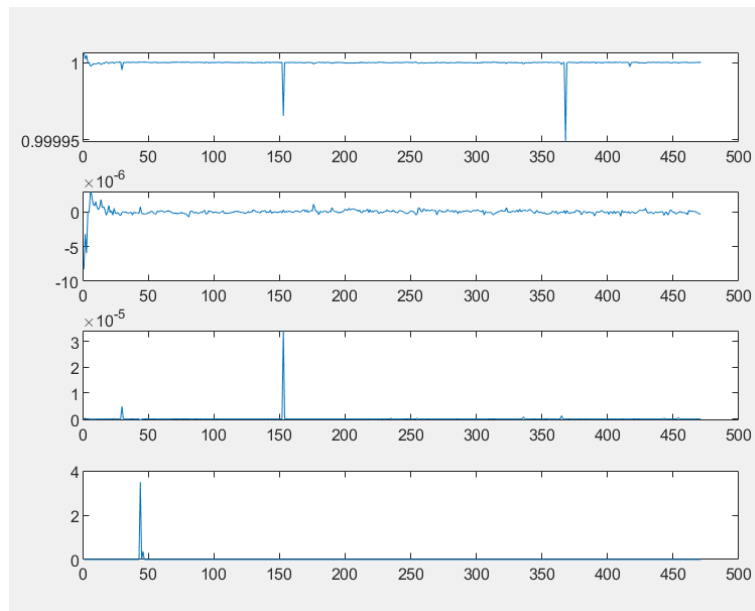


Figura 20. Gráficas de la salida de la red neuronal con el primer fichero de datos.

Ahora se pasa a analizar la salida de la red con los datos del fichero dos que se encuentran en la figura 21.

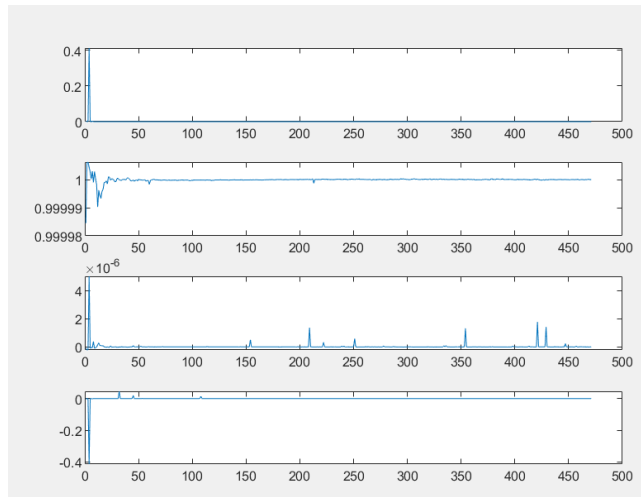


Figura 21. Gráficas de la salida de la red con los datos del fichero 2.

Los datos del fichero dos, se pueden considerar datos de fallo dos, dado que el valor 1 aparece en la segunda gráfica y en el resto de los casos vale cero. Se pueden observar unas pequeñas desviaciones al principio de la simulación, pero mucho menores que las que aparecen en la salida de datos de fallo1. Para esta situación podríamos decir que la clasificación es análoga a la que se da con los datos de entrenamiento y de test.

Las gráficas en las que se ven los datos de salida del fichero de datos tres se encuentran en la figura 22.

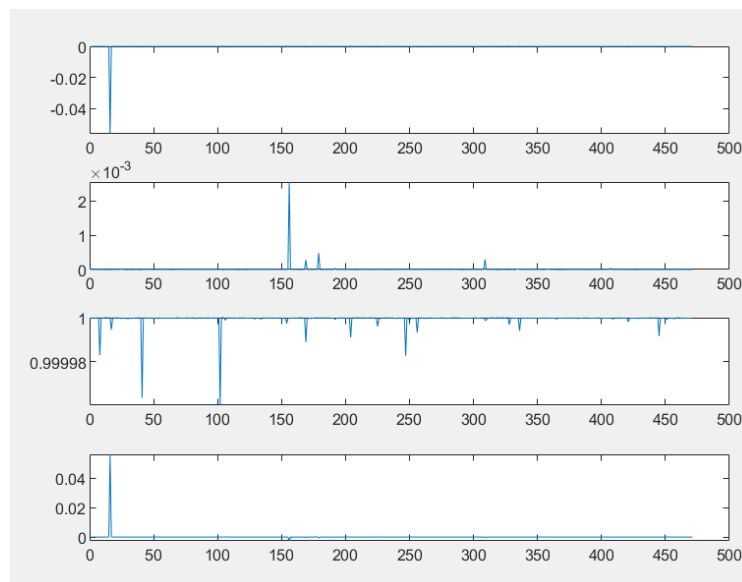


Figura 22. Gráficas de la salida de la red con los datos del fichero 3.

Se llega a la conclusión de que es fallo 4, porque el valor 1 se encuentra en salida 3. Aunque aparecen más oscilaciones entorno al valor 1, que las que se ocurren para fallo 1 y fallo 2, hay menos oscilaciones en torno al valor de cero en las otras gráficas. La clasificación se da como buena.

La salida de la red con los datos del fichero cuatro se observa en la figura 23. Se puede observar que es fallo 5, dado que, en la cuarta gráfica, todos los datos valen 1, o toman un valor muy cerca de este. En el resto de las gráficas, los valores toman valores muy pequeños y muy cercanos a cero, por lo que la red neuronal clasifica de manera adecuada los fallos de tipo 5. Aparecen algunas oscilaciones, en varios puntos de las gráficas, pero nada destacables.

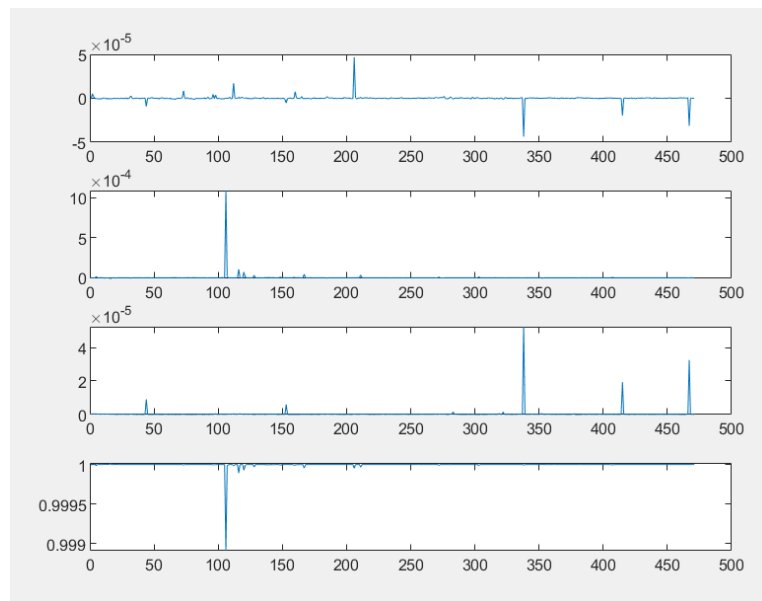


Figura 23. Gráficas de la salida de la red neuronal con datos del fichero 4.

En la simulación para los datos no conocidos se calcula también el porcentaje de clasificación de manera analítica. Se considera que un dato está bien clasificado cuando tiene un valor mayor de 0,5. Los porcentajes de los distintos fallos se encuentran en la tabla 1.

Tabla 1. Porcentaje clasificación Red Neuronal datos no conocidos.

	Porcentaje F1	Porcentaje F2	Porcentaje F4	Porcentaje F5
Matriz 1	99,7877	0	0	0,2123
Matriz 2	0	100	0	0
Matriz 3	0	0	100	0
Matriz 4	0	0	0	100

Se puede observar que todos los datos de fallo 2, fallo 4 y fallo 5 se clasifican de manera correcta y su salida es como se esperaba. Los datos de fallo 1, se clasifican bien, aunque una parte muy baja de estos datos son clasificados como fallo 5.

5.3. REDES NEURONALES PROBABILÍSTICAS.

El primer paso que hay que llevar a cabo para la clasificación de datos de fallo mediante redes neuronales probabilísticas (RP) o redes de probabilidad es, al igual que en las redes neuronales preceptron multicapa, la creación del clasificador, en este caso basado en una red probabilística.

Como se ha visto en el capítulo dos de este mismo trabajo, las redes neuronales y las redes probabilísticas tienen una cierta semejanza, por lo tanto, su diseño se hará también de manera similar. Las redes probabilísticas tienen una capa de entrada, que serán los datos de entrada, y se genera una capa de salida con una única neurona, por lo que la salida de esta red se verá representada, por una única fila.

Lo primero que hay que realizar para crear la red neuronal, es definir cuáles son las entradas a la red. En este caso con todas las variables (52) y todos los datos de entrenamiento de cada fallo puestos unos detrás de otros. Es decir, se crea una matriz de entrada formado por 52 variables y (590*4) observaciones, donde las primeras 590 son de fallo 1, las siguientes 590 son de fallos 2 y así sucesivamente.

Una vez definida la entrada de la red, hay que definir la salida de la misma. En este caso los datos de salida estarán definidos en un vector, es decir una única fila. Este vector tendrá la forma que el usuario desee y será el encargado de asociar los datos de entrada a un tipo de fallo y tendrá una dimensión de $1 \times m$. Al crear la red para clasificar 4 tipos de fallos, se asocia un valor a cada clase de fallo. Los primeros valores, hasta el 590, valdrán 1. Los valores del 590 hasta el 1180 tendrán un valor de 2. Los siguientes 590 valdrán 3 y los últimos 590 datos valdrán 4. De tal manera que los 590 primeros valores irán asociados a datos de fallo de tipo 1, los 590 siguientes datos estarán asociados a datos de fallo tipo 2, los siguientes a fallo tipo 4 y los siguientes a fallo tipo 5. La figura 24 muestra la forma que tiene el vector de los datos de salida de la red.

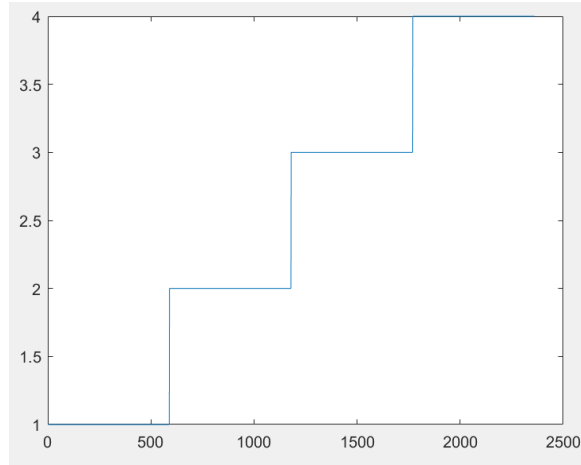


Figura 24. Gráfica de datos de la salida deseada para el diseño red probabilística.

Según el diseño, el vector de los datos de salida definidos, cuando el fallo de entrada en la red probabilística sea tipo 1, la salida de la red dará un 1. Cuando los datos de entrada sean de tipo 2, la salida de la red dará un 2, cuando los datos de entrada sean de tipo 4, la salida mostrará un 3 y por último cuando los datos de entrada sean de tipo 5, la salida mostrará un cuatro. La salida de la red para los datos de entrenamiento se encuentra en la figura 25, como es de esperar, la clasificación es perfecta y no aparecen oscilaciones.

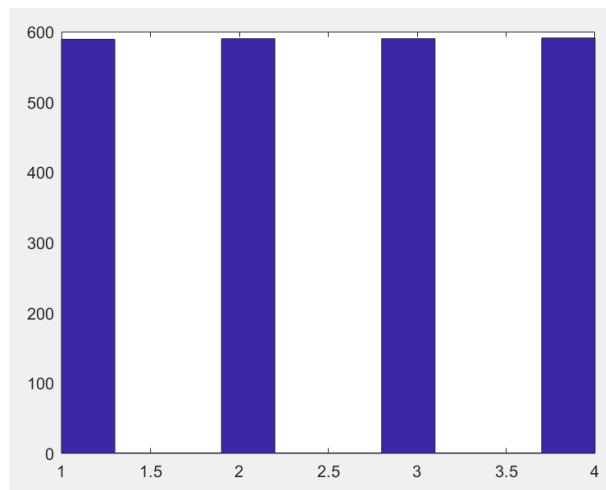


Figura 25. Histograma de clasificación de fallos de la red probabilística con datos de entrenamiento.

Una vez se ha realizado la simulación para crear la red probabilística, se pasa a probarla con los datos de test para ver si es capaz de generalizar y si se podría extrapolar el uso de este tipo de redes a aplicaciones industriales. Para

ello se crea una matriz de datos de entrada con los datos de entrenamiento de cada tipo de fallo de manera separada, esto difiere de lo realizado para la creación de la red en la que todos los tipos de fallos estaban en una única matriz de datos de entrada. Se prueba la red con los distintos tipos de fallo por separado para ver su comportamiento. Las salidas para los distintos tipos de fallos se encuentran en las figuras 26, 27, 28 y 29 respectivamente.

El histograma de la figura 26, indica cuantos datos se han clasificado de manera correcta. Se observa que la mayor parte de los datos valen 1 o entorno a este valor, por lo que la clasificación mediante redes probabilísticas para datos de fallo 1 parece buena. Hay que comentar que hay unos valores que se saldrían un poco del margen de aceptación de buena clasificación dado que hay datos que llegan a valer casi 2, por lo que hay algunos valores que no se clasifican de manera óptima.

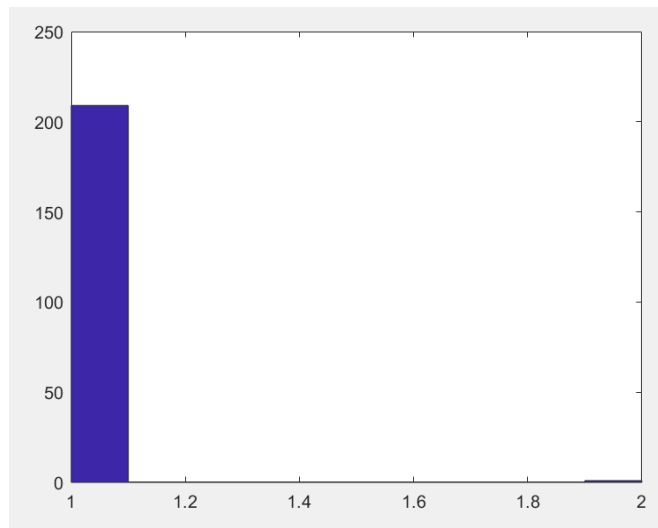


Figura 26. Histograma de clasificación de la RP con datos de test con fallo 1.

Como se puede observar en la figura 27 la clasificación de los datos de test de fallo 2 no es muy buena dado que aproximadamente la mitad de los datos aparecen como fallo 1 en vez de como fallo 2. También hay algunos valores, muchos menores, que aparecen como fallo 4. Se esperará a probar la red probabilística con datos no conocidos para poder llegar a una conclusión sobre su utilización como clasificador.

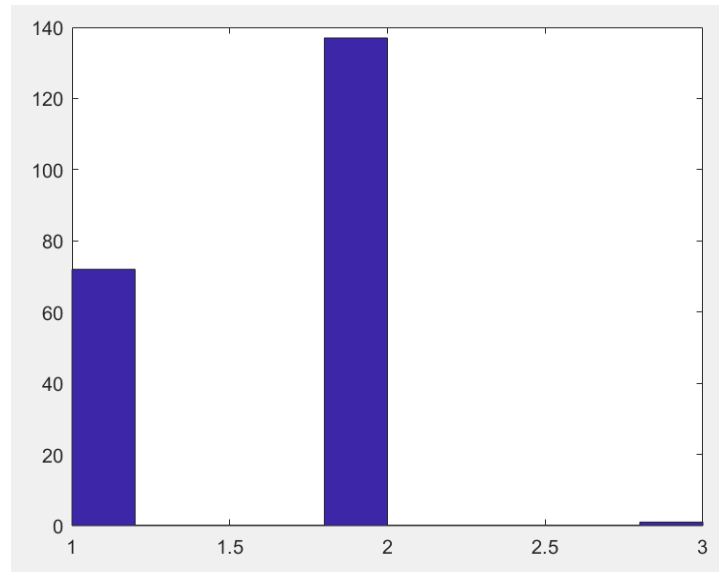


Figura 27. Histograma de clasificación de la RP con datos de test de fallo 2.

En la figura 28 se observa que la clasificación de los datos de test de fallo 4 es buena, aunque aparezcan algunos datos de los otros tipos de fallo. Cerca del 70% de los fallos, de manera aproximada, se clasifican bien, por lo que se consideraría una buena clasificación, siendo mejor que la clasificación para los datos de test de fallo 2 pero peor que la que se da con los datos de test de fallo 1.

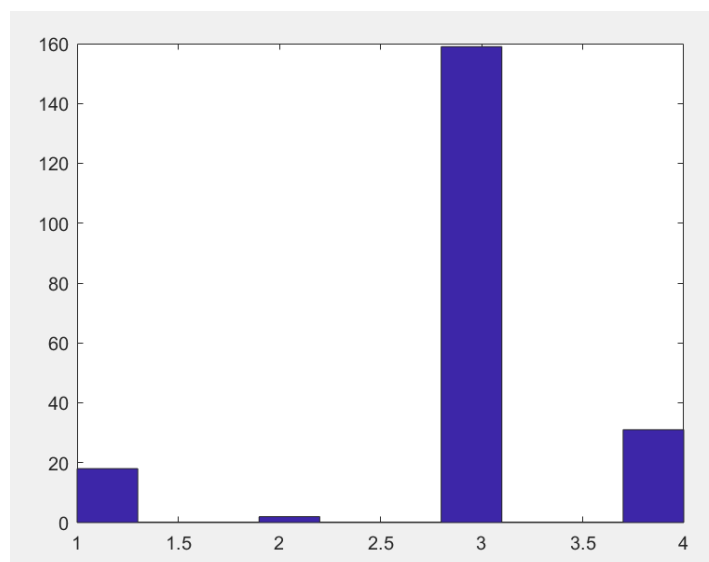


Figura 28. Histograma de clasificación de la RP con datos de test de fallo 4.

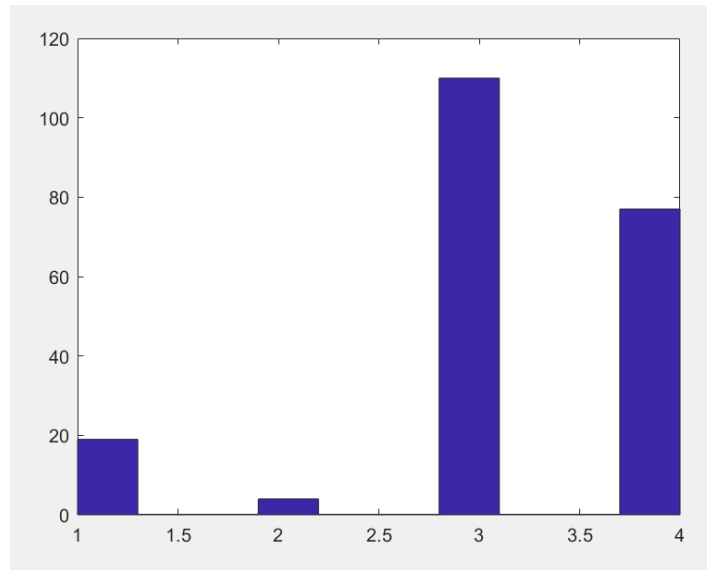


Figura 29. Histograma de clasificación de la RP con datos de test de fallo 5.

Finalmente, en la figura 29 se observa que la clasificación de los datos de test de fallo 5 que no es buena, ya que la mayoría de los datos los clasifica como fallo 4.

Para llegar a una conclusión definitiva sobre la clasificación de datos de fallo usando las redes probabilísticas se realiza una simulación con datos no conocidos, cogidos de la planta real para ver si somos capaces de clasificar los fallos que ocurren en la planta de forma adecuada. Se crean cuatro matrices de datos de entrada para los cuatro ficheros de datos no conocidos y se observa su salida en las figuras 30, 31, 32 y 33 respectivamente.

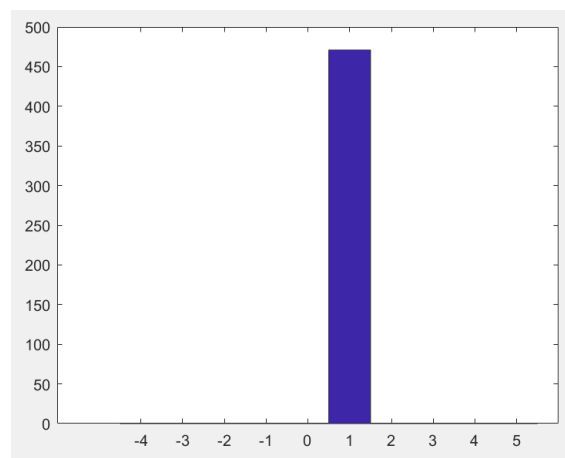


Figura 30. Histograma de clasificación de la RP con la matriz datos no conocidos 1.

Se puede observar en la Figura 30 que la clasificación de estos datos es perfecta. Los datos de fallo serán de tipo 1, dado que todos los datos aparecen concentrados en el valor 1.

Se observa en la figura 31 que los datos no se clasifican bien, dado que aparecen datos de fallo de todos los tipos, aunque predominando los datos de fallo de tipo 1 y de tipo 2. Se cree que los datos de fallo que almacena el segundo fichero son datos de fallo de tipo 2, dado que si fueran de fallo tipo 1 se clasificarían bien, como se ha descrito en este mismo capítulo del trabajo. La posible implementación de un clasificador de este tipo en la industria tendría que estudiarse a fondo, dado que dependiendo de las características del proceso y de los requisitos de calidad se podría usar o no.

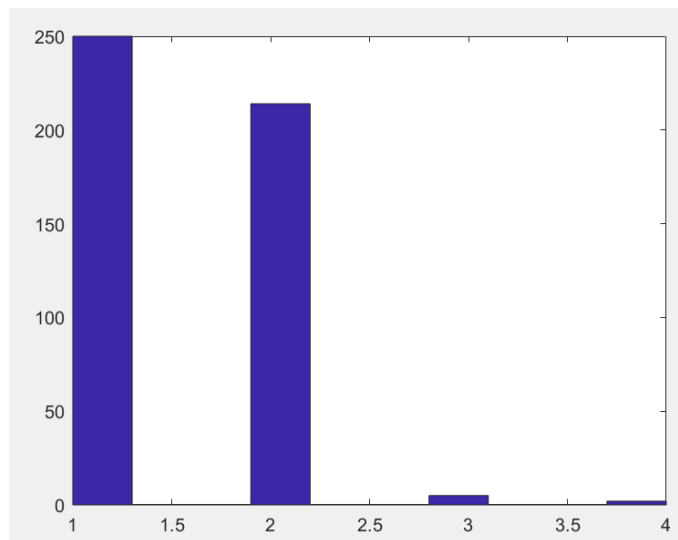


Figura 31. Histograma de clasificación de la RP con una matriz de datos no conocidos 2.

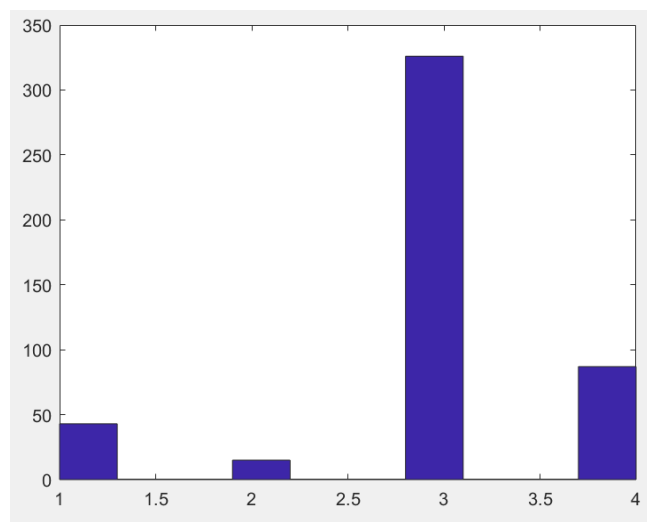


Figura 32. Histograma de clasificación de la RP con la matriz de datos no conocidos 3.

En la figura 32, se observa que los datos que contiene el fichero son datos de fallo 4. Aunque la red probabilística no clasifica los datos tan bien como los de fallo 1, se podría considerar una buena clasificación. La mayoría de los datos tienen el valor 3, aunque también aparecen bastantes datos de valor 4, es decir de fallo tipo 5, son muchos menos de los que aparecerían al utilizar la red para clasificar el segundo fichero por lo que es una mejora.

En esta figura 33, se puede observar que la clasificación es mala. Nos aparecen datos de los cuatro tipos de fallo. Dentro de estos, la cantidad de datos de fallo tipo 1, fallo tipo 4 y fallo tipo 5 es bastante grande y muy similar entre ella. Aparecen entre 150 y 200 datos de cada tipo de fallo, por lo tanto la red probabilística no nos serviría para clasificar este tipo de datos.

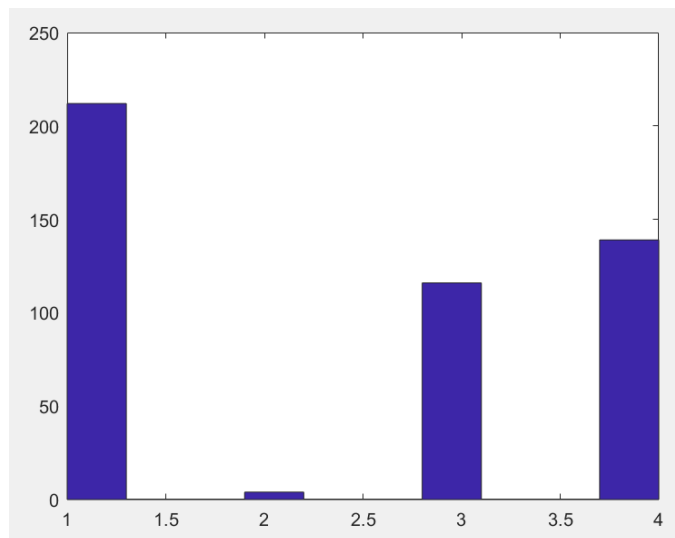


Figura 33. Histograma de clasificación de la RP con la matriz de datos no conocidos 4.

Se calculan los porcentajes de clasificación de manera analítica, los cuales se encuentran en la tabla 2.

Tabla 2. Clasificación datos no conocidos Red Probabilística.

	Porcentaje F1	Porcentaje F2	Porcentaje F4	Porcentaje F5
Matriz 1	100	0	0	0
Matriz 2	53,0786	45,4352	1,06161	0,4246
Matriz 3	9,1295	3,1847	69,2144	18,4713
Matriz 4	45,0106	0,8493	24,6285	29,5117

Observando los porcentajes de la tabla 2, nos reafirmamos en lo expuesto anteriormente. La red probabilística será válida para clasificar los datos de fallo de tipo 1. Para los datos de fallo de tipo 4 y de tipo 2, podría considerarse apta dentro de un rango, porque la red ofrece un porcentaje de

clasificación alto y en ninguno de los casos se acerca al 100% de clasificación. Como se ha expresado antes el uso de este clasificador para fallos de este tipo dependerá de la actividad o de los requisitos de la empresa. Se descarta el uso de redes probabilistas para la clasificación de fallos de tipo 5, dado que el porcentaje de clasificación es muy bajo y esto no suele convenir en las actividades industriales, porque supondría un gasto de implementación de la red que no aporta casi beneficios al proceso.

5.4. DISCRIMINANTE DE FISCHER.

El primer paso para poder realizar una clasificación de los datos mediante una simulación utilizando el discriminante de Fischer, es crear el propio discriminante, de manera análoga a como hemos creado en los dos casos anteriores las redes.

Para crear el discriminante de Fischer, hay que definir cuáles serán las entradas de datos para que posteriormente con una función matemática, nos pueda clasificar los diferentes datos. Dependiendo de la variabilidad de los datos elegida, en la creación del discriminante de Fisher, aparecen una serie de autovalores de los que dependerá la buena o mala clasificación de los datos. En este caso, al igual que en los dos anteriores, con todas las variables (52) y todos los datos de entrenamiento de cada fallo puestos unos detrás de otros, se crea una matriz de entrada que estará formada por 52 variables y (590*4) observaciones, donde las primeras 590 son de fallo 1, las siguientes 590 son de fallos 2 y así sucesivamente.

Una vez se ha creado la matriz de los datos de entrada, se calculan la matriz de dispersión de cada tipo de fallo, la matriz de dispersión dentro de las clases, la matriz de dispersión total y la matriz de dispersión entre clases. Estas matrices son necesarias para calcular los autovalores. Dependiendo de la variabilidad de los datos que se tome, se tendrá un número de autovalores, a mayor número de autovalores, mejor clasificación se dará. Una vez se determina la variabilidad, se obtienen los autovalores y autovectores y se aplica la función de Fischer que nos clasificara los datos.

Para una variabilidad del 80% elegida por el usuario se aplica el discriminante de Fischer para los datos de entrenamiento, se calculan las funciones g definidas en el capítulo 2 de esta memoria y se obtienen los resultados que se encuentran representados en las figuras 33, 35, 36 y 37. Los datos que se pueden observar en azul son los datos de fallo tipo 1, los datos en rojo los datos de fallo tipo 2, los datos de fallo tipo 4 se encuentran representados en verde y por último los datos de fallo tipo 5 se encuentran

representados en negro. El color asociado al tipo de fallo queda establecido de esta manera para toda la descripción de este método de clasificación.

En la figura 34 se puede observar como el discriminante separa los datos de fallo 1 del resto de datos, salvo al principio de la gráfica en la que los datos pueden aparecer mezclados, en el resto se ve que la clasificación es buena y que se separan de manera adecuada.

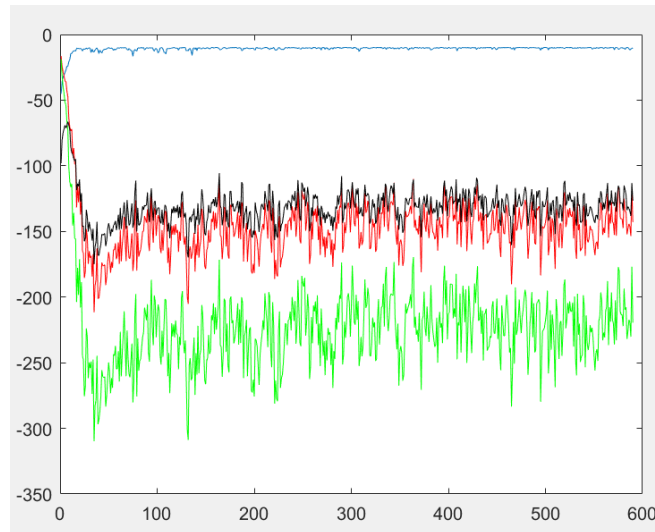


Figura 34. Gráfica FDA con los datos de entrenamiento.

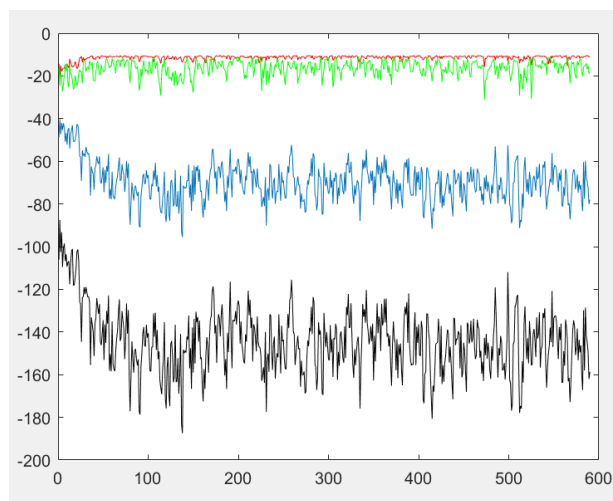


Figura 35. Gráfica FDA con datos los de entrenamiento.

Los datos de fallo tipo 2 y los datos de fallo tipo 4 no se clasifican bien. Se puede observar que los datos de fallo tipo dos quedan por encima de los de

fallo tipo 4, pero se encuentran muy mezclados y esto conllevaría a problemas. Hay que destacar que, aunque los datos de fallo 2 y de fallo 3 aparecen mezclados, al principio de la simulación todos los datos se encuentran bien clasificados y no aparecen juntos como si se podía observar en la figura 34. También se puede mencionar que, aunque la figura 35 represente la clasificación de los datos de fallo 2 respecto al resto de datos, los datos de fallo 1, representados en azul, y los datos de fallo 5, representados en negro, se encuentran separados.

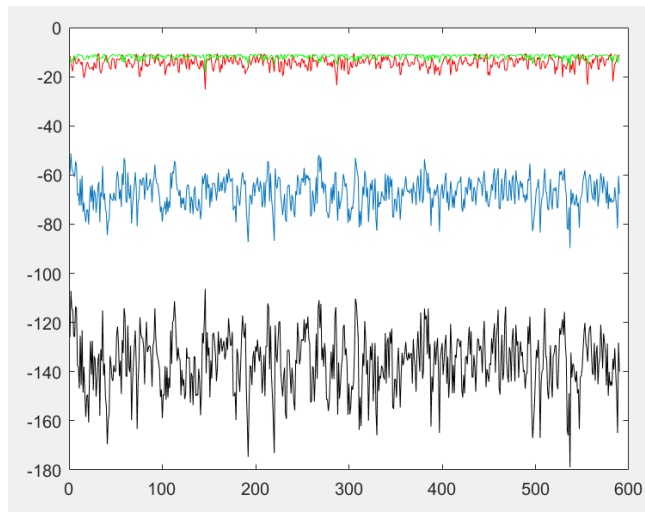


Figura 36. Gráfica FDA con los datos de entrenamiento.

En la figura 36, se observa algo muy parecido a lo visto en la figura 35. Los fallos de tipo 2 y de tipo 4 no se clasifican bien y no se separan entre ellos. En este caso los valores de fallo 4 son mayores que los de fallo 2, dado que la gráfica verde está por encima que la gráfica roja. Los datos no aparecen juntos al principio de la simulación. Las conclusiones que se han llevado para la figura 35, también pueden aplicarse a este caso, dado que lo que ocurre es muy parecido.

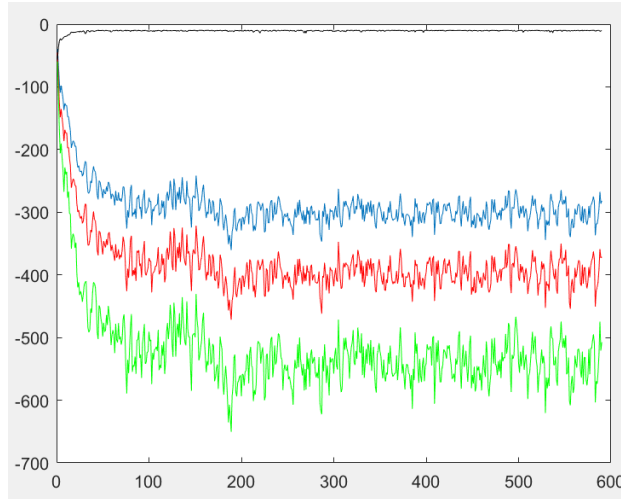


Figura 37. Gráfica FDA con los datos de entrenamiento.

Por último, en la figura 37, se puede ver como se clasifican los datos de tipo 5 del resto de fallos. Ocurre algo similar a lo visto en la figura 34. Los datos del tipo de fallo se separan bien del resto de fallos durante toda la simulación, salvo al principio donde los datos se encuentran un poco mezclados.

Según lo visto hasta ahora, el uso del discriminante de Fischer como método de clasificación de fallos sería óptimo para separar fallos de tipo 1 y fallos de tipo 5 del resto de fallos cuando la variabilidad de los datos es del 80%. Dado que esto no es de interés y no tendría aplicación industrial se realiza una nueva simulación con una variabilidad de los datos del 90% cuyos resultados se observan en la figura 38, 39, 40 y 41.

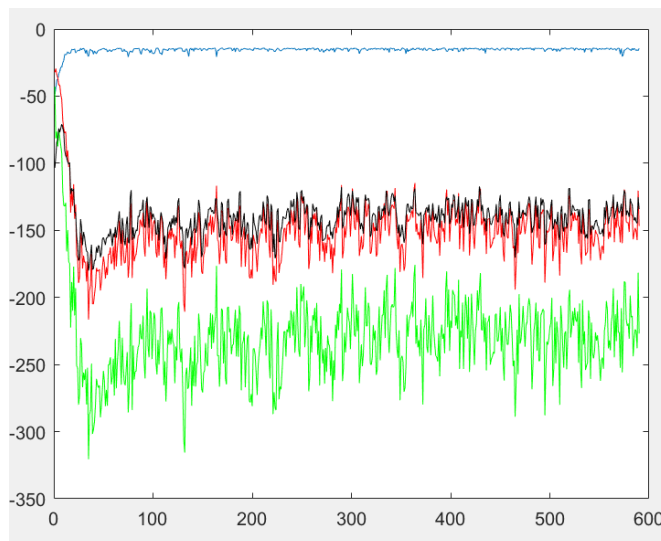


Figura 38. Clasificación FDA con los datos de entrenamiento y 90% de variabilidad.

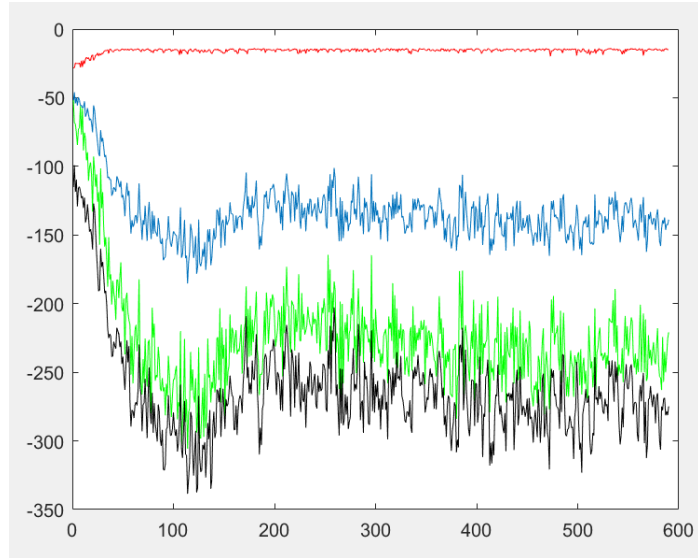


Figura 39. Clasificación FDA con los datos de entrenamiento y 90% de variabilidad.

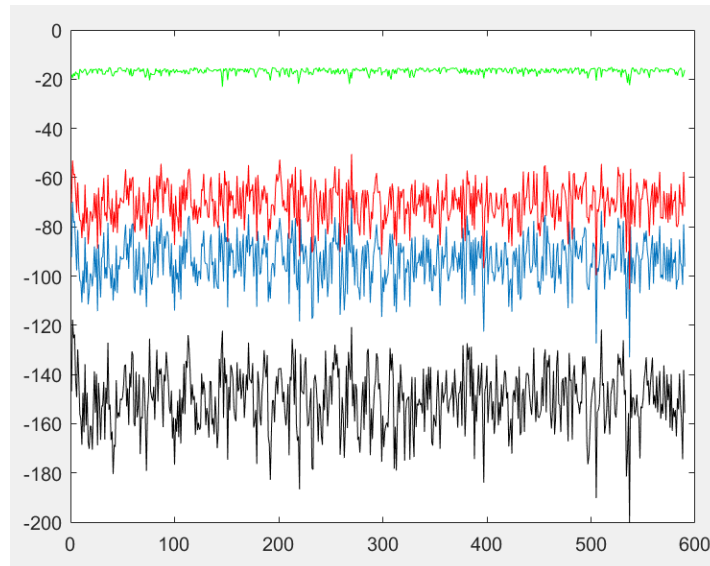


Figura 40. Clasificación FDA con los datos de entrenamiento y 90% de variabilidad.

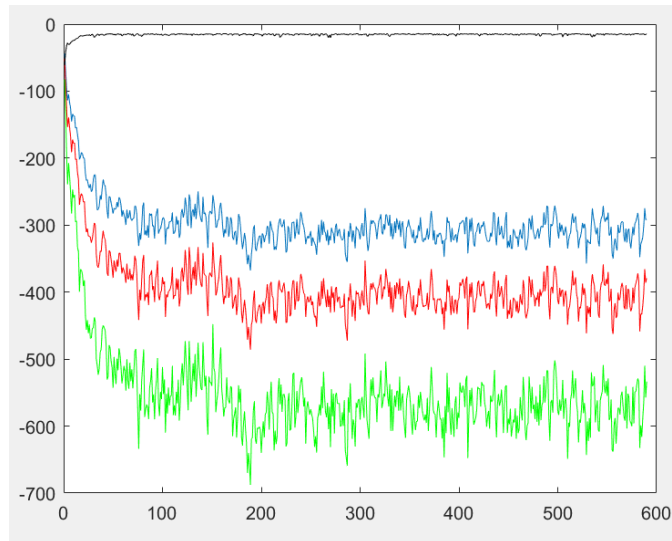


Figura41. Clasificación FDA con los datos de entrenamiento y 90% de variabilidad.

Se puede observar en estas cuatro figuras, que la clasificación que se obtiene en este caso es muy buena. En ningún punto se mezclan las gráficas y quedan muy diferenciadas en todo momento, por lo que se puede decir que con una variabilidad del 90% se podría utilizar el discriminante de Fischer como método de clasificación, lo que se va a comprobar ahora aplicando el discriminante a datos de test y a datos no conocidos.

Se crea una nueva matriz de datos de entrada por cada tipo de datos de fallo a testear, es decir 4 matrices de datos de entrada. Se aplica la función del discriminante de Fischer para cada una de las matrices de datos de testeo y se obtienen las salidas que se encuentran en las figuras 42, 43, 44 y 45 respectivamente.

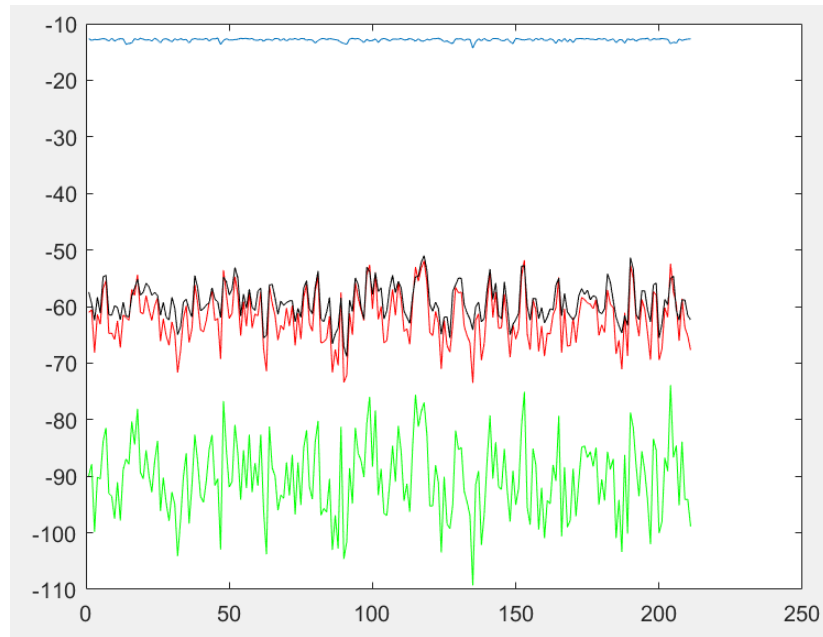


Figura 42. Clasificación FDA con datos test de fallo 1.

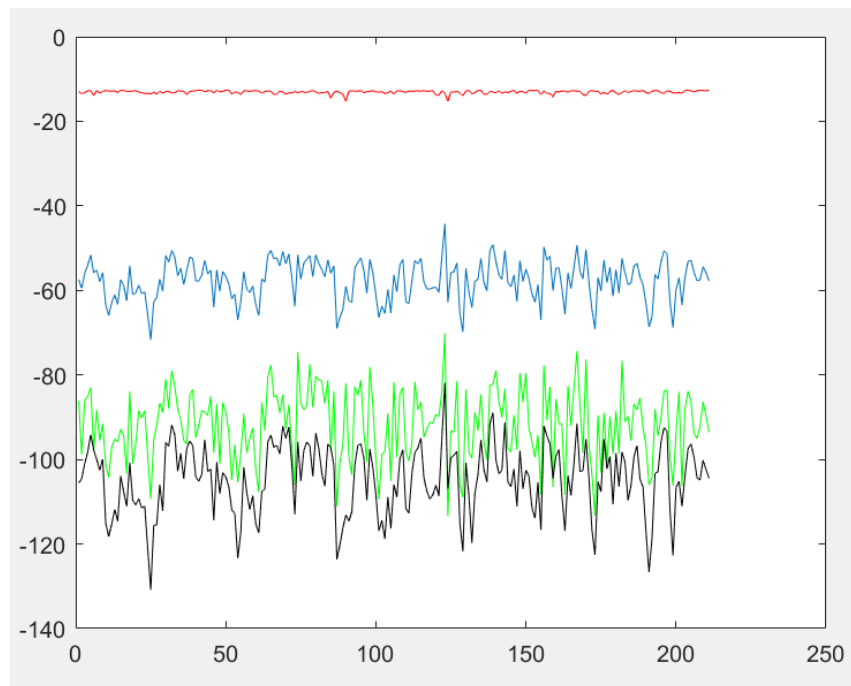


Figura 43. Clasificación FDA con datos de test de fallo 2.

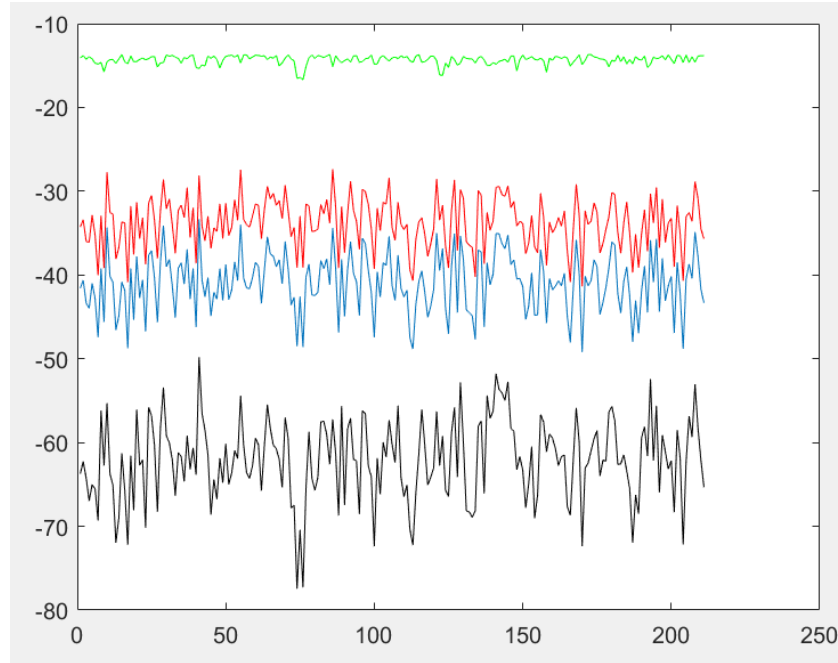


Figura 44. Clasificación FDA con datos de test de fallo 4.

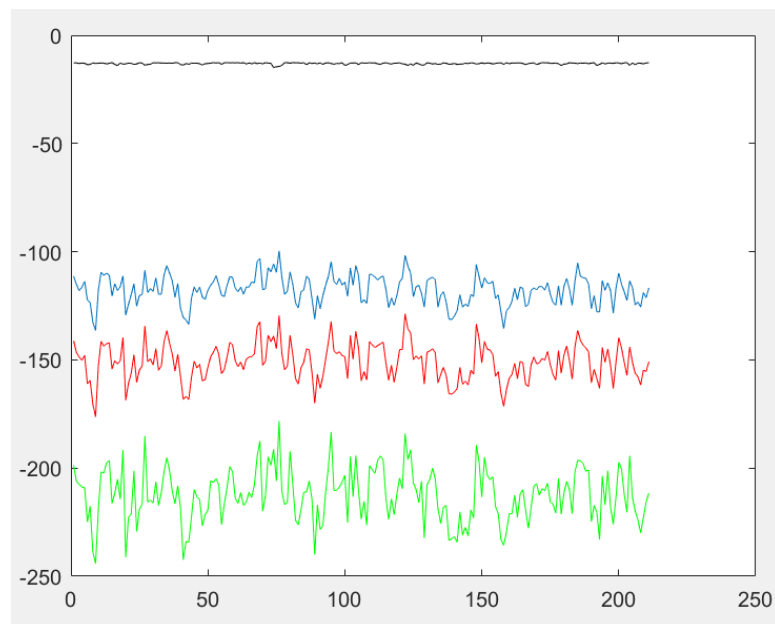


Figura 45. Clasificación FDA con datos de test de fallo 5.

Como se puede observar, la clasificación es perfecta y ocurre lo mismo que ocurría con los datos de entrenamiento. Hay mucha separación entre todos los tipos de datos y en ningún momento cabe duda del tipo de fallo que se está clasificando.

Para acabar la simulación con este tipo de clasificador se comprueba la eficacia del mismo con datos no conocidos. Para ello, se crean cuatro matrices de entrada con los datos no conocidos, y se les aplica la función del discriminante. Las gráficas de salida para la clasificación de los datos no conocidos, se encuentran en las figuras 46, 47, 48 y 49 respectivamente.

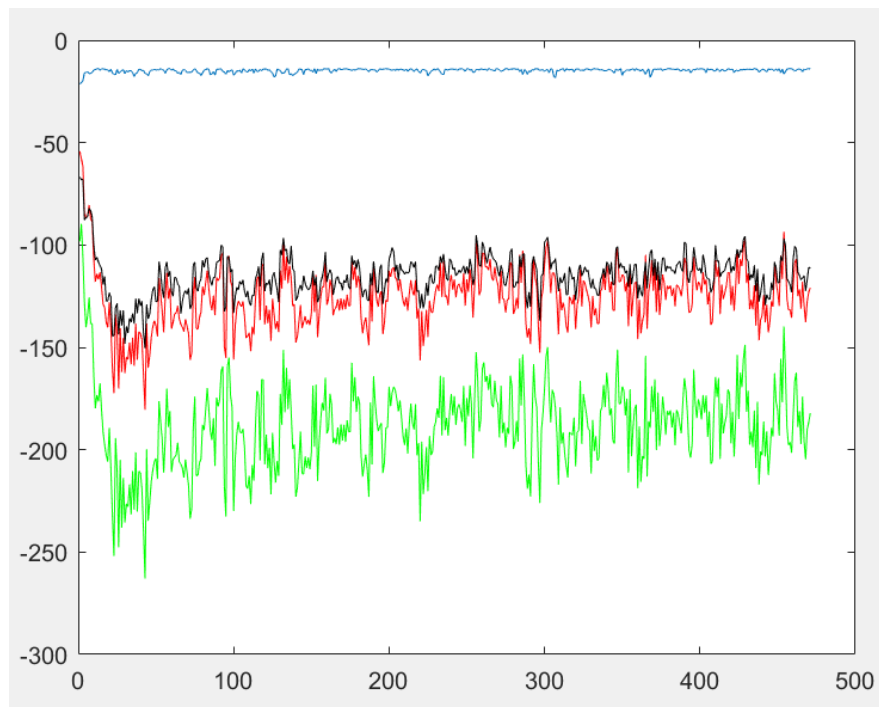


Figura 46. Clasificación FDA con la matriz 1 de datos no conocidos.

Se observa que los datos de fallo que contenía la matriz 1 de datos de entrada no conocidos, son datos de fallo tipo 1, dado que los datos representados por la gráfica azul son los que más arriba de la figura se encuentran. La clasificación es muy buena, porque en ningún momento hay oscilaciones y la separación de los datos se da en todo momento.

Los datos de fallo contenido en la matriz de datos de entrada 2, eran datos de fallo de tipo 2, como se ve en la Figura 47. Se pueden llegar a las mismas conclusiones que se llegan para la figura anterior 46, la clasificación de los datos es muy buena.

Para la figura 48 y 49 respectivamente se podría decir lo mismo que anteriormente, la clasificación es muy buena dado que los datos se encuentran separados en todo momento y prevalece una gráfica con valores máximos respecto al resto. La figura 35 nos indica que los datos de fallo de la matriz 3

son datos de fallo tipo 4 y la figura 36, nos indica que los datos de fallo de la matriz de datos de entrada 4 son datos de fallo tipo 5.

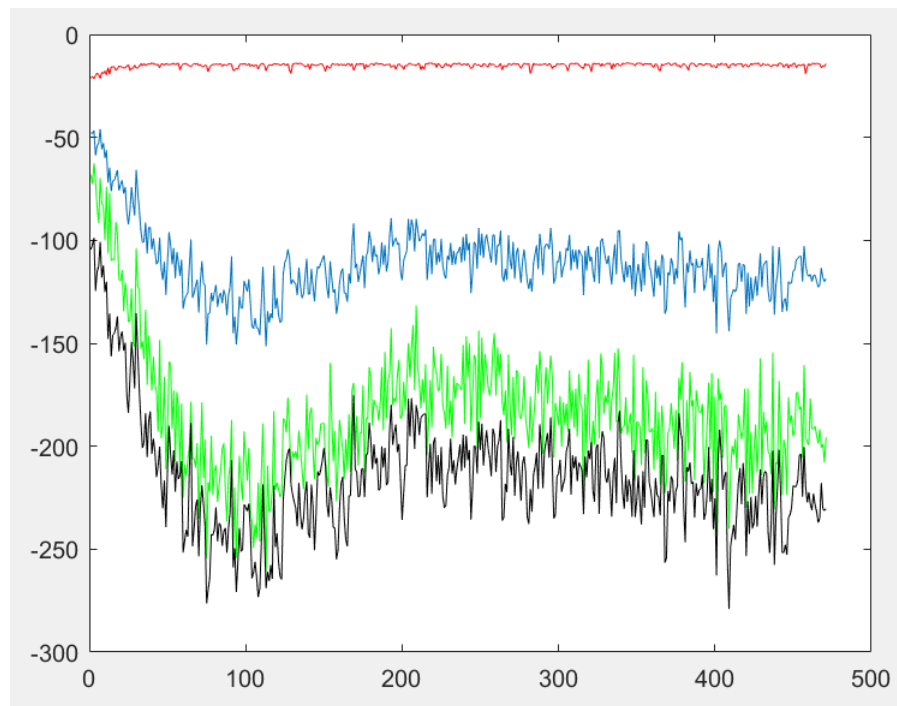


Figura 47. Clasificación FDA con la matriz 2 de datos no conocidos.

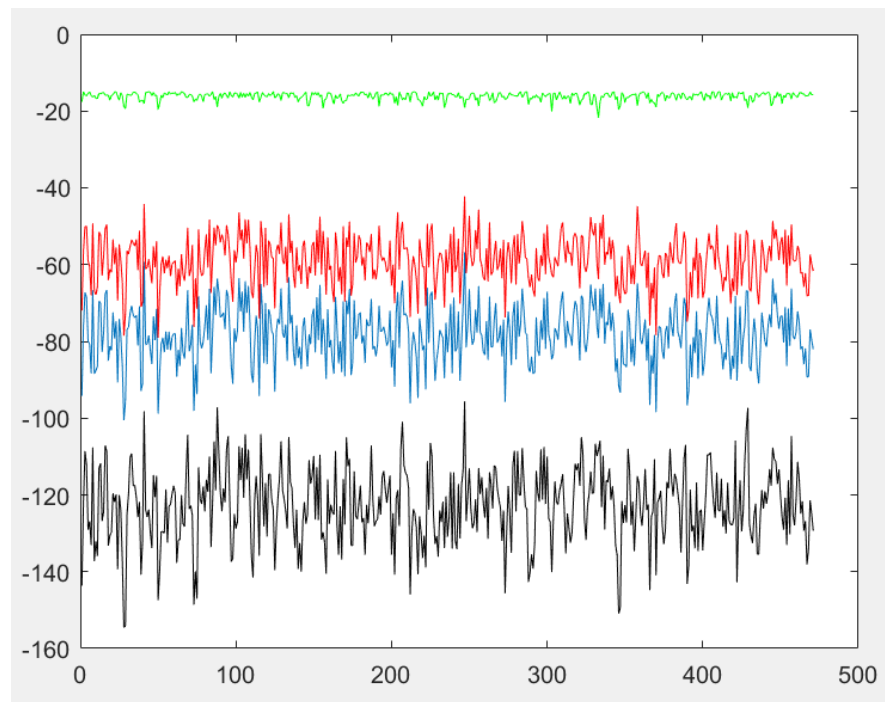


Figura 48. Clasificación FDA con la matriz 3 de datos no conocidos.

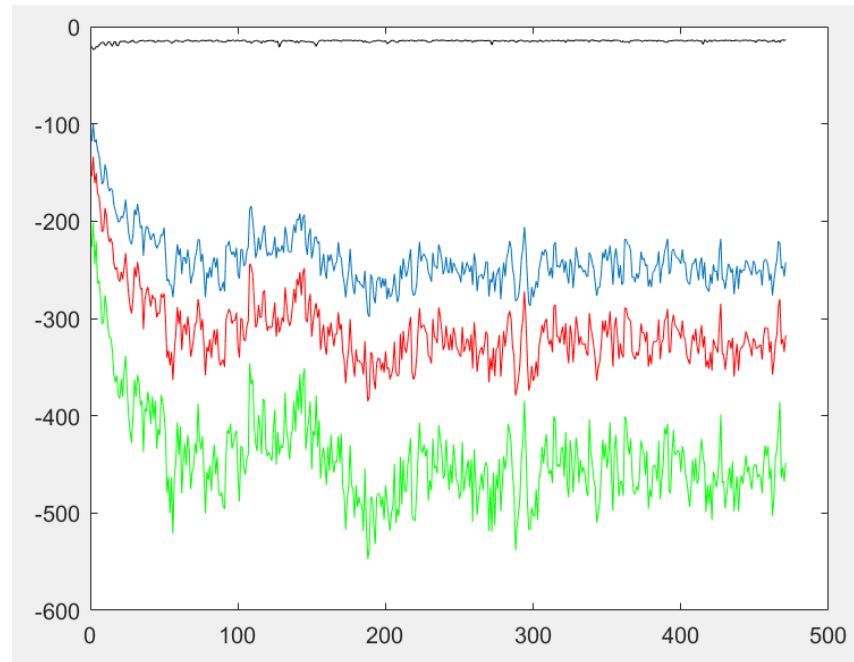


Figura 49. Clasificación FDA de la matriz 4 de datos no conocidos.

Aunque mediante las figuras se observa claramente el tipo de fallo de cada una de las matrices de datos y la calidad de la clasificación, se calcula de manera analítica los porcentajes de clasificación, cuyos datos se encuentran en la tabla 3. La clasificación de todos los datos que contienen las matrices de entrada es perfecta, dado que se clasifican todos los datos de manera adecuada, por lo que el uso para actividades industriales del discriminante de Fischer como clasificador es muy apropiado.

Tabla 3. Porcentaje Clasificación datos no conocidos con FDA.

	Porcentaje F1	Porcentaje F2	Porcentaje F4	Porcentaje F5
Matriz 1	100	0	0	0
Matriz 2	0	99,3631	0	0,6369
Matriz 3	0	0	100	0
Matriz 4	0,2123	0	0	99,7877

5.5. ARBOLES DE DECISIÓN.

Para poder llevar a cabo una clasificación mediante el uso de árboles de decisión, lo primero que se tiene que realizar, como en el resto de los clasificadores es la creación del mismo. En este trabajo, se utiliza no sólo un árbol de decisión, sino un conjunto de árboles, cuyo número debe decidir el usuario, que trabajan conjuntamente para hacer el clasificador, de ahí su nombre, bosque aleatorio o “random forest” (RF).

Como en los casos anteriores, lo primero que hay que definir son los datos de entrada. En este caso, al igual que en los anteriores, con todas las variables (52) y todos los datos de entrenamiento de cada fallo puestos unos detrás de otros. Es decir, se crea una matriz de entrada formado por 52 variables y (590*4) observaciones, donde las primeras 590 son de fallo 1, las siguientes 590 son de fallos 2 y así sucesivamente.

Para la salida de los árboles de decisión, se define un vector de dimensión de $1 \times n$, siendo n el número de filas que tiene la matriz de los datos de entrada, por lo que la salida se mostrara en una única fila. Como en las anteriores simulaciones la salida se define por el usuario y es la que va a ayudar a ver si la clasificación se realiza de manera correcta o no es así. Al crear este tipo de clasificador para clasificar 4 tipos de fallos, se asocia un valor a cada clase de fallo. Los primeros valores, hasta el 590, valdrán 1. Los valores del 590 hasta el 1180 tendrán un valor de 2. Los siguientes 590 valdrán 4 y los últimos 590 datos valdrán 5. De tal manera que los 590 primeros valores irán asociados a datos de fallo de tipo 1, los 590 siguientes datos estarán asociados a datos de fallo tipo 2, los siguientes a fallo tipo 4 y los siguientes a fallo tipo 5.

Se puede elegir el número de árboles que vaya a contener este paquete. Para la simulación de este trabajo se crea un paquete de 100 árboles. Las características que tiene el paquete de árboles creado se encuentran en la figura 50.

```

TreeBagger
Ensemble with 100 bagged decision trees:
    Training X:      [2360x52]
    Training Y:      [2360x1]
    Method:          classification
    NumPredictors:   52
    NumPredictorsToSample: 8
    MinLeafSize:     1
    InBagFraction:   1
    SampleWithReplacement: 1
    ComputeOOBPrediction: 0
    ComputeOOBPredictorImportance: 0
    Proximity:       []
    ClassNames:      '1'          '2'          '4'          '5'

Properties, Methods

```

Figura 50. Características de los árboles de decisión

Una vez se ha creado y entrenado el paquete de árboles de decisión (RF) que se utilizará como clasificador con los datos de entrenamiento, se prueba con los datos de test, para poder empezar a tomar conclusiones sobre la capacidad de clasificación. Se crea nuevamente una matriz de datos de entrada y se realiza la clasificación de estos datos, en la figura 51 se puede observar la salida del RF con los datos de test.

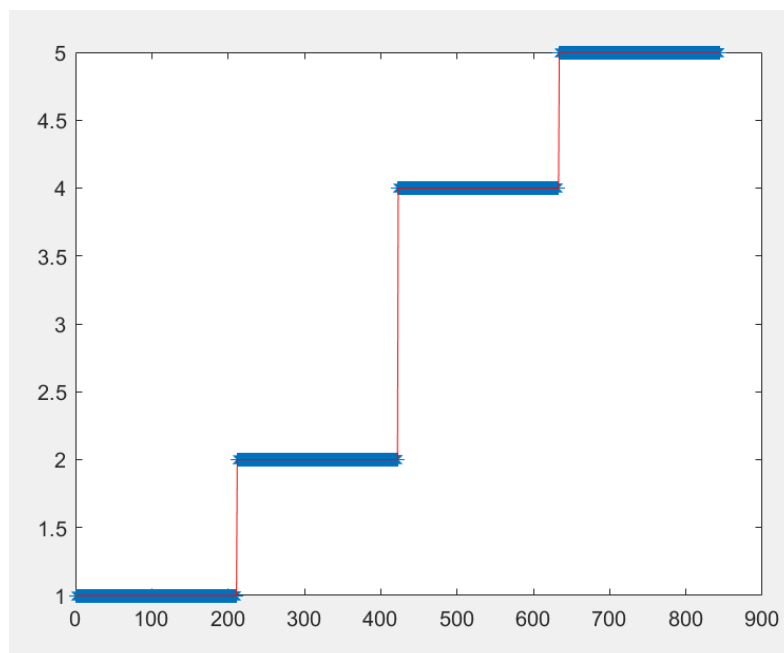


Figura 51. Salida del ensamble de árboles de decisión con los datos de test.

Se puede comprobar que la clasificación es perfecta, porque no hay ningún dato de la planta (asteriscos azules) que no se ajuste con la salida esperada (línea roja).

Se realiza una última simulación para los datos no conocidos. El modo de implementación es igual que para los datos de test, solo que los ficheros con los que se crea la matriz de entrada son los de datos no conocidos. Una

vez creada la matriz, se realiza la clasificación de los datos no conocidos la cual se puede observar en la figura 52.

Se observa en la figura 52, que la clasificación es muy buena salvo muy pocos valores. Aparecen algunos datos de fallo 5 en los datos de fallo 2 y algunos datos de fallo 1 en los datos de fallo 5. Aun con estas pequeñas desviaciones, la clasificación es muy buena.

Se calcula también el porcentaje de clasificación de manera analítica. Los porcentajes de clasificación para cada tipo de fallo se encuentran en la tabla 4.

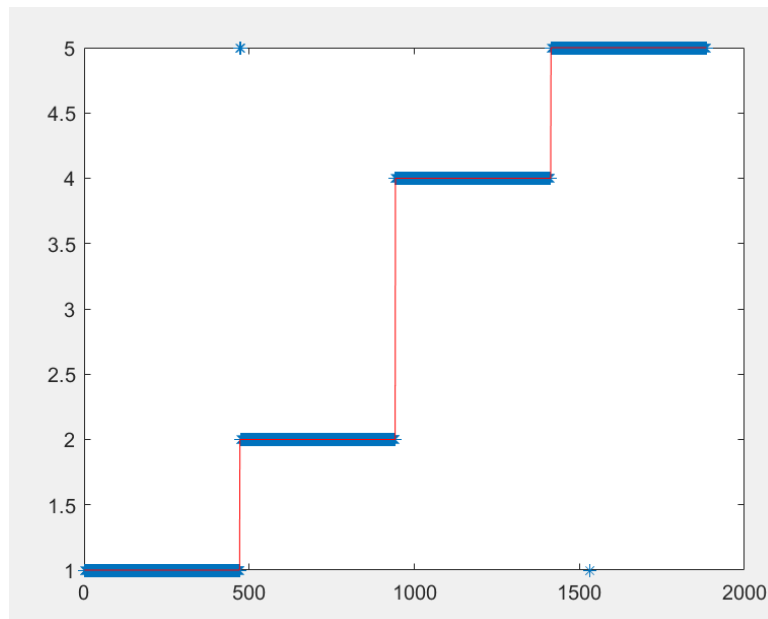


Figura 52. Salida del ensamble de árboles de decisión con datos no conocidos.

Tabla 4. Clasificación datos no conocidos árboles de decisión.

	Porcentaje F1	Porcentaje F2	Porcentaje F4	Porcentaje F5
Matriz 1	100	0	0	0
Matriz 2	0	99,3631	0	0,6369
Matriz 3	0	0	100	0
Matriz 4	0,2123	0	0	99,7877

Los porcentajes de clasificación de los fallos son extremadamente buenos, dado que en ningún caso son menores que el 99%. Se puede ver que los valores analíticos se corresponden con lo visto en la figura 39, hay algunos valores que en la clasificación del fallo 2 y del fallo 5 no corresponden por eso no se llega a tener un valor de clasificación del 100% como ocurre para el fallo 1 y el fallo 4.

5.6. MÁQUINAS DE VECTORES SOPORTE (SVM).

Como bien se ha expuesto en el capítulo dos de este trabajo la clasificación mediante máquinas de vectores soporte es un poco diferente al resto de maneras de clasificar. En los otros clasificadores valía con crear una red o un conjunto de árboles compactos, sin embargo, aquí se tendrán que crear tres clasificadores, ya que las SVM son clasificadores binarios, es decir, sólo pueden distinguir dos clases diferentes. La creación de los tres clasificadores se hará en la misma simulación con los datos de entrenamiento. Cuando se vayan a realizar los test y el análisis para los datos no conocidos también se aplicarán los tres clasificadores de manera simultánea.

El primer clasificador que se creará será capaz de distinguir los datos de fallo tipo 1 del resto de datos. El segundo clasificador se diseñará para clasificar los datos de fallo tipo 2 del resto de datos. En este caso los otros datos serán únicamente datos de fallo de tipo 4 y de tipo 5, por que los datos de fallo de tipo 1 ya se habrían descartado en el primer clasificador. El tercer y último clasificador se utilizará para clasificar los datos de fallo tipo 4 de los fallos de tipo 5.

Para crear el primer clasificador la matriz de datos de entrada contendrá todos los datos de fallo de entrenamiento. Es decir, se crea una matriz de entrada formado por 52 variables y $(590 \cdot 4)$ observaciones, donde las primeras 590 son de fallo 1, las siguientes 590 son de fallos 2 y así sucesivamente.

Para definir la salida del primer clasificador se define un vector de datos de salida que tendrá una dimensión $1 \times n$, siendo n la dimensión de las filas de la matriz de entrada. Tomará un valor de 1, para los primeros 590 valores, dado que corresponden con datos que se asocian a fallo 1. Para el resto de los valores valdrá -1, dado que no son datos de fallo 1.

De manera análoga, se crea la matriz de datos de entrada y la matriz de datos de salida para los otros dos clasificadores, lo único que cambia es que las dimensiones de ambas van cambiando. Para el segundo clasificador la matriz de datos de entrada solo contiene datos de entrenamiento de fallo 2, fallo 4 y fallo 5. El vector de salida, de dimensión $1 \times n_1$, valdrá 1, para los primeros 590 valores, dado que corresponden con datos que se asocian a fallo 2. Para el resto de los valores valdrá -1, dado que no son datos de fallo 2. Para el tercer clasificador la matriz de datos de entrada solo contiene datos de entrenamiento de fallo 4 y fallo 5. El vector de los datos de salida tiene la dimensión, $1 \times n_2$, pero en este caso los 590 primeros datos valen 1, por ser de fallo tipo 4 y los 590 datos restantes valen -1, por no ser datos de fallo tipo 4, y por lo tanto serán datos de fallo tipo 5.

En las figuras 53, 54 y 55 se puede observar la salida para los tres clasificadores con la clasificación para los datos de entrenamiento.

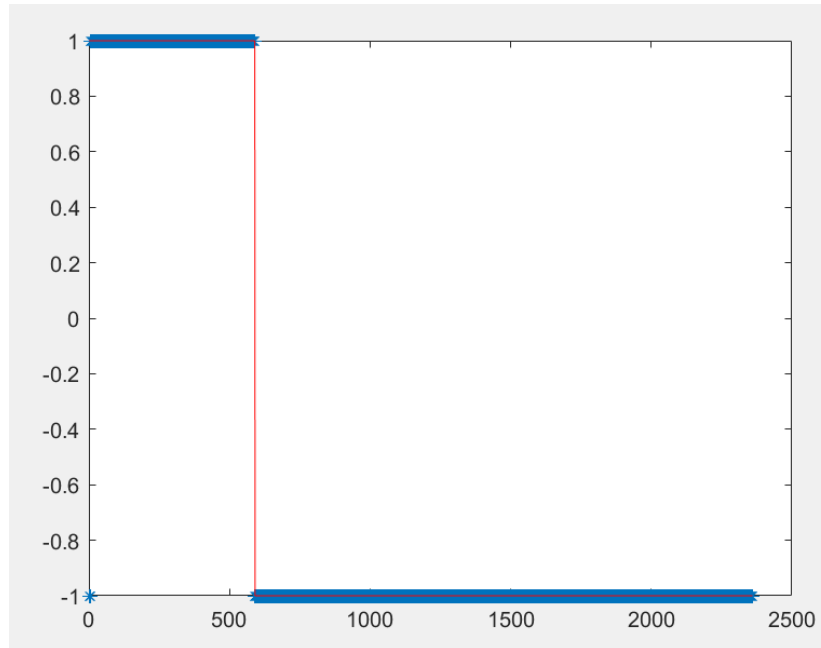


Figura 53. Salida de la SVM1 con datos entrenamiento

La línea roja representa la salida esperada. Los asteriscos azules representan los datos. Se puede ver que la clasificación es muy buena, solo hay un dato que se salga del rango. Esto es de esperar dado que los datos utilizados son los datos entrenamiento.

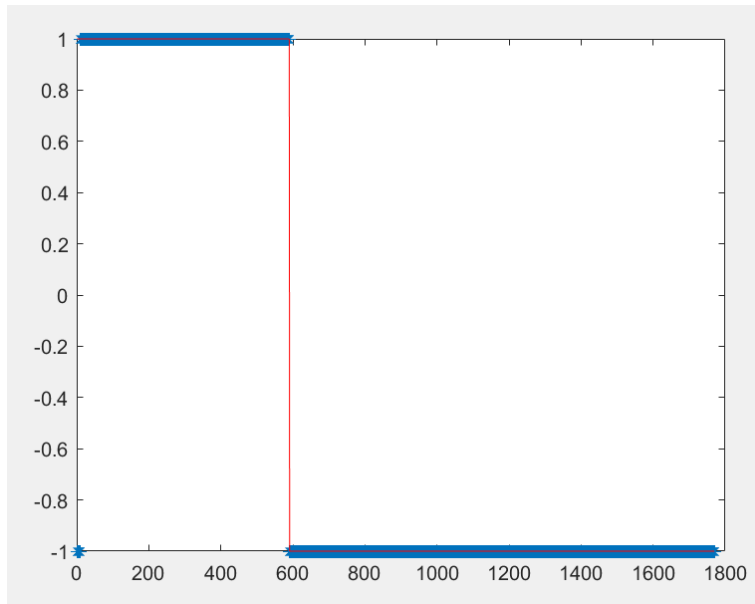


Figura 54. Salida de la SVM2 con los datos de entrenamiento

La línea roja representa la salida esperada. Los asteriscos azules representan los datos. Se puede ver que la clasificación es muy buena, solo hay un dato que se salga del rango. Esto es de esperar dado que los datos utilizados son los datos conocidos. Ocurre lo mismo que con los datos de fallo 1, algunos datos al principio no se clasifican de manera correcta.

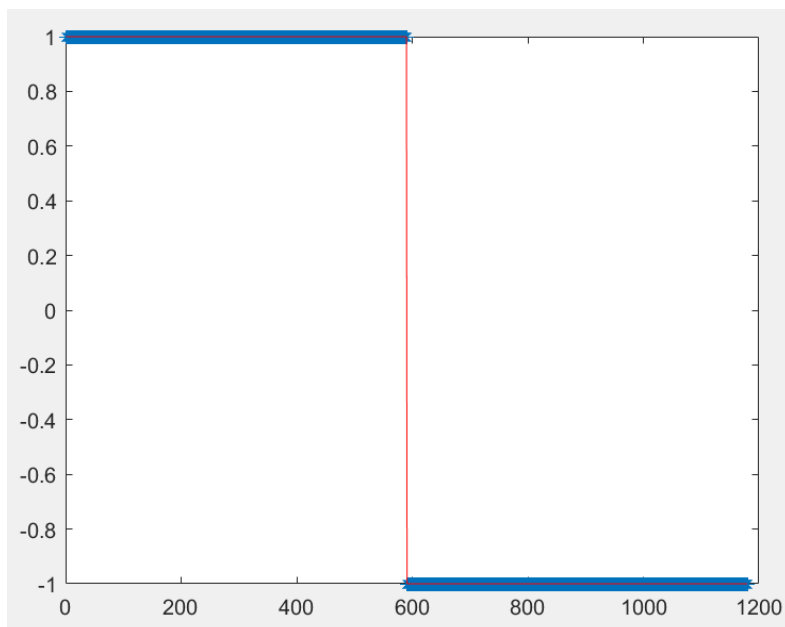


Figura 55. Gráfica salida de la SVM3 con los datos de entrenamiento.

La línea roja representa la salida esperada. Los asteriscos azules representan los datos. Se puede ver que la clasificación es mejor que las anteriores, dado que todos los datos se clasifican como corresponden. Esto puede ser debido a que tenemos menor número de datos.

Una vez se han creado los tres clasificadores, se realiza una simulación para testarlos con los datos seleccionados para esto. Se realiza el mismo procedimiento que el realizado para crear los clasificadores solo que los datos utilizados son los de prueba. Se crearán tres matrices de datos de entrada y se realiza la clasificación con las 3 SVMs previamente entrenadas y se obtiene la clasificación para los datos de test que se puede observar en las figuras 56, 57 y 58 respectivamente.

Al realizar la simulación se obtienen las siguientes gráficas de las matrices de salida (gráfica roja → salida deseada) comparadas con los datos clasificados (asteriscos azules).

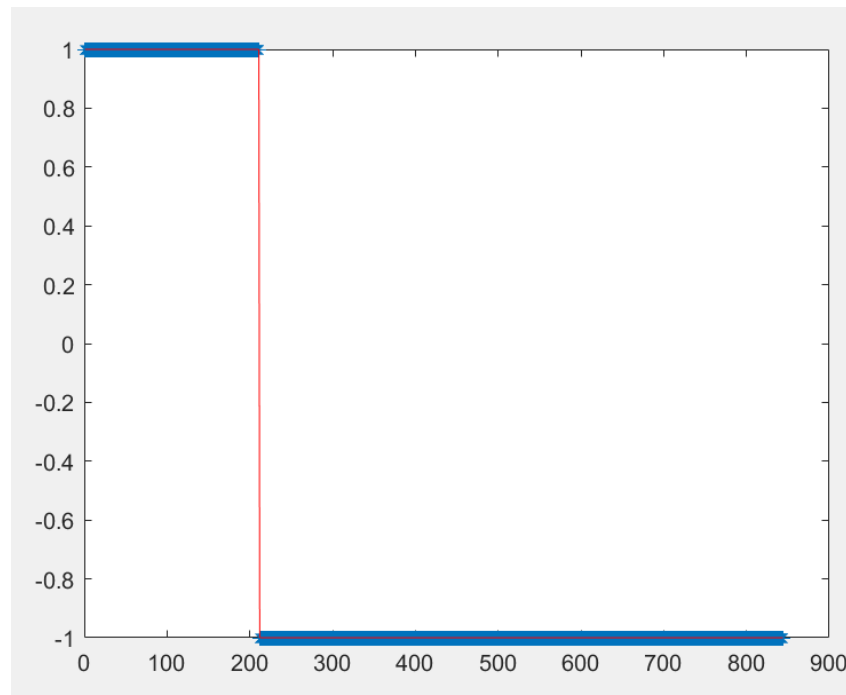


Figura 56. Salida de SVM1 con los datos de test

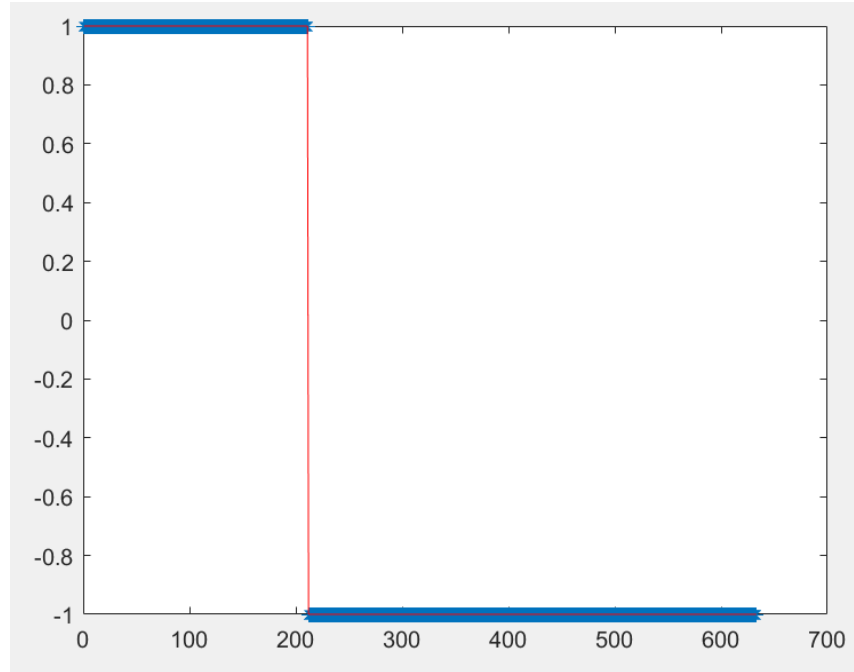


Figura 57. Salida de la SVM2 con los datos de test.

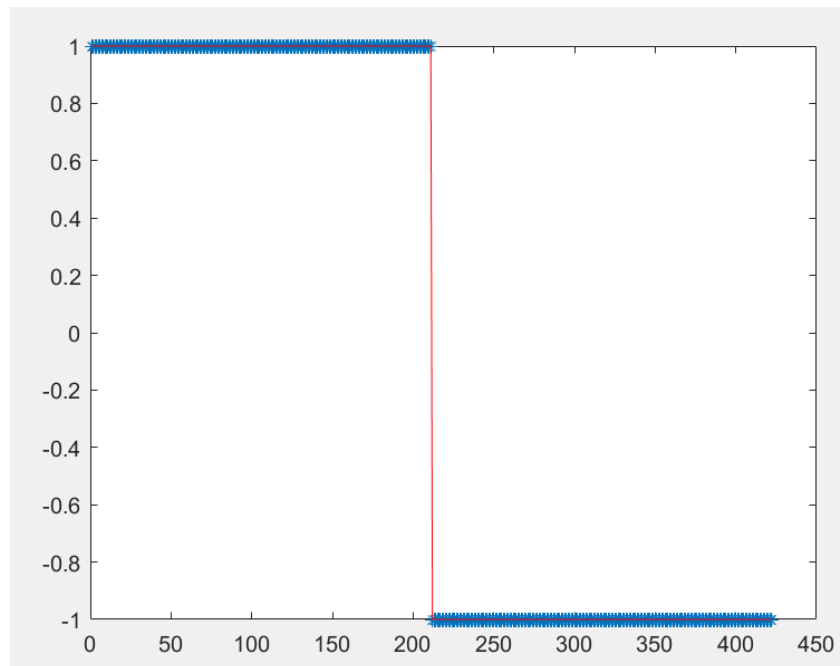


Figura 58. Salida de la SVM3 con datos de test.

Como se puede observar en las figuras 56, 57 y 58 la clasificación mediante vectores soporte para los datos de test es excelente. No hay ningún dato que se encuentre fuera del rango, por lo tanto, los tres clasificadores se han creado de manera apropiada.

Para finalizar la simulación con máquinas de vectores soporte y a su vez la simulación que se ha hecho dentro del trabajo, se van a probar los datos no conocidos con estos tres clasificadores. Esta simulación difiere de las realizadas anteriormente, dado que en la matriz de entrada habrá únicamente un tipo de fallo, no los datos de los cuatro tipos de fallo con los que se ha trabajado. La simulación nos mostrara de qué tipo de fallo se trata, el porcentaje de acierto del mismo y las gráficas de salida de estos datos.

Para la primera matriz de datos de entrada, el simulador nos indica que es fallo 1 y tiene un porcentaje de clasificación del 100%. En la figura 59, podemos observar los datos de salida. El tipo de datos que hay en el fichero son datos de fallo de tipo 1, por lo tanto, la matriz de datos de entrada solo ha pasado por el primer clasificador. El porcentaje de clasificación es el máximo y como se puede ver en la figura 47 no hay ningún dato que no tenga el valor de 1, la clasificación de datos de fallo de tipo 1 usando este método es muy apropiada.

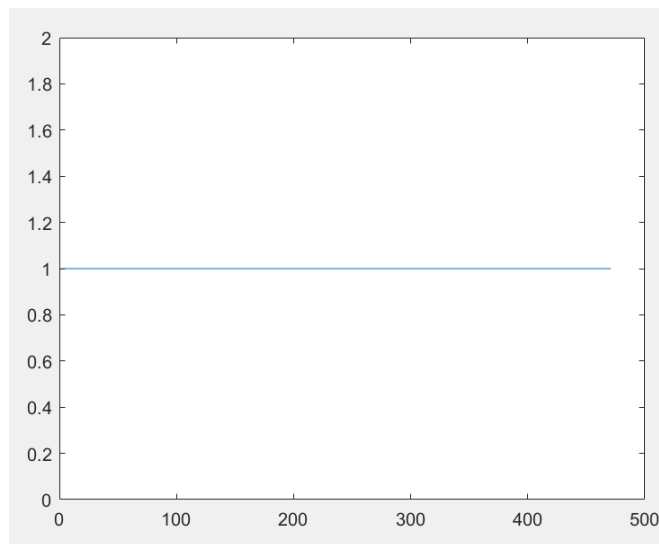


Figura 59. Gráfica de clasificación de la SVM1 con datos del fichero 1.

Para la segunda matriz de datos de entrada no conocidos, se obtiene fallo tipo 2 con un porcentaje de clasificación del 99,3631% y los datos de salida los podemos apreciar en la figura 60.

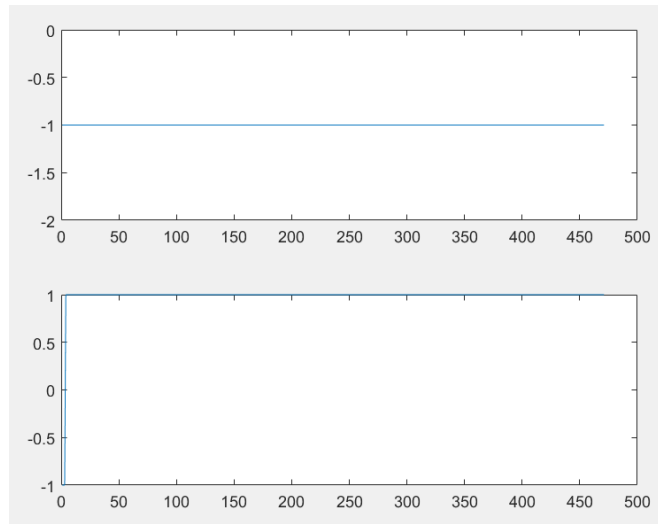


Figura 60. Gráfica clasificación de la SVM con datos del fichero 2

Los datos que se han introducido en el clasificador son datos de fallo de tipo 2. Aunque el porcentaje de clasificación de este tipo de fallo no llega al 100% tiene un porcentaje de clasificación muy elevado y que se daría por bueno en cualquier industria. En la figura 60, se ve como al pasar por el primer clasificador estos datos no son detectados, ya que son datos de fallo 2, y el resultado de este primer clasificador (SVM1) es siempre -1, es decir, no es fallo 1. Ahora pasamos el fichero de datos por el segundo clasificador (SVM2), éste los detecta y ya aparece en la gráfica el valor de 1. Se puede observar también que hay algunos valores que no los clasifica como fallo tipo 2. Esto puede ser debido a que no sean datos de este tipo de fallo o a que el clasificador no tenga una capacidad de clasificación del 100%.

Para la tercera matriz de datos de entrada no conocidos, se obtiene fallo tipo 4 con un porcentaje de clasificación del 100% y los datos de salida los podemos apreciar en la figura 61.

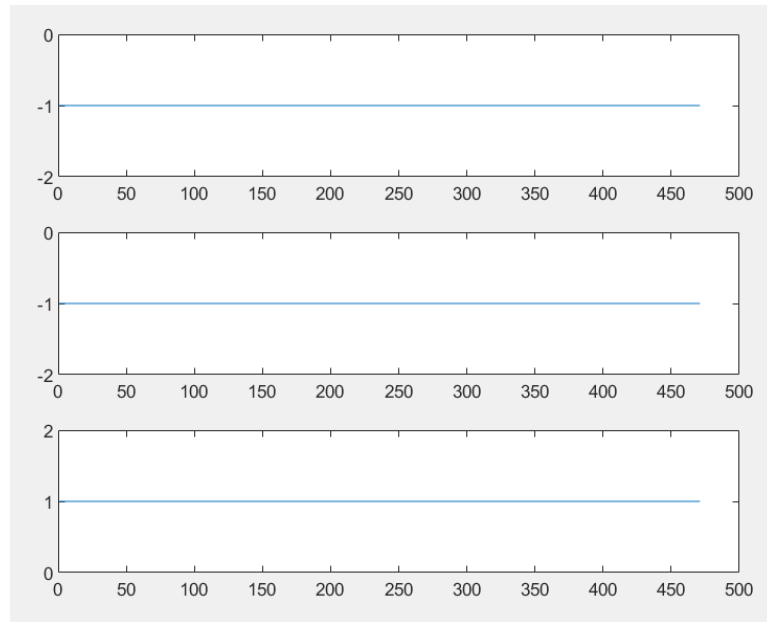


Figura 61. Gráfica de clasificación con las SVM para el fichero 3

El tipo de datos que contiene el fichero son de fallo 4. El porcentaje de clasificación es del 100%, por lo que sería una buena manera de clasificar este tipo de datos de fallo con máquinas de vectores soporte. La matriz de datos ha tenido que pasar por los tres clasificadores hasta que se ha llegado a la conclusión que los datos son de fallo 4, dado que los dos primeros clasificadores detectan fallo 1 (y por tanto para este fallo el resultado debe ser -1, es decir no es fallo 1) y fallo 2 respectivamente (con el mismo resultado, para este segundo clasificador: SVM2 el resultado debe ser siempre -1: no es fallo2). Esto se observa claramente en la figura 49 en la que las dos primeras gráficas, equivalentes a los dos primeros clasificadores, tienen una salida de -1. Sin embargo, la última gráfica tiene valor de 1, por lo que se reafirma que los datos de fallo son de tipo 4.

Para la cuarta y última matriz de datos de entrada no conocidos, se obtiene fallo tipo 5 con un porcentaje de clasificación del 100% y los datos de salida los podemos apreciar en la figura 62

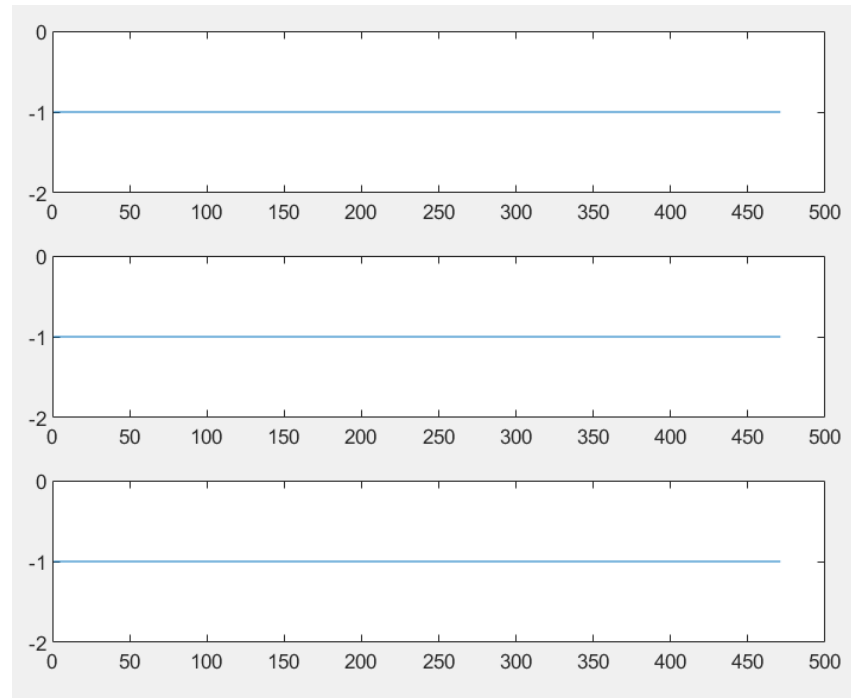


Figura 62. Gráfica de clasificación de las SVM para el fichero 5.

Los datos que conforman el último fichero son datos de fallo 5. El porcentaje de clasificación obtenido para este tipo de fallos es muy bueno y mucho mejor que el que en muchas ocasiones se aceptaría en la industria. El fichero de datos ha pasado por los tres clasificadores, obteniéndose en ellos resultado de -1, como puede observarse en la figura 50 dado que no están creados para clasificar este tipo de fallos.

5.7. COMPARACIÓN.

En este último punto del capítulo se comparan todos los métodos utilizados en la simulación. Esta comparación se realiza para la clasificación de los datos de fallo no conocidos. En la tabla 5 se puede encontrar una comparación del porcentaje de clasificación por cada tipo de fallo y método empleado.

Tabla 5. Comparación clasificadores datos no conocidos.

	Red neuronal	Red probabilística	FDA	Arboles clasificación	SVM
FALLO 1	99,7877	100	100	100	100
FALLO 2	100	45,43	100	99,3631	99,3631
FALLO 4	100	69,2144	100	100	100
FALLO 5	100	29,5117	100	99,7877	100

Como se puede observar para la detección de fallo 1 se puede utilizar cualquiera de los clasificadores estudiados, dato que el porcentaje de clasificación es muy alto para todos los casos.

La detección del fallo 2, se podría realizar con cualquiera de los clasificadores salvo con la red probabilística dado que el porcentaje que nos proporciona este método de clasificación solo es del 45%. Para casi ninguna actividad industrial u otro tipo de actividad en el que se requiera una clasificación de los datos no interesaría un porcentaje de clasificación tan bajo.

Para detectar fallos de tipo 4 cualquier tipo de clasificador, redes neuronales, FDA, arboles de decisión y SVM, serian adecuados dado que para ambos se obtiene una clasificación correcta de todos los datos. la clasificación de este tipo de fallo con redes probabilísticas también se podría dar por válido, dado que el porcentaje de clasificación es casi del 70%, dependiendo del tipo de actividad industrial o del producto que se esté fabricando se podría considerar este porcentaje como bueno.

Para el fallo 5, ocurre algo muy similar que para el fallo 2, dado que la clasificación mediante el uso de redes probabilísticas no es nada bueno. Para el resto de los clasificadores, la clasificación es muy buena dado que el porcentaje de clasificación es del 100% o se encuentra muy cerca de este valor.

Dado que la red probabilística no clasifica bien todos los tipos de fallos, se descarta su uso para la detectar y clasificar fallos. De las cuatro técnicas de clasificación restantes, se podría usar cualquiera. Por la facilidad y simplicidad de simulación, los buenos porcentajes de clasificación y el corto tiempo en el que clasifica los fallos se selecciona como mejor clasificador los árboles de clasificación, RANDOM FOREST. Aunque otros clasificadores, por ejemplo, FDA, proporcionan un porcentaje de clasificación más elevado, la complejidad de la simulación hace dejar este clasificador como una segunda opción. Cuanto más compleja sea su implementación y simulación más posibilidad de que se cometan errores en la programación de la misma.

6. CONCLUSIONES. TRABAJO FUTURO.

Una vez se han analizado las técnicas de diagnóstico de fallos y sus algoritmos de manera teórica, se han analizado los datos de la planta y se han simulado los clasificadores a estos datos se llegan a las conclusiones que se expondrán a lo largo de este capítulo.

Se mencionará el trabajo futuro que se podría realizar entorno al trabajo realizado. Se podría realizar un ensamble de todos los clasificadores diseñados, así como también analizar todos los datos del TEP con los clasificadores diseñados. También se menciona la realización de un análisis de los componentes principales para separar los tipos de fallo de comportamiento normal de los de fallo.

6.1. CONCLUSIONES.

En primer lugar, el objetivo de este trabajo era plantear técnicas de diagnóstico de fallos que pretenden detectar cualquier anomalía o malfuncionamiento en la planta. A través del desarrollo del trabajo se puede comprobar que se plantean de manera teórica las técnicas adecuadas y sus algoritmos para clasificar, lo cual se encuentra definido en el capítulo 3. Se estudian las redes neuronales perceptrón multicapa (MLP), y redes probabilísticas (PN), las máquinas de vectores soporte (SVM), los árboles de decisión y el discriminante de Fischer. Las cuatro primeras son técnicas de clasificación no-lineal, y como ya se ha dicho anteriormente, la última, el discriminante de Fischer clasifica de manera lineal.

A continuación, se requería la aplicación de estos métodos sobre un benchmark, en este caso la planta de proceso Tennessee (TEP), usada para probar técnicas de control de procesos y de monitorización y detección de fallos. Toda la información relevante a la aplicación de las técnicas se encuentra descrita en el capítulo 5.

Por último, se realizó una comparativa de todas las técnicas aplicadas para ver cuál es el mejor en este campo de la detección y diagnóstico de fallos. Todo este proceso genera las conclusiones que se describen a continuación:

- No todos los métodos de clasificación estudiados (redes neuronales, redes probabilísticas, discriminante de Fischer, árboles de decisión y vectores soporte) clasifican los cuatro tipos de fallo. Se descarta el uso de las redes neuronales probabilísticas como clasificador.
- Se selecciona como mejor clasificador los árboles de decisión, dado que los porcentajes de clasificación para todos los tipos de fallos son muy elevados y de casi el 100%. Además, su programación en el simulador es sencilla y el tiempo de clasificación es corto.

- El fallo de tipo 1, cambio en la relación de los componentes A/C, se detectada de manera óptima, con un porcentaje muy elevado de clasificación, casi el 100% de clasificación en todos los casos. Por lo tanto, si solo se quisiera detectar este tipo de fallo se podría usar cualquier tipo de clasificador. La calidad del proceso no se vería afectada por ser capaz de detectar las anomalías de este.
- El fallo de tipo 2, cambio en la composición del flujo de B, es detectado de manera óptima por todos los clasificadores salvo por la red neuronal probabilística diseñada, para la cual solo se da un 45% de clasificación. En caso de que solo se quisiera detectar este tipo de fallo, se podría usar cualquier tipo de clasificador salvo la red neuronal probabilística.
- El fallo de tipo 4, salto de la temperatura del agua de refrigeración del reactor, es detectado con una clasificación del 100% mediante el uso de red neuronal, FDA, árboles de clasificación y SVM, por lo que, si se quisiera detectar únicamente los cambios de temperatura en el agua de refrigeración, cualquiera de estos clasificadores se podría utilizar. Mediante el uso de la red neuronal probabilística, se obtiene un porcentaje de clasificación del 70%. Es difícil llegar a una conclusión del uso de este tipo de clasificador para detectar este tipo de fallo, dado que es un porcentaje de clasificación alto, aunque no clasifique todos los datos de este tipo de fallo. Desde un punto de vista conservador, se descarta su uso.
- El fallo de tipo 5, salto de la temperatura del agua de refrigeración del separador, es detectado de manera óptima por todos los clasificadores salvo por la red neuronal probabilística diseñada, para la cual solo se da un 30% de clasificación, siendo el porcentaje de clasificación más bajo que aparece en toda la parte de la simulación. En caso de que solo se quisiera detectar este tipo de fallo, se podría usar cualquier tipo de clasificador salvo la red neuronal probabilística.

6.2. TRABAJO FUTURO.

Una vez se ha llevado a cabo el estudio de todas las técnicas de clasificación y de diagnóstico de fallos propuestas para el desarrollo de este trabajo, hay que mirar hacia el futuro y ver como se podría complementar este estudio.

Se podría realizar un ensamble de todos los clasificadores diseñados, salvo la ya descartada red neuronal probabilística. Los ensambles permiten que los errores no correlacionados de clasificadores individuales puedan eliminarse por votación mayoritaria, es decir, se clasificaría el dato con los cuatro clasificadores y se tomaría como clasificación final un único dato. Para obtener este dato final, se hace una “votación” entre los cuatro clasificadores, es decir el tipo de fallo que se detecte por medio de los clasificadores será el tipo de fallo que se está detectando. [14]

Las estrategias de ensamble más utilizadas son Bagging, Boosting y Stacking. También hay varias estrategias de voto, entre las que se encuentran el voto mayoritario, el voto Bayesiano, la técnica de “ranqueo”, el voto pesado y mediante combinación lineal entre otros. [14]

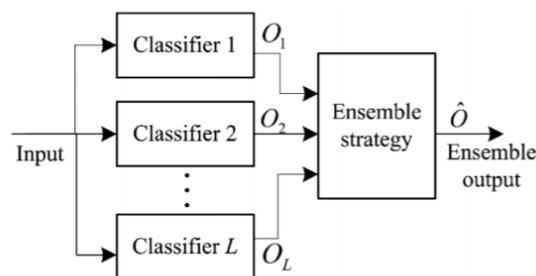


Figura 14. Esquema Ensamble. (Fuente: ccc.inaoep.mx)

Otra opción bastante recomendable de trabajo futuro sería aplicar los clasificadores diseñados al resto de tipos de fallo de la planta y ver a que conclusiones se llegarían. La programación y la implementación para esta situación, sería muy sencilla y fácil de hacer, simplemente cambiando los datos de entrada y salida del entrenamiento de cada clasificador y volver a entrenarlos. En los resultados obtenidos, lo más probable es que hubiera que descartar algún tipo más de clasificador. También al haber un mayor número de fallos, el tiempo de la clasificación cabe esperar que sea mayor, por lo que este parámetro también sería de interés para obtener el mejor clasificador.

Los datos que se han utilizado para el diseño y estudio de los clasificadores han sido únicamente datos de fallo, pero en una situación de trabajo normal los datos de fallo se mezclan los de comportamiento normal de

la planta o fabrica. El paso previo que se podría haber realizado sería la detección del fallo con el análisis de los componentes principales (PCA) y una vez detectado el fallo utilizar las técnicas estudiadas en este trabajo para diagnosticar los mismos. [15]

El análisis de los componentes principales es una técnica matemática que sirve para reducir el número de variables de forma que el mínimo número de variables esté presente en el problema a tratar. Implementando algoritmos matemáticos de PCA, se lograría extraer los datos de comportamiento normal de los datos de fallo. [15]

Otra opción, hubiera sido introducir los datos de comportamiento normal como otra clase de comportamiento, llamada por ejemplo fallo 0 y tratar de clasificar también esta nueva clase.

Lo explicado anteriormente, desde mi punto de vista, es lo prioritario y lo que más intereses pueden proporcionar para un estudio futuro, dado que también se podría llevar a cabo la implementación de otros tipos de clasificadores.

7. BIBLIOGRAFÍA.

[1] *Bakdi, Azzeddine y Kouadri, Abdelmalek.* A new adaptive PCA based thresholding scheme for fault detection in complex systems.

[2] *Rivas, R, Zubieta, P y Garcini, H.* Detección y diagnóstico automático de fallos en procesos industriales. Enero 2002.

[3] *Ngua Ngua Ayecaba, Adolfo Cursillo.* Diseño de sensores software no lineales para la estimación de las variables de calidad de los procesos, Trabajo Fin de Grado de Ingeniería Electrónica Industrial y Automática, EII, Universidad de Valladolid. 2019.

[4] *Guemes García, Enrique.* Diseño de sensores software para el control de calidad de un proceso. TFG Ingeniería en Tecnologías Industriales, EII Universidad de Valladolid. 2018.

[5] *Specht, Donald F.* Probabilistic Neural Networks. Junio 1989.

[6] *Robles, Víctor.* Tesis Doctoral: Clasificación Supervisada basada en Redes Bayesianas. Aplicación en Biología Computacional. 2003

[7] *Cadena, Juan Andrés. Cadena, Juan Mauricio. Pérez, Sandra Milena.* Aplicación de redes neuronales probabilísticas en la detección de fallas incipientes en transformadores. Septiembre 2008.

[8] *L. Puigjaner, P. Ollero, C. De Prada, L. Jiménez,* Detección y diagnóstico de fallos, capítulo 10 de: Estrategias de modelado, simulación y optimización de procesos químicos, Ed, Síntesis. 2006,

[9] *Roman, Víctor.* Aprendizaje Supervisado: Introducción a la Clasificación y Principales Algoritmos. Marzo 2019. Consulta(8/7/2019) <https://medium.com/datos-y-ciencia/aprendizaje-supervisado-introducci%C3%B3n-a-la-clasificaci%C3%B3n-y-principales-algoritmos-dadee99c9407>

[10] *Breiman, Leo.* Random Forests. Machine Learning. Mayo 2001.

[11] *Alba, José Luis.* Máquinas de Vectores Soporte (SVM). Agosto 2014.

[12] *J.J. Downs, E.F. Vogel*, A plant-wide industrial process control problem, *Compututer Chemical Engineering* 17. 1993.

[13] *L. Chiang, E. Russell, R. Braatz*, Fault Detection and Diagnosis in Industrial Systems, Springer-Verlag. 2000.

[14] *Morales, Eduardo*. Instituto Nacional de Astrofísica óptica y electrónica de Méjico. Consulta (8/7/2019) <https://ccc.inaoep.mx/~emorales/Cursos/Aprendizaje2/Acetatos/ensambles.pdf>

[15] *Autor desconocido*. Consulta (8/7/2019). <https://www.lis-solutions.es/blog/algoritmo-pca-de-lo-complejo-a-lo-sencillo/>

