Universidad de Valladolid

ESCUELA DE INGENIERÍAS INDUSTRIALES

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTIALES

MÁSTER EN INVESTIGACIÓN EN INGENIERÍA DE PROCESOS Y SISTEMAS INDUSTRIALES

Clasificación de fallos con métodos no lineales y algoritmos de agrupación basados en densidad

Autor:

Caminero Lozano, Ricardo Antonio

Tutor:

De la Fuente Aparicio, María Jesús
Sainz Palmero, Gregorio Ismael
Departamento de Ingeniería de Sistemas y Automática.

Valladolid, septiembre de 2020

Resumen

En este trabajo se pretende mejorar la calidad de los procesos industriales mediante la clasificación de fallos. Es de vital importancia tener un buen control sobre la calidad en un proceso industrial, dado que un fallo puede provocar la disminución de la calidad del proceso que puede suponer un elevado costes personales, materiales y medioambientales. Debido a la entrada de la industria 4.0 es cada vez más común la monitorización de los procesos industriales la cual acaba resultando en una gran cantidad de datos que pueden aprovecharse para mejorar la calidad del proceso. Con este estudio se analiza el uso de métodos de reducción de la dimensionalidad no lineales, concretamente *Locally Linear Embedding* (*LLE*) e *ISOMAP*, para la clasificación de fallos en plantas. Aplicando diferentes aproximaciones y realizando diferentes experimentos se intentará identificar una serie de fallos pertenecientes a un *benchmark*, muy utilizado en la literatura científica, de una estación depuradora de aguas residuales (EDAR) llamado BSM2. Además, se realiza un segundo análisis utilizando algoritmos de agrupamiento basados en densidad (DBSCAN, HDBSCAN y OPTICS), con los que se tratará de formar agrupaciones que sean capaces de clasificar los fallos de la EDAR. Por último, puntualizar que la realización de los experimentos ha sido realizada con lenguaje *Python*.

Abstract

This work aims to improve the quality of industrial processes by classifying failures. It is vitally important to have good quality control in an industrial process, since a failure can cause a decrease in the quality of the process that can lead to high personal, material and environmental costs. Due to the entry of industry 4.0, the monitoring of industrial processes is increasingly common, which ends up resulting in a large amount of data that can be used to improve the quality of the process. This study analyzes the use of non-linear dimensionality reduction methods, specifically Locally Linear Embedding (LLE) and ISOMAP, for the classification of plant failures. Applying different approaches and carrying out different experiments, an attempt will be made to identify a series of failures belonging to a benchmark, widely used in scientific literature, of a wastewater treatment plant (WWTP) called BSM2. In addition, a second analysis is performed using density-based clustering algorithms (DBSCAN, HDBSCAN and OPTICS), with which it will be tried to form clusters that are capable of classifying the failures of the WWTP. Finally, point out that the experiments have been carried out with the Python language.

Desearía agradecer a María Jesús de la Fuente Aparicio y a Gregorio Sainz Palmero su inestimable ayuda e interés, sin los cuales hubiera sido muy difícil la realización de este trabajo. También por trasmitirme sus conocimientos, al igual que los demás profesores que me han acompañado durante este curso y a la Universidad de Valladolid por hacerlo posible. Además, dar las gracias por haber podido participar en este proyecto, con el cual he mejorado mucho mis habilidades como investigador.

Por último, quisiera agradecer a mi familia, a mi madre y a mi padre, a mi hermano y a todos los que desinteresadamente me han apoyado, no sólo durante este último año, sino durante todos los años en los que han estado ahí para mí. También a Aleix, Oriol y Sergi que desde hace más de 18 años llevan animándome y acompañándome fielmente en los momentos buenos y los malos. Por último y especialmente a Bruna, que con paciencia, cariño y amor me ha ayudado y aconsejado durante este proyecto y desde que entró en mi vida.

Índice

Índice de figuras

Índice de tablas

Capítulo 1.-Introducción y objetivos

1.1 Introducción

De forma general se podría decir que todo tipo de sistema industrial puede verse afectado por un fallo, que en los casos más leves ese mismo fallo no supondría ningún problema y en los casos más graves puede suponer un coste elevado tanto personal, medioambiental como económico. En la actualidad, es cada vez más común que cualquier empresa y proceso industrial esté ampliamente monitorizado, tendencia que se acrecienta por la entrada en el panorama de la industria 4.0 [1]. Además, esta cantidad de datos que diariamente se recopila tiene un gran potencial para mejorar la eficiencia y la eficacia de la planta, sin embargo, no es suficiente con representar la variable medida en una interfaz vigilada por un operario. Por lo tanto, si se tratan los datos de forma correcta se puede optimizar, mejorar y robustecer el proceso en cuestión [2].

La detección y diagnóstico de fallos cobra cada vez un papel más relevante en la industria. Se entiende como fallo una desviación no permitida de al menos una propiedad característica o parámetro del sistema de la condición aceptable / habitual / estándar [3]. Existen diferentes métodos y aproximaciones para implementar un sistema que sea capaz de realizar esta función que deben tener ciertas cosas en común, la rapidez detectando cuando ocurre un fallo y la capacidad de acotar donde ha ocurrido el fallo dentro del proceso. Los métodos también deben ser capaces de funcionar en tiempo real de poco sirve un método de detección, que para clasificar un fallo correctamente, necesite de gran número de ciclos de producción porque tardaría demasiado en subsanarlo. Hay que tener en cuenta que el límite de tiempo para detectar un fallo depende del sistema en concreto. Existen métodos que son capaces de detectar rápidamente el fallo, no obstante, no consiguen una clasificación precisa sobre que ha fallado. Entre los diferentes métodos encontramos técnicas basadas en el conocimiento del funcionamiento del proceso, técnicas basadas en modelos teóricos [4], y técnicas basadas en el análisis de datos [5].

Es evidente que según las intenciones y los resultados que se esperen realizando el análisis, el proceso puede tornarse una tarea compleja. Analizar una o varias variables sin tener en cuenta que un proceso no depende únicamente de una variable y que, de forma general, las variables están relacionadas entre sí, no resultará en un análisis correcto y preciso de la situación del sistema. En el caso concreto en el que una variable diese una lectura alertando de un fallo, este fallo no tendría por qué haber ocurrido en esta parte del proceso, podría haber ocurrido antes o después de entrar en juego esta variable. Es importante entonces, no solo que el sistema pueda detectar cuando ha ocurrido un fallo, sino que además el sistema de monitorización tuviera la capacidad de indicar donde se ha producido, de esta manera no sería trabajo del operario estar monitorizando constantemente diversas variables para detectarlo. Por lo tanto, se

facilitaría y optimizaría la puesta en marcha de nuevo del proceso una vez subsanado dicho fallo.

En cuanto a este Trabajo Fin de Máster (TFM) se trata de implementar métodos de detección y diagnóstico de fallos en una planta industrial, usando métodos basados en datos. Además, como las plantas industriales tienen muchas variables interconectadas, es decir, son de alta dimensionalidad, se experimentará con métodos de reducción de la dimensionalidad no lineales y algoritmos de agrupamiento basados en densidad, con los que se tratará de crear un método que permita la clasificación correcta de los fallos. Dado que necesitamos estudiar múltiples variables nos centraremos en los métodos de detección de fallos basados en las técnicas estadísticas multivariantes [6], los cuales están pensados para analizar muchas variables, en este tipo de métodos se busca eliminar aquellas variables menos representativas para simplificar el modelo, es decir, reducimos la dimensionalidad de los datos, para representar el comportamiento del sistema en dos o tres dimensiones, en vez de tener tantas dimensiones como variables medidas. Al aplicar estas técnicas a la detección de fallos obtenemos diferentes métodos de detección como Análisis de Componentes Principales (PCA), Análisis del Discriminante de Fisher (FDA), Mínimos Cuadrados Parciales (PLS), etc.

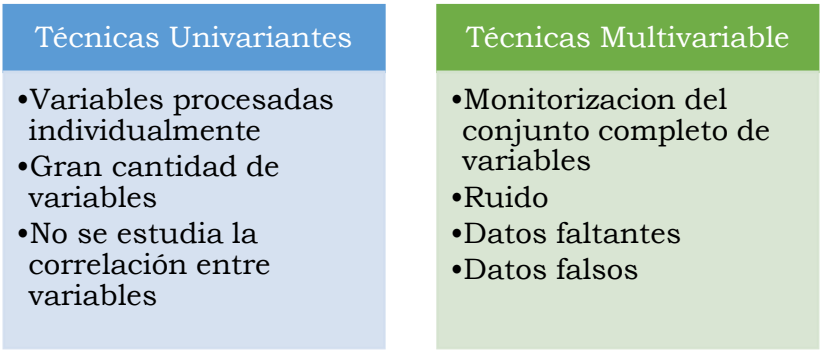


Figura 1 Comparación Univariante-Multivariante.

Existe otra diferenciación entre los métodos mencionados anteriormente como el PCA o el PLS los cuales son métodos lineales, es decir, buscan una relación lineal entre sus variables. Pero en los sistemas industriales, aunque se pueda encontrar cierta relación, lo normal es que esta sea no lineal, por lo que los métodos lineales no acaban de conseguir una buena localización del fallo. Por este motivo, se ha decidido buscar métodos de análisis no lineales e intentar aplicarlos al campo de la detección y diagnóstico de fallos. Los métodos no lineales que estudiaremos serán el LLE (*Locally linear Embedding*) y el método ISOMAP.

1.2 Objetivos

El objetivo principal de este TFM es desarrollar métodos de detección y aislamiento de fallos (FDI), y que estos sean aplicables al mayor número de plantas e industrias posibles. Las técnicas FDI han evolucionado en el tiempo junto con la industria y en la actualidad, con la llegada de la industria 4.0, es posible y necesario analizar un gran número de datos (*Big Data*), es aquí donde entran los métodos que se estudian en este trabajo, de los cuales se intenta aprovechar su no linealidad para superar a los métodos lineales (bien probados y robustos).

En los métodos de detección de fallos generalmente hay dos pasos básicos para detectar y diagnosticar un fallo. El primer paso es detectar que ha ocurrido un fallo, el segundo paso es intentar diagnosticarlo, este último paso es el punto débil de los métodos lineales. Puesto que lo más complicado en la actualidad es llevar a cabo la clasificación del fallo y no tanto su detección, es en esta última parte en la que se centrará este estudio, usando métodos no lineales.

Por lo tanto, se busca crear un método de identificación de fallos con métodos no lineales, que sea capaz de clasificar correctamente diferentes fallos, de una Estación Depuradora de Aguas Residuales (EDAR). Para ello se utilizarán los métodos ISOMAP y LLE para la reducción de la dimensionalidad y algoritmos de *clustering* basados en densidad DBSACAN, HDBSCAN y OPTICS para clasificación. Así pues, tratando de realizar diferentes pruebas y combinaciones para lograr un buen método.

1.3 Organización de la memoria

La memoria de este trabajo se organiza en seis capítulos.

En el primer capítulo, se introduce el tema tratado, la detección y diagnóstico de fallos, dónde se da una visión general y se ayuda a comprender la necesidad de desarrollar nuevos métodos como los que se tratan en este trabajo. Se definen, además, los objetivos del mismo.

En el segundo capítulo, se realiza un estudio teórico sobre los métodos y algoritmos utilizados en los experimentos. Se divide en dos grupos los métodos *manifold* para la reducción de la dimensionalidad y los algoritmos de *clustering*. En un último apartado se explica brevemente el *software* que se ha utilizado para llevar a cabo los experimentos.

En el tercer capítulo se explica cómo funcionan las EDAR y se introduce un *benchmark* de una planta EDAR, muy utilizado en la literatura científica, llamado BSM2, desarrollado por la International Water Association (IWA) [7], de la cual se han extraído los datos utilizados en el experimento. Se muestra la manera en la que están organizados los datos y que series de fallos se van a tratar.

En el cuarto capítulo se expone la experimentación desarrollada para este trabajo, se definen dos grupos de experimentos, el primero donde se aplican los métodos *manifold* y el segundo donde se combinan estos métodos con los algoritmos de *clustering*. En cada grupo se definen los experimentos a realizar y se muestran los resultados obtenidos con conclusiones breves.

En el quinto y último capítulo se exponen las conclusiones obtenidas tras estudiar y analizar los experimentos realizados. Para finalizar se sugieren futuras líneas de investigación y experimentación.

Capítulo 2.- Estudio Teórico

2.1 Introducción

En los sistemas industriales es necesario aplicar medidas de seguridad y controles de calidad tanto sobre los productos como en los propios procesos industriales. Asimismo, por la necesidad de cumplir las leyes y regulaciones que los rigen y teniendo en cuenta que una planta funcionando mal puede provocar tanto accidentes como modificaciones no deseadas en los productos, que pueden suponer costes humanos y económicos graves. Además, es necesario proteger la salud de todos aquellos relacionados con el producto y el sistema de producción (trabajadores, clientes, etc.) y también proteger el medioambiente. En ese sentido es deseable disponer de métodos que permitan controlar, monitorizar y minimizar todos los fallos que puedan ocurrir en una planta y mantener unos estándares y requisitos mínimos sobre las especificaciones del producto.

Otro aspecto a tener en cuenta al estudiar como implementar un control de calidad es que existen multitud de causas que pueden hacer variar el producto. Entonces si la variación va ligada al proceso se denomina como causa común y se pueden tratar estadísticamente, es decir, son predecibles, estables, etc. Por otro lado, si la causa de la variabilidad es externa al proceso, errática o puntual se define como causa especial; estas causas de variabilidad, generalmente, deben solucionarse modificando el sistema y con actuaciones locales respectivamente y son las variaciones que producen los fallos que se deberán detectar y diagnosticar.

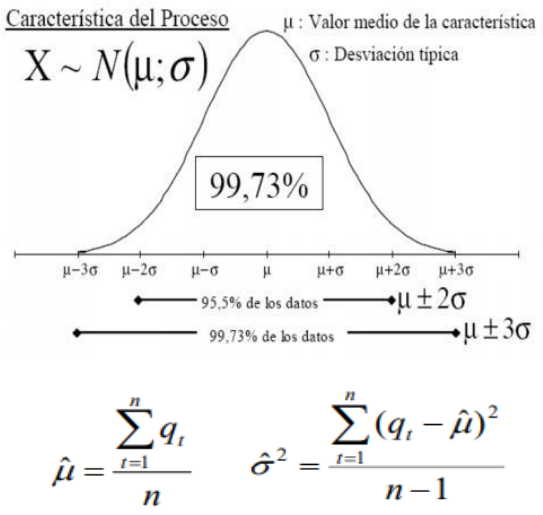


Figura 2 Característica del proceso.

Con las especificaciones y características del proceso se establecen unos límites inferiores (LPL) y superiores de proceso (UPL) y unos límites inferiores (LSL) y superiores (USL) de especificación. Cuando el proceso está dentro de todos los límites se dice que el proceso es estable y capaz, por lo que produce correctamente el producto y con pocas desviaciones. En cambio, cuando los límites de control son mayores que los de especificación aparecen errores y aunque el sistema sea estable y pueda seguir produciendo, no es capaz de reproducir el producto con la calidad y fidelidad adecuada (Figura 3). Al contrario, si el sistema se encuentra fuera de ambos límites pasa a estar fuera de control estadístico como se aprecia en la Figura 4, en este caso la salida del proceso no es predecible, y no se sabe cómo se comportará el proceso.

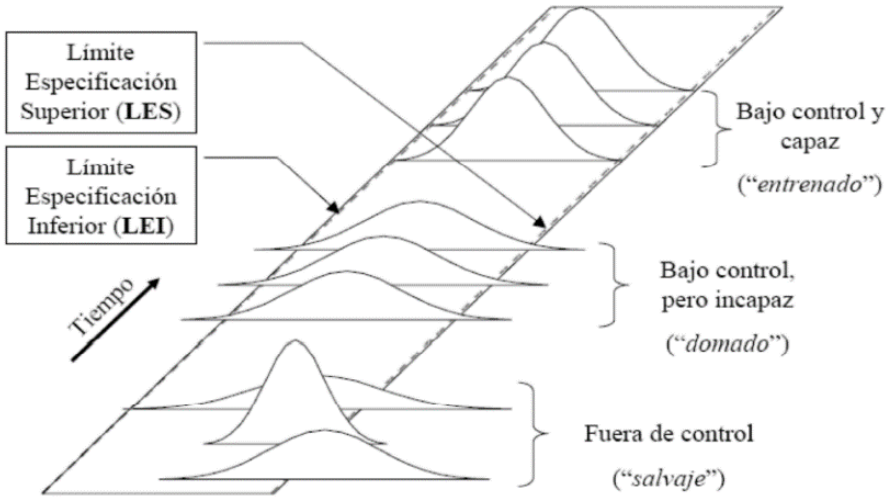


Figura 3 De arriba abajo, un proceso capaz y controlado, un proceso incapaz pero controlado, y un proceso fuera de control.

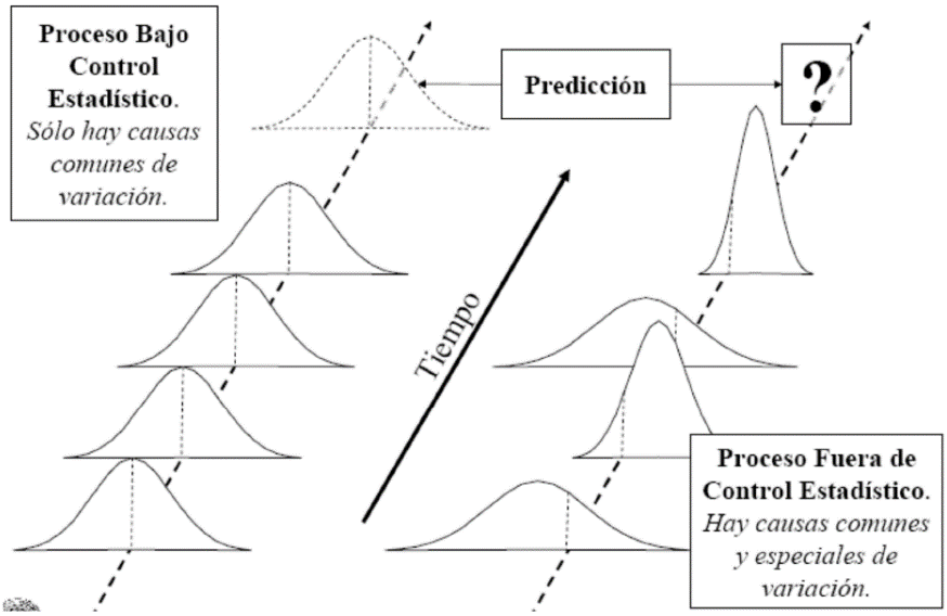


Figura 4 A la izquierda, un proceso bajo control fácilmente predecible. A la derecha, las causas especiales de variación impiden las predicciones.

Es necesario recalcar que para poder conseguir un buen control de calidad se ha de poder medir de forma precisa las variables del proceso utilizando instrumentación adecuada. Después trasmitir estos datos de forma rápida y fiable para no errar en los análisis que deban hacerse. Y, por último, es conveniente obtener un método que sea capaz de detectar si está ocurriendo un fallo o no se está cumpliendo con los estándares impuestos. De igual importancia es que además sea capaz de indicarnos por qué y donde está ocurriendo el fallo a lo largo de la cadena de producción. La influencia que se ejercen entre sí las variables del proceso impide definir o achacar a una variable concreta la causa que la ha producido, y es este el problema que se intenta solucionar al aplicar métodos FDI.

Por lo que se refiere al funcionamiento de un método FDI este puede dividir en tres fases principales: en primer lugar, la detección de la existencia de un fallo, en segundo lugar, su identificación y clasificación (que es el objeto de este trabajo) y por último las consecuencias producidas por el fallo. A su vez los métodos FDI se dividen en dos grupos, aquellos métodos basados en modelos y los métodos basados en el análisis de datos. Los métodos basados en modelos comparan la señal recibida de los sensores y la comparan con una señal generada matemáticamente por un modelo de la planta en cuestión, la diferencia entre las señales permite el análisis y la detección de fallos en el sistema, este sistema es una mejora de los métodos más tradicionales basados en tener varios sensores midiendo la misma variable. En cambio, los métodos basados en análisis de datos utilizan los datos extraídos de la planta y aplicándoles métodos matemáticos y estadísticos permite realizar control de calidad analizando la variabilidad del proceso. Además, la aparición de nuevos métodos de tratamiento de datos como el *Machine learning* y el *Deep learning*, que permiten el análisis de grandes cantidades de datos, permiten la aparición de nuevos métodos FDI. Es en los métodos basados en análisis de datos donde pertenecen los métodos nombrados anteriormente como PCA.

2.2 Análisis de Componentes Principales (PCA)

El método PCA es una técnica de proyección que usa transformaciones ortogonales para convertir un conjunto de variables correlacionadas en un conjunto de valores linealmente no correlacionados llamados componentes principales. Produce una reducción de la dimensionalidad del conjunto de datos originales, preservando su estructura de correlación. Seguidamente, en la Figura 5 podemos ver un ejemplo sencillo de reducción de la dimensionalidad, como un elemento en tres dimensiones puede reducirse a una dimensión menor.

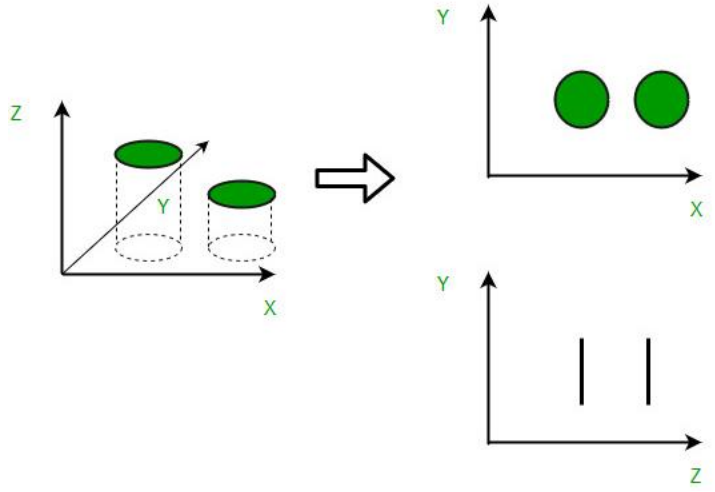


Figura 5 Reducción de la dimensionalidad

Se puede apreciar que en el caso de las dos nuevas proyecciones que aparecen son distintas entre sí, por lo que es necesario al aplicar estos métodos escoger correctamente como se quiere hacer la reducción. Como se ha dicho, el método PCA intenta reducir un número de dimensiones cualquiera, no obstante, se busca que las dimensiones o componentes resultantes sean aquellas que tengan una mayor variabilidad entre sí. Por consiguiente, serán más o menos según el número de dimensiones de la muestra y las variabilidades de los datos.

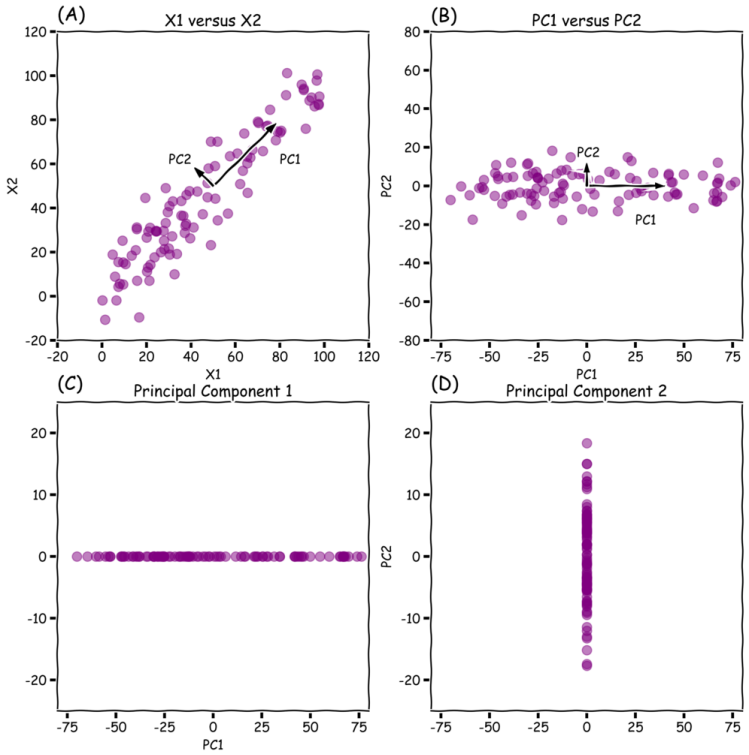


Figura 6 Variabilidad PCA 2 dimensiones.

Como se aprecia claramente en la Figura 6, de dos dimensiones y dos componentes, la dirección del componente PC1 representa la mayor parte de variabilidad de los datos y PC2 la menor. Por tanto, al pasar de dos dimensiones a una se debe escoger la componente PC1 la cual nos proporciona una menor dimensión perdiendo menos información que con la componente PC2. Se puede ver de la misma forma como en la figura siguiente (Figura 7) se reduce de tres dimensiones a dos escogiendo el plano que otorga la máxima variabilidad entre las dos componentes.

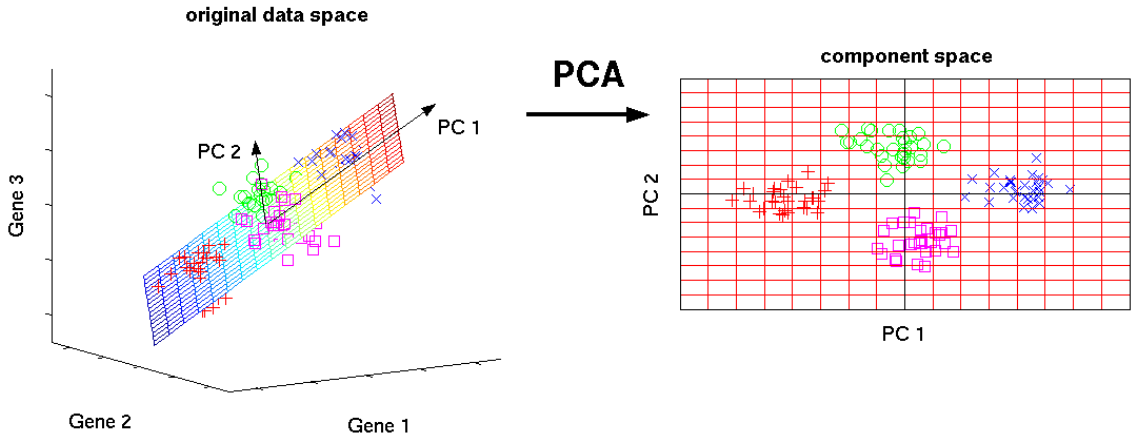


Figura 7 Variabilidad PCA 3 dimensiones.

En cuanto a la Figura 8 se puede apreciar como al aplicar el método PCA comentado anteriormente obtenemos una línea recta, sin embargo, debido a la poca linealidad de los datos perdemos un 23,23% de la información. Por otro lado, al aplicar el método SOM (*self-organizing map*), que es no lineal, la pérdida de información pasa a ser del 6,86%.

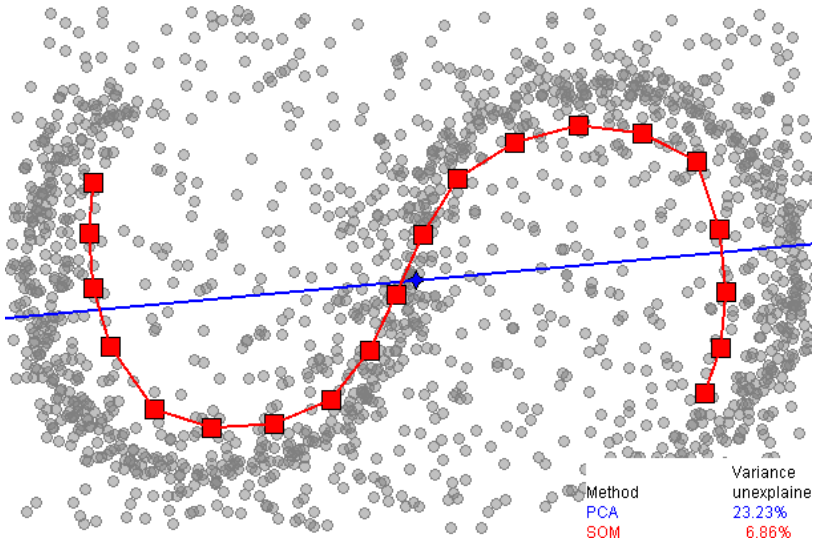


Figura 8 Comparación entre un método lineal y no lineal.

Una vez hemos visto que es el método PCA y utilizando los valores propios resultantes de la transformación realizada a los datos aplicando este método, entonces se puede extraer el porcentaje de variabilidad acumulada de los datos (Figura 9). Este índice nos indica cuanta variabilidad del sistema puede atribuirse a cada variable, es decir, cuanta responsabilidad tiene cada variable sobre los cambios que ocurren en el sistema. Por consiguiente, esto nos permite elegir el número de componentes principales necesarios para retener la variabilidad deseada en los datos originales, es decir, cuanto reducimos la dimensionalidad del problema. Por ejemplo, en los resultados obtenidos más adelante indican que de las 141 variables únicamente con 18 de ellas se podría explicar casi al completo el comportamiento del sistema. Cabe tener en cuenta que se trata de un método lineal aplicado sobre un sistema no lineal y que por esta razón existirá una pérdida de información al realizar el proceso.

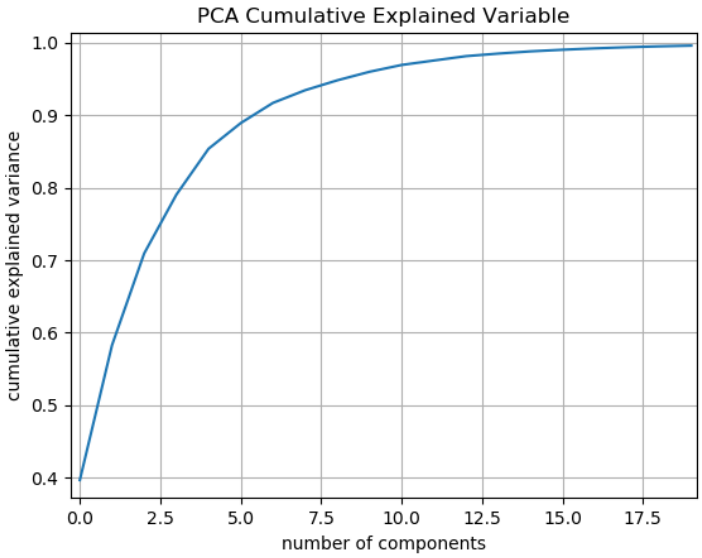


Figura 9 Variabilidad acumulada con cada variable.

2.3 Métodos *manifold*

Como ya hemos visto, el método PCA es capaz de reducir la dimensionalidad de un problema buscando una relación lineal entre las variables. No obstante, se debe destacar que normalmente los procesos industriales no son lineales, por lo que el método PCA no funcionará adecuadamente. Este es el principal motivo por el cual en este estudio se recurre a los métodos *manifold*, que se aplicarán al proceso tratando de reducir su dimensionalidad, pero preservando la estructura de los datos.

Estos métodos *manifold* utilizan las propiedades geométricas de la serie de datos de alta dimensión para implementar el siguiente algoritmo,

1. *Clustering*: Busca grupos de puntos similares, formando funciones que contienen puntos próximos.
2. Reducción de dimensionalidad: Proyecta los puntos en una dimensión inferior, pero preserva la estructura de la serie de datos.
3. Supervisión: Especificando puntos de la serie que tengan etiquetas y otros que no debe ser capaz de otorgar la misma etiqueta a dos puntos similares.

Es la distribución de los datos la que determina si dos puntos son o no similares,

- Punto de vista probabilístico: A mayor densidad de puntos, menor es la distancia entre ellos.
- Punto de vista del grupo: Los puntos conectados en las regiones comparten las mismas propiedades.
- Punto de vista del *manifold*: La distancia entre puntos se debe medir siguiendo los datos del *manifold*.
- Versión mixta: Dos puntos similares son aquellos conectados por una distancia pequeña en zonas de alta densidad de puntos.

2.3.1 ISOMAP

ISOMAP es un método no lineal para la reducción de dimensionalidad que además es capaz de preservar las propiedades globales de la serie de datos que se analiza. Se destaca por incluir el cálculo de las distancias geodésicas entre puntos en lugar de utilizar la distancia euclídea. En particular la distancia geodésica en ISOMAP viene definida como la suma de los pesos de los bordes a lo largo del camino más corto entre dos puntos. Así mismo, ISOMAP representa las coordenadas del nuevo espacio euclidiano como los vectores propios superiores de la matriz de distancia geodésica.

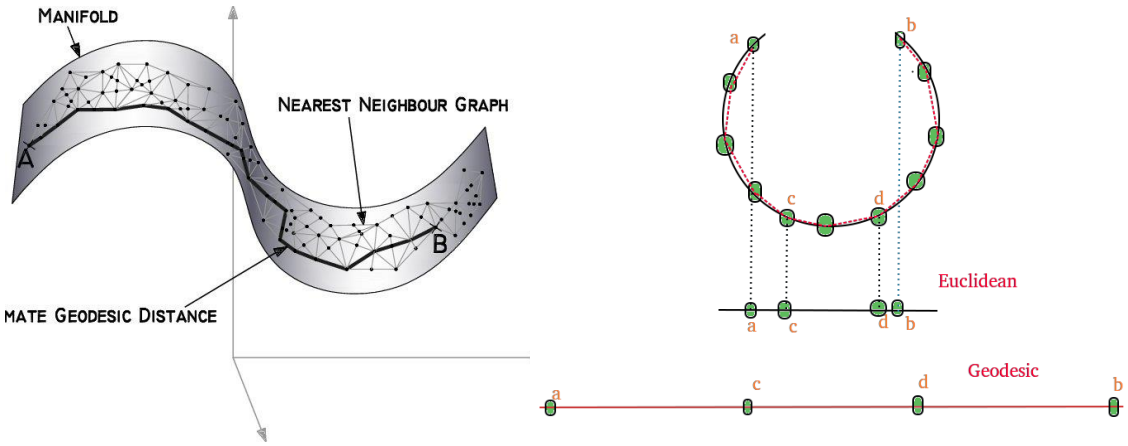


Figura 10 Distancia geodésica.

De forma resumida ISOMAP opera con el siguiente algoritmo,

1. Determina los vecinos cercanos, existen diferentes métodos para hacerlo.
    a. Todos los puntos dentro de un radio fijado.
    b. Los k vecinos más cercanos.
2. Construir el mapa de vecinos.
    a. Cada punto conectado con otro es un vecino k cercano.
    b. La longitud de borde es igual a la distancia euclídea.
3. Calcula la distancia más corta entre dos nodos.
    a. Algoritmo de Dijkstra, Floyd-Warshall, etc.
4. Calcula la reducción de dimensionalidad.
    a. Algoritmo de escalamiento multidimensional.

Cabe destacar que durante las pruebas realizadas se fijarán diversos parámetros, como el número de coordenadas que seleccionará para obtener una representación en 2 dimensiones. De modo que en cada grupo de pruebas se especificará que parámetros se han escogido.

2.3.1.1 Algoritmo Dijkstra

Uno de los primeros y más famosos algoritmos para encontrar la distancia más corta entre dos nodos es el algoritmo de Dijkstra, desarrollado por Edsger W. Dijkstra en 1956 [8], también llamado algoritmo de caminos mínimos. El objetivo de este algoritmo es trazar el camino más corto dado un vértice origen, hacia el resto de los vértices en un grafo que tiene pesos en cada arista. En otras palabras, la idea general es explorar todos los caminos más cortos posibles que parten del vértice origen y que van al resto de vértices y una vez se ha obtenido el camino más corto al vértice final el algoritmo se detiene. El algoritmo opera de la siguiente forma.

1. Se escoge el nodo inicial y se asignan las distancias a los nodos vecinos.

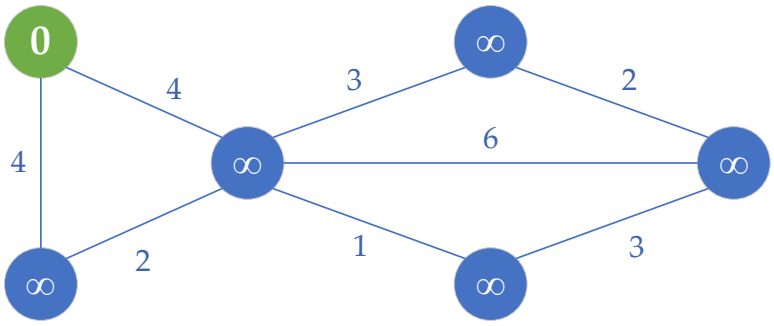


Figura 11 Paso 1 del algoritmo Dijkstra.

2. Se asigna al vértice el valor de la distancia que hay que recorrer para llegar hasta él.

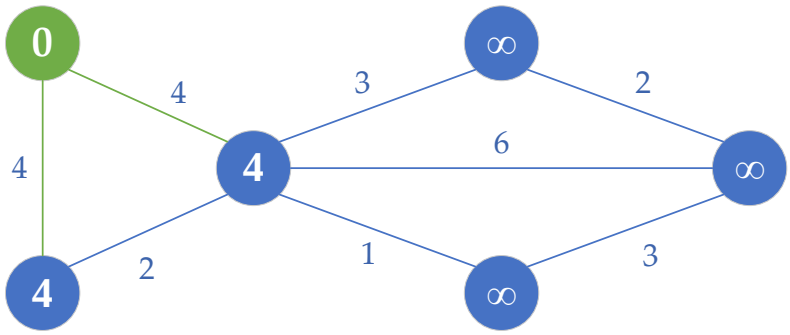


Figura 12 Paso 2 del algoritmo Dijkstra.

3. Se calculan las distancias al nuevo vértice asignándole la distancia más corta posible.

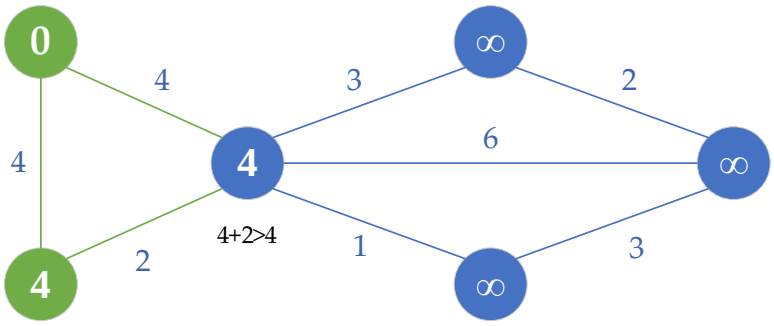


Figura 13 Paso 3 del algoritmo Dijkstra.

4. Se repite el proceso evitando los vértices que ya se han visitado.

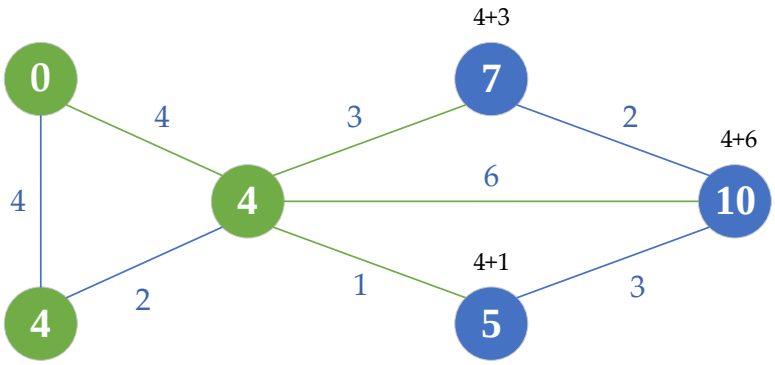


Figura 14 Paso 4 del algoritmo Dijkstra.

5. Tras cada iteración se escoge el vértice no visitado que tenga el camino más corto.

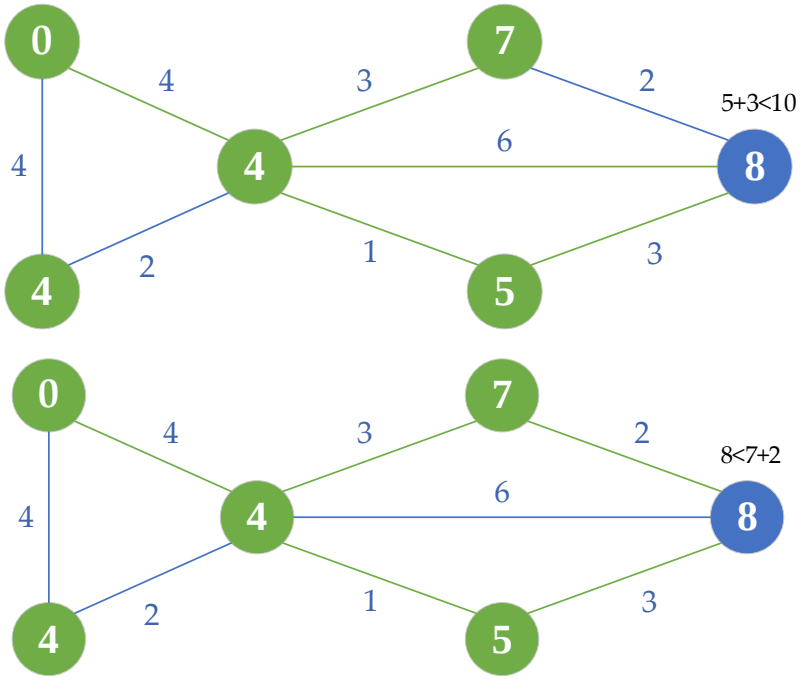


Figura 15 Paso 5 del algoritmo Dijkstra.

6. Por último, se repiten los cálculos hasta a ver visitado todos los vértices.

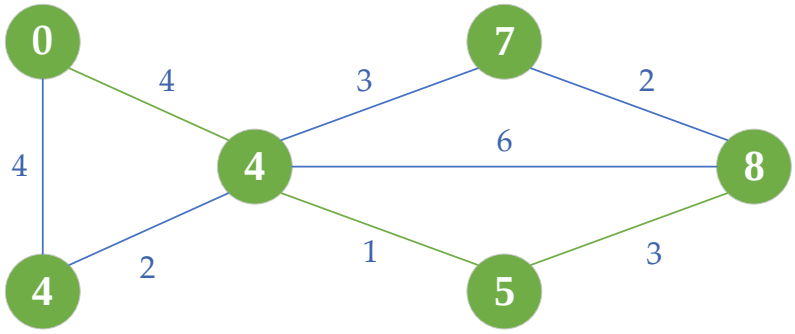


Figura 16 Paso 6 del algoritmo Dijkstra.

Existen otros métodos para encontrar el camino más corto entre dos nodos y de hecho la función ISOMAP utilizada en la fase experimental permite utilizar además del algoritmo Dijkstra, el algoritmo de Floyd-Warshall. No obstante, la función escogida selecciona automáticamente el método utilizado, debido a que la elección de los algoritmos afecta únicamente a la velocidad de ejecución, porque de una forma u otra, ambos métodos acabarán calculando el camino óptimo.

2.3.2 LLE (Locally Linear Embedding)

En el método Locally Linear Embedding (LLE) la idea principal es que el conjunto de datos altamente dimensional puede aproximarse a un conjunto de baja dimensionalidad. Para lograr esta aproximación se crean pequeños parches o grupos del *manifold*, de manera que cada uno contiene un grupo de datos del conjunto al que se le

pueden agregar coordenadas locales. Simultáneamente estas coordenadas locales se pueden relacionar con las coordenadas globales del conjunto de alta dimensionalidad. De ahí que el objetivo sea encontrar una transformación entre la alta y la baja dimensionalidad utilizando la información que nos ofrecen los puntos cercanos entre sí. Resumiendo, el método LLE opera de la siguiente forma,

- Busca los k vecinos más cercanos de cada punto del conjunto de alta dimensionalidad.
- Construir una matriz de pesos W con los k vecinos más cercanos.
- Crear el conjunto de baja dimensionalidad a partir de la matriz de pesos W.

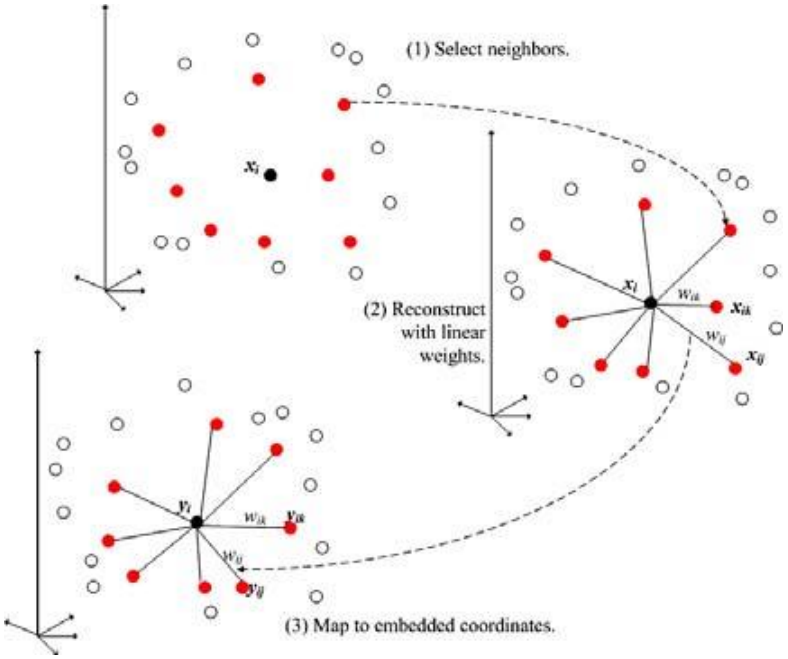


Figura 17 Conexión entre vecinos de LLE.

A partir del algoritmo mostrado se observa que el número de *k* vecinos afecta de la misma manera a la transformación de los datos que en el método ISOMAP. Específicamente, si el número de vecinos escogido es muy bajo se crearán más parches de los necesarios. Por el contrario, si es demasiado grande se agruparán datos que no deberían agruparse.

De ahí que el cálculo de los *k* vecinos sea factor importante, porque si el número de vecinos es demasiado grande es posible que se realicen conexiones entre puntos que no deberían estar conectados, creando cortocircuitos en la transformación ISOMAP o LLE. Por el contrario, si el número de vecinos es muy bajo se crearán desconexiones y divisiones dentro del mismo grupo de vecinos. Por norma general mientras mayor es el número de vecinos, mayor es coste computacional. Por otro lado, el número de vecinos utilizado suele encontrarse entre 8 y 12 [9] para conjuntos de datos con una

dimensionalidad no demasiado alta el número. No obstante, en sistemas con más dimensionalidad no hay nada especificado sobre el número óptimo de vecinos, por lo que las transformaciones realizadas en la fase experimental se deberá variar el número de vecinos con el objetivo de lograr la mejor transformación posible.

2.4 Algoritmo de *clustering* basados en densidad

Para crear una clasificación de los fallos con datos con poca linealidad una de las opciones es aplicar algoritmos de *clustering* o agrupamiento basados en densidad, que crean diferentes núcleos o grupos de datos dentro de una misma serie. De manera que, según el tamaño de los grupos, cantidad de puntos y ruido que se obtengan al aplicar el algoritmo se puede tratar de realizar alguna clasificación de cada tipo de fallo y además si el método fuera suficientemente preciso también identificar la intensidad. De la misma manera que con los métodos *manifold*, se pueden ajustar varios parámetros de cada algoritmo para ajustarlos a las necesidades del problema.

Existen diferentes tipos de algoritmos de agrupación, se puede apreciar en la Figura 18 una comparación entre diferentes tipos de estos algoritmos. Cabe destacar que es importante escoger el tipo de algoritmo según el objetivo marcado y los datos que se tienen que tratar. Es preciso mostrar que no todos los algoritmos de densidad funcionan siguiendo los mismos principios. Hay que tener en cuenta que existen diferentes maneras de separar los datos en agrupaciones, una de las maneras es crear centroides entre los vecinos, como por ejemplo el método *AffinityPropagation*, otra forma es realizar agrupaciones jerárquicas, utilizando matrices de conectividad, etc. De la misma manera, se pueden lograr agrupaciones analizando la densidad de los datos siendo esta la técnica escogida para este trabajo.

Para llevar a cabo los experimentos en este TFM, los algoritmos de agrupamiento basados en densidad elegidos son los métodos DBSCAN, OPTICS y una mejora de ambos llamada HDBSCAN. Cada método tiene sus propios parámetros y será necesario realizar una buena selección de estos si queremos obtener resultados válidos [10].

Una vez se procesan los datos utilizando los algoritmos de densidad, se puede tratar de definir lo bien que ha funcionado un algoritmo utilizando diferentes métodos. Para analizarlos algoritmos existen dos opciones principales:

1.- Externas: Dependen de una base de datos previa o expertos.

- Pureza
- Índice de Rand
- Índice de Jaccard
- Matriz de confusión
- Etc.

2.- Internas: Se evalúan los datos con los que se está trabajando, aplicando herramientas matemáticas

- Coeficiente de Silueta
- Índice de Davies-Bouldin
- Evaluación de Calinski-Harabasz

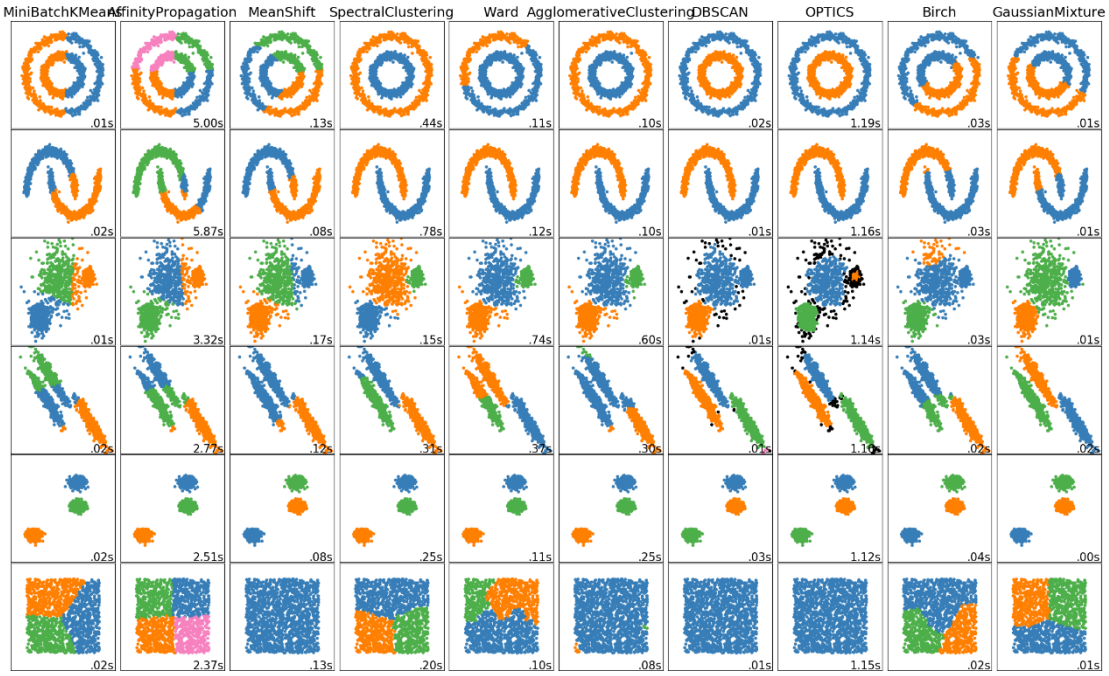


Figura 18 Tabla comparativa de algoritmos de agrupación [11].

Este trabajo se centra en las herramientas internas, ya que no se dispone de bases de datos o expertos que puedan aportar información sobre los datos que se están utilizando, por lo que es necesaria una breve explicación sobre los métodos en cuestión.

- Coeficiente de Silueta

En primer lugar, se explica el coeficiente de silueta que se basa en la comparación de estanqueidad y separación [12] de los datos. Contrasta la distancia promedio a los elementos en el mismo grupo con la distancia promedio a los elementos en otros grupos. Los objetos con un valor de silueta alto se consideran bien agrupados, los objetos con un valor bajo pueden ser valores atípicos, el rango va de -1 a 1.

- Índice de Davies-Bouldin

A continuación, se muestra el índice de Davies-Bouldin que indica lo buena es la agrupación, el cálculo se realiza utilizando cantidades y características inherentes al conjunto de datos. Este índice tiene el inconveniente de que un buen valor dado por este método no implica recuperación de la mejor información. Debido a la forma en que se

define, en función de la relación de la dispersión dentro del grupo y a la separación entre grupos, un valor más bajo significará que la agrupación es mejor.

- Evaluación de Calinski-Harabasz

Por último, la Evaluación de Calinski-Harabasz es un indicador del número óptimo de agrupaciones que se han realizado. mientras mayor sea el valor más se aproxima al óptimo. No existen unos rangos definidos, sino que se debe comparar el número con otros obtenidos sobre los mismos datos. El índice de Calinski-Harabasz se basa en la relación entre la varianza entre los grupos y la varianza dentro de los grupos, de manera que un valor mayor del cociente indica una mejor partición [13].

2.4.1 DBSCAN

En primer lugar, se analiza DBSCAN dado que tanto OPTICS como HDBSCAN son modificaciones de este. Conviene subrayar que DBSCAN es un algoritmo con bastante relevancia en el campo del *machine learning* y la minería de datos. En síntesis, DBSCAN utiliza la distancia Euclidiana para buscar los puntos cercanos entre sí y que formen un grupo con cierto número mínimo de puntos. Además, es capaz de distinguir zonas de baja densidad [14].

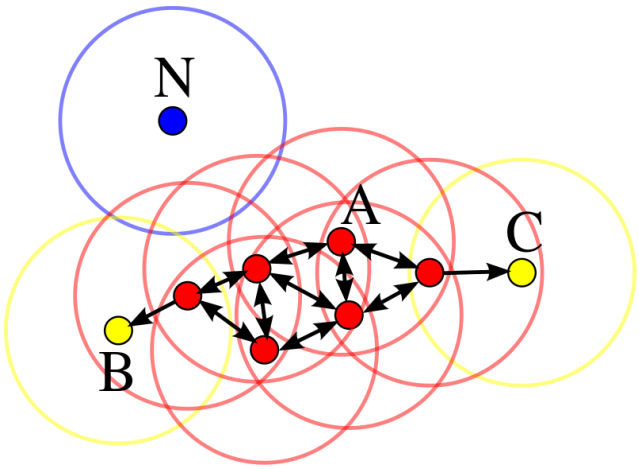


Figura 19 Los puntos marcados como A son puntos núcleo. Los puntos B y C son densamente alcanzables desde A y densamente conectados con A, y pertenecen al mismo clúster. El punto N es un punto ruidoso que no es núcleo ni densamente alcanzable.

Cabe destacar que este algoritmo de *clustering* tiene varias ventajas como son: que no es necesario especificar el número de grupos deseados, puede encontrar grupos de formas arbitrarias, distingue los valores atípicos y los clasifica como ruido. Además, puede utilizarse en bases de datos mejorando el rendimiento.

Por otro lado, y como cualquier método tiene desventajas, por ejemplo, los puntos fronterizos entre dos grupos pueden cambiar según la distribución y el orden de los datos creando pequeñas variaciones en los resultados. Además, es menos eficaz si los datos tienen diferencias de densidades muy grandes dado que no es capaz de ajustar los parámetros dinámicamente. También es poco eficaz si los datos son de muy alta dimensionalidad (miles de dimensiones) produciéndose el efecto Hughes [15]. En consecuencia, si no se conocen los datos con los que se está tratando puede ser complicado ajustar los parámetros.

En consecuencia, en la fase experimental se partirá de un punto inicial en el que se generen grupos de forma aceptable y se realizarán diferentes transformaciones variando los parámetros sistemáticamente. Puesto que existen infinitas combinaciones se escogerán valores interesantes hasta obtener resultados aceptables. Debe tenerse en cuenta que para cada caso existen unos parámetros óptimos.

2.4.2 HDBSCAN

En segundo lugar, se utilizará HDBSCAN que es una mejora de DBSCAN. Tienen en común que en ambos casos el algoritmo empieza buscando las distancias entre un punto y su vecino más alejado definido por el parámetro *minPoints*. A continuación, es donde difieren ambos métodos y es que el algoritmo HDBSCAN, al igual que OPTICS [16], opera con densidades de grupos variables para delimitarlos, eliminando así el defecto principal del DBSCAN que recordemos es la falta de variación dinámica de los parámetros.

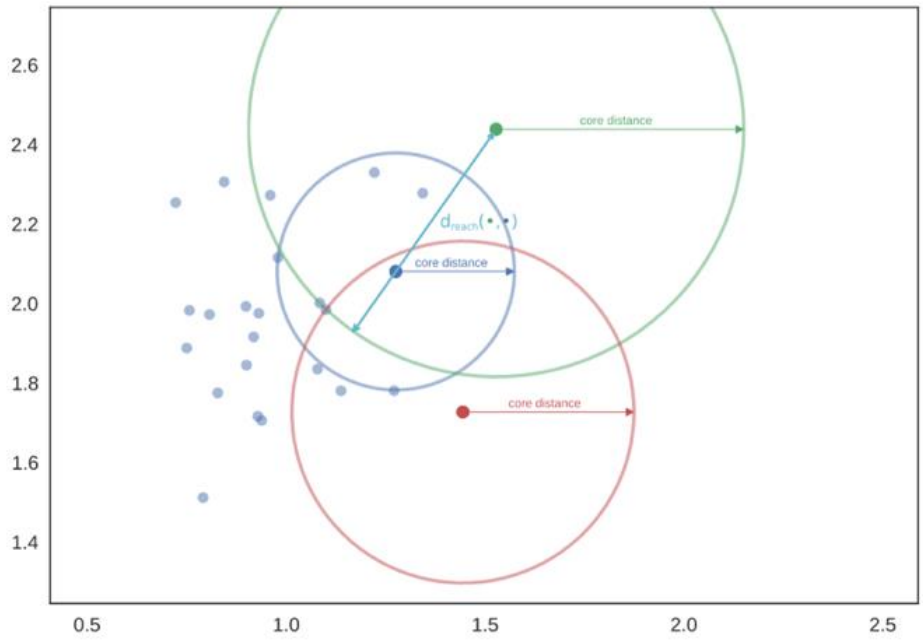


Figura 20 Funcionamiento HDBSCAN.

2.4.3 OPTICS

Por último, se analiza el método OPTICS el cual funciona de la misma forma que DBSCAN. En el caso de OPTICS se intenta resolver una de las mayores desventajas que tiene DBSCAN que es ser capaz de trabajar con datos que tienen densidades diferentes. Además, OPTICS, mantiene la jerarquía de los grupos para lograr un radio de agrupación variable [17].

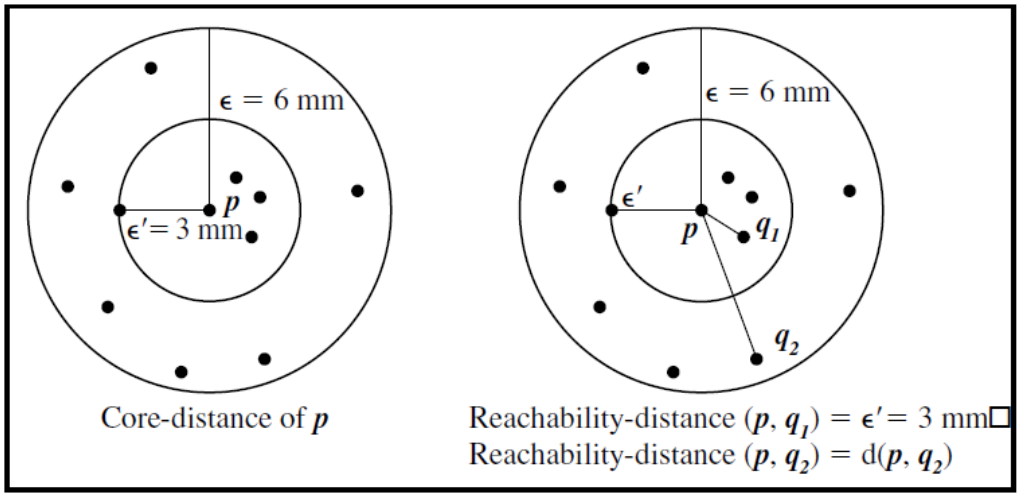


Figura 21 Funcionamiento OPTICS.

2.5 Python

Por lo que se refiere a Python es el lenguaje de programación que se ha utilizado para llevar a cabo las pruebas experimentales en este trabajo. En pocas palabras, Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Así mismo, se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Además, es un lenguaje interpretado, dinámico y multiplataforma [18].

Otra ventaja de Python es su modularidad y extendido uso, que permite utilizar una gran cantidad de librerías creadas por otros usuarios. Cabe destacar que existen diversas librerías básicas que gozan de la robustez comparable a las de otras plataformas como *Matlab*, entre ellas se encuentra *Scikit-Learn* [19], una librería orientada al *machine learning* que contiene funciones enfocadas a la clasificación de grupos incluidas ISOMAP [20] y LLE [21] [22]. Además *Scikit-Learn* contiene los algoritmos de agrupación utilizados en este trabajo, DBSCAN [23] y OPTICS [24], a excepción de HDBSCAN que

lo contiene una librería independiente [25] [26], aunque hay que subrayar que está basado en el método DBSCAN de *Scikit-Learn*. Además de estas librerías principales se utilizarán otras librerías auxiliares que suelen estar presentes en cualquier tipo de programación científica realizada en Python. Por ejemplo, Numpy que es una de las principales librerías científicas para Python y permite tratar objetos matemáticos como vectores, matrices, manipulación matemática, lógica, de formas, clasificación, selección, entradas/salidas, transformadas discretas de Fourier, álgebra lineal básica, operaciones estadísticas básicas, simulación aleatoria, etc. [27] ofreciendo un funcionamiento similar fundamental al de *Matlab* [28]. Para el tratamiento de datos se utiliza la librería Pandas que permite gestionar y procesar las series de datos de fallos que necesitamos para realizar los experimentos [29]. Y por último la librería Pyplot [30] mejora la creación y modificación de gráficas de Python y las hace semejarse a la que utiliza *Matlab*.

Por último, para crear correctamente los algoritmos que nos permitan cumplir los objetivos es necesario conocer las funciones que debemos utilizar. Durante las pruebas que realizaremos la mayoría de parámetros estarán fijos, como el número de coordenadas que será 2, para obtener una representación en 2 dimensiones. Como ya se ha comentado en cada grupo de pruebas se especificará que parámetros se han escogido.

Capítulo 3.-Planta de estudio

La planta utilizada para los experimentos se trata de un modelo de una Estación Depuradora de Aguas Residuales (EDAR). La característica principal de estas plantas es que tienen la capacidad de eliminar los contaminantes que se encuentran en el agua que se evacua de los hogares, poblaciones e industrias. Para lograr esta meta se realizan varios procesos físicos y químicos para lograr que no sean (o lo menos) contaminantes.

En cuanto a las aguas residuales pueden contener impurezas que por su estado físico se pueden distinguir entre:

- Fracción suspendida: aquellas partículas no solubles que se dispersan en el agua.
- Fracción coloide: Las partículas que forman espumas como grasas o aceites.
- Fracción soluble: Las partículas que quedan disueltas en el agua.

De la misma manera, las impurezas, que forman el 1% (normalmente) del agua residual, se clasifican en dos grupos:

- Sólidos orgánicos: formados principalmente por nitrógeno, fósforo, cloruros, sulfatos, carbonatos, bicarbonatos y algunas sustancias tóxicas como arsénico, cianuro, cadmio, cromo, cobre, mercurio, plomo y zinc.
- Sólidos inorgánicos: Los sólidos orgánicos se pueden clasificar en nitrogenados y no nitrogenados. Los nitrogenados, es decir, los que contienen nitrógeno en su molécula, son proteínas, ureas, aminas y aminoácidos. Los no nitrogenados son principalmente celulosa, grasas y jabones.

Así mismo el proceso de depuración consta de 5 pasos principales, en los que se eliminan del agua las impurezas de mayor tamaño a menor, desde cualquier trozo de mueble, plástico, latas, etc. a las partículas microscópicas.

1.- Pretratamiento

- Se eliminan los objetos de mayor tamaño, utilizando normalmente barras de acero que los retienen, para posteriormente retirarlos. Parte de este proceso se lleva a cabo antes de que el agua entre en el alcantarillado de las ciudades. Según la ratio de acumulación se programan las retiradas de los residuos que se trasportan a plantas especializadas (incineradoras, plantas de reciclaje, etc.) para su tratamiento.
- Sedimentación de partículas pesadas como grava, arena, materia orgánica (partículas mayores de unos 0,2 milímetros). Los principales objetivos de esta fase son: reducir los sedimentos que se puedan formar en los tanques,

reducir las limpiezas necesarias de la planta por acumulación de sedimentos y proteger los elementos móviles del desgaste por abrasión añadido.

- Igualación de caudales para mantener un sistema lo más estacionario posible y que no dependa tanto de si aumenta o disminuye en exceso la aportación de agua. Esto se logra almacenando el agua sobrante en tanques.
- Eliminación de grasas y aceites que puedan quedar en la superficie y no decante.

2.- Tratamiento primario

- Se introduce agua en tanques de sedimentación donde por gravedad los elementos más pesados precipitan al fondo y los aceites y grasas suben a la superficie, donde se retiran mecánicamente para su tratamiento.
- Los sólidos precipitados se evacuan por el fondo del tanque y se traslada a los centros de tratamiento.

3.- Tratamiento secundario

- Se utilizan métodos biológicos aeróbicos donde las bacterias y protozoos consumen la materia orgánica contaminante y la transforman permitiendo que se produzca el proceso de *floculación*, que transforma la materia en partículas que precipitan al fondo del tanque permitiendo su separación.
- Existen dos métodos principales para llevar a cabo este tratamiento y son los sistemas de crecimiento adjunto, que utiliza varias etapas y es más adaptable. Y los sistemas de crecimiento suspendido que incluyen la depuración biológica por fango activo que consiste en un único depósito que contiene las bacterias, agitado, aireado y alimentado con agua residual.

4.- Tratamiento terciario

- Se filtra con arena las partículas que todavía puedan quedar y posteriormente con carbón activo se eliminan las toxinas residuales.
- Se almacena el agua en lagunas o estanques donde plantas acuáticas y/o pequeños invertebrados continúan eliminando partículas.
- En este punto el agua esta eutrofizada, contiene un exceso de minerales y nutrientes, nitrógeno y fósforo principalmente, que de ser devuelta a la naturaleza podría crear una proliferación de algas generando un aumento de la biomasa y pérdida de la diversidad que puede acabar intoxicando a la fauna local al tener que consumir estas algas. Por lo tanto, se aplican procesos para la eliminación del exceso de minerales y nutrientes.
- La eliminación del nitrógeno se realiza a través de un proceso biológico donde primero una bacteria oxida el amoniaco que contiene el nitrógeno y

lo transforma a nitrito el cual se ve transformado por otra bacteria a nitrato. Una vez conseguido el nitrato se desnitrifica utilizando aguas anóxicas (donde el oxígeno disuelto está agotado), que permite que se forme nitrógeno en gas que es finalmente liberado en la atmósfera.

- La eliminación del fósforo puede realizarse de diversas formas, por osmosis inversa, donde se utiliza una membrana para filtrar el agua con un proceso complejo, de forma biológica donde se utilizan bacterias específicas que son capaces de almacenar fósforo en sus células y que posteriormente se utilizan como abono o de forma química donde se mezcla con sales de hierro, aluminio o cal, el cual anula el fosforo produciendo más fango que se debe retirar.
- Para finalizar esta fase se realiza una desinfección del agua para reducir el número de microorganismos que contiene el agua después de todos los tratamientos. Existen diversos métodos para la eliminación de microorganismos.
    a. Cloración del agua para desinfectar el agua. Es un método barato y eficaz pero que puede formar compuestos orgánicos clorados que sean cancerígenos o dañinos para el medio ambiente, por lo que se necesitaría tratar estos elementos también.
    b. Rayos ultravioletas que causen daño en la estructura genética de los microorganismos sin necesidad de utilizar otros productos en el agua, aunque el coste de mantenimiento es elevado.
    c. Tratamiento con ozono que oxida la materia orgánica y elimina los microorganismos, aunque es menos tóxico que el cloro su utilización es más cara.

5.- Tratamiento final

- Puede ser necesario un tratamiento final si se detectan rastros de elementos farmacéuticos u otros elementicos químicos contaminantes que no puedan ser tratados con las estrategias convencionales. Sería necesario pues aplicar métodos y filtrados adicionales para evitar contaminar la masa de agua donde desagüe el agua tratada.
- En las primeras etapas del tratamiento se pueden generar malos olores que se pueden eliminar con reactores de carbón o con sales de metálicas.

Como se ha comentado en el 0 los datos empleados para la realización de los experimentos pertenecen a un sistema que simula todo el proceso que se acaba de explicar. El modelo es benchmark de una planta EDAR, muy utilizado en la literatura científica, llamado BSM2, desarrollado por la International Water Association (IWA), [7].

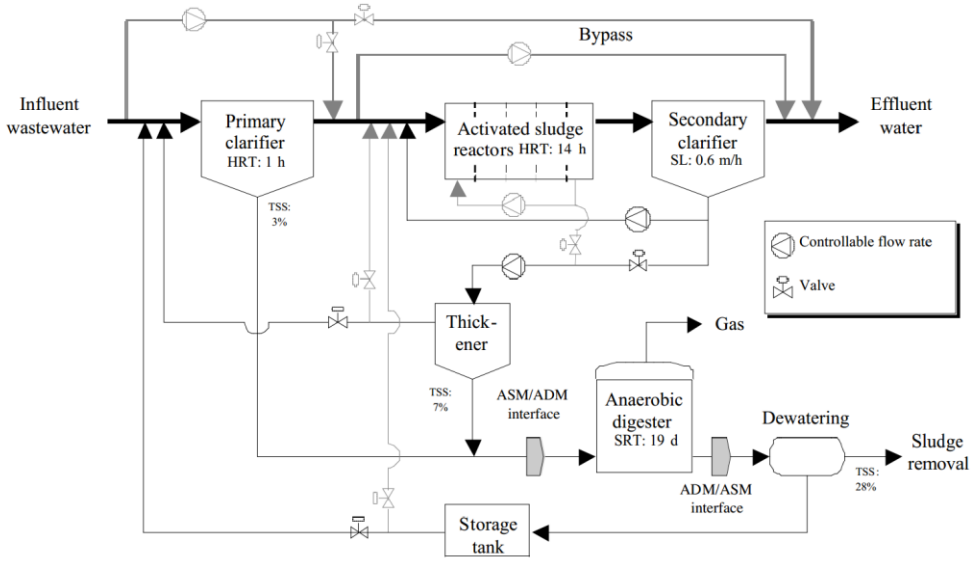


Figura 22 Modelo de la depuradora.

En particular se han extraído los siguientes datos, medidas de 141 sensores, tomados con un período de muestreo de 15 minutos, durante 609 días. Hay datos de comportamiento normal, y se tienen 16 series de fallos resumidas aquí:

- 17 series de datos
  a. 1 sin fallo (fallo 0)
  b. 16 con fallos (ocurren en el día 200)
     i. Sensor de oxígeno 4º reactor, magnitud del 70%
     ii. Sensor de oxígeno 4º reactor, magnitud del 50%
     iii. Sensor de oxígeno 4º reactor, magnitud del 20%
     iv. Sensor de oxígeno 4º reactor, magnitud del -20%
     v. Sensor de oxígeno 4º reactor, magnitud del -50%
     vi. Alcalinidad del influente, magnitud del -50%
     vii. Alcalinidad del influente, magnitud del 20%
     viii. Alcalinidad del influente, magnitud del 40%
     ix. Alcalinidad del primer reactor, magnitud del 10%
     x. Alcalinidad del primer reactor, magnitud del 20%
     xi. Alcalinidad del primer reactor, magnitud del -30%
     xii. Caudal de salida del decantador, magnitud del -50%
     xiii. Caudal de salida del decantador, magnitud del -30%
     xiv. Fuga en el tanque de almacenamiento, magnitud del -50%
     xv. Reparto de flujos entre Qr y Qw, magnitud del -50%
     xvi. Reparto de flujos entre Qr y Qw, magnitud del -25%

Los 141 sensores se dividen en grupos de 21 medidas que se recogen a lo largo de la planta, es decir, únicamente se miden las variables de la Figura 23, pero se realizan las mediciones en diferentes puntos de la planta.

1. SI(inert soluble material, g COD.m^{-3});
2. SS(readily biodegradable substrate, g COD.m^{-3});
3. XI(inert particulate material, g COD.m^{-3});
4. XS(slowly biodegradable substrate, g COD.m^{-3});
5. XB,H (heterotrophic biomass, g COD.m^{-3});
6. XB,A (autotrophic biomass, g COD.m^{-3});
7. XP(inert particulate material from biomass decay, g COD.m^{-3});
8. SO (dissolved oxygen, g (-COD).m^{-3});
9. SNO(nitrate and nitrite, g N.m^{-3});
10. SNH (ammonia and ammonium, g N.m^{-3});
11. SND(soluble organic nitrogen associated with SS, g N.m^{-3});
12. XND(particulate organic nitrogen associated with XS, g N.m^{-3});
13. SALK(alkalinity);
14. TSS (total suspended solids, g SS.m^{-3});
15. flow rate, $m^3.d^{-1}$;
16. temperature, °C;
17. dummy variable (soluble) no 1;
18. dummy variable (soluble) no 2;
19. dummy variable (soluble) no 3;
20. dummy variable (particulate) no 1;
21. dummy variable (particulate) no 2

Figura 23 Esquema de variables del modelo.

Los experimentos realizados se plantean como si el sistema funcionará en tiempo real y se obtuvieran las medidas en el momento de funcionamiento (*on-line*). Por lo tanto, no es posible obtener todas las medidas descritas en la Figura 23, debido a que varias de las medidas es necesario obtenerlas a través de análisis en el laboratorio y no se podría cumplir con el análisis en tiempo real. Como resultado, las variables utilizadas en los experimentos son las siguientes:

- DQO: demanda química de oxígeno en g COD.m^{-3}. Es la suma de los valores de las 4 primeras variables: SI (materia orgánica inerte soluble), SS (substrato de biodegradación rápida), XI (materia orgánica inerte insoluble) y XS (substrato de biodegradación lenta).
- O2: oxígeno disuelto en el líquido en g (-COD).m^{-3}, se corresponde con el elemento 8 de la lista de variables.
- Salk: mide la alcalinidad del fluido, es la variable número 13.

- N: mide el nitrógeno presente en el agua residual en g $N.m^{-3}$, es la suma de las variables 9, 10 y 11; SNO (contenido en nitratos y nitritos), SNH (contenido de amoníaco y amonio) y SND (nitrógeno orgánico soluble asociado a SS).
- SS: sólidos suspendidos en g $N.m^{-3}$, corresponde a la variable número 14.
- Caudal: caudal del fluido en $m3d^{-1}$, variable número 15.
- Temperatura: mide la temperatura del agua en ºC, es la variable 16 de la lista.

En conclusión, se tienen entonces, 7 variables tomadas en 20 puntos diferentes, que da un total de 140 variables; más una variable extra que es la señal de control Kla siendo finalmente 141 variables a analizar. Así mismo, se estudiarán todas las variables al mismo tiempo, pues lo que interesa es saber dónde a ocurrido el fallo en toda la planta. No obstante, otra aproximación válida sería analizar cada grupo de variables por separado, aplicando técnicas o modelos basados en agentes e identificando en cada sector de recogida de datos los fallos que pudieran aparecer.

Capítulo 4.- Propuesta experimental

4.1 Introducción

Con respecto a la detección de fallos con métodos lineales existen multitud de experimentos y métodos bien validados que permiten conocer cuando ocurre un fallo. Por el contrario, con los métodos no lineales de reducción de dimensionalidad apenas se ha investigado en el campo de la detección de fallos. Aunque si se han estudiado cómo funcionan los métodos *manifold* y los algoritmos de agrupamiento, hay que subrayar que los estudios encontrados en la literatura se han enfocado más en la reducción de dimensionalidad y en la clasificación de imágenes para los métodos *manifold* [31]. Mientras que para los algoritmos de agrupamiento se ha centrado en lograr métodos que sean capaces de clasificar correctamente grupos previamente definidos [32], como por ejemplo clasificar las especies de flores que existen en los datos de Iris (Figura 24), un conjunto de datos muy utilizados para probar métodos de clasificación encontrados en la literatura científica.

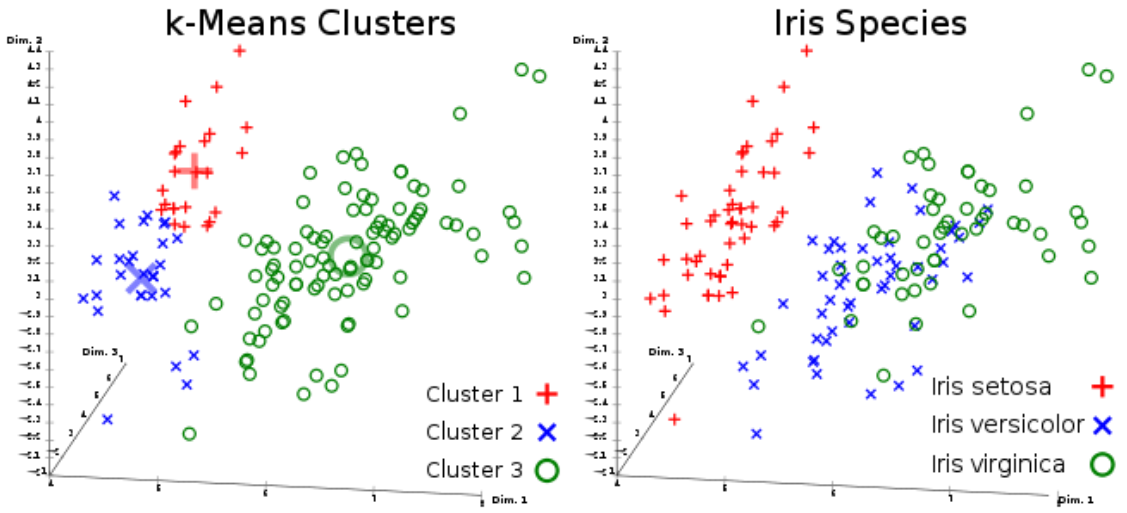
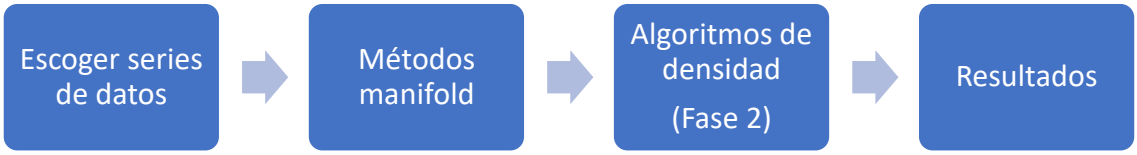


Figura 24 Ejemplo comparativo del algoritmo de agrupamiento K-Mean Cluster aplicado al Iris Flower data set de Ronald Fisher [33].

En cuanto a las pruebas realizadas en este trabajo se centran, en una primera instancia, en lograr clasificar los fallos correcta y sistemáticamente. En general, no se parte de ninguna condición, como por ejemplo que se tenga que realizar una clasificación correcta con un número mínimo de muestras o con una precisión determinada. Debido

a que el primer paso lógico antes de poder condicionar el proceso de clasificación es asegurarse de que el método es viable, robusto y sistemático.

Por otro lado, las pruebas realizadas siguen un esquema general a pesar de que sufran variaciones como el número de parámetros de cada método o el entrenamiento de las transformaciones.



Finalmente se realizan dos fases de pruebas,

1. La primera fase se centra en analizar la aplicación de los métodos *manifold*, tanto por tipo de fallos y entrenamiento de los métodos, como por evolución temporal, sin aplicar los algoritmos de densidad, tratando de obtener agrupaciones de forma general o analizando si los fallos se manifiestan con características muy marcadas que permitan la clasificación.
2. La segunda fase de experimentos se centra en la aplicación los algoritmos de densidad y se realizan diferentes análisis, combinando de varias formas los métodos y algoritmos para conseguir un buen clasificador de fallos. Además, se utilizan coeficientes matemáticos para tratar de discernir la calidad de las agrupaciones de los algoritmos de densidad.

Hay que destacar, que a no ser que se especifique lo contrario, los experimentos se realizan normalizando con media 0 y variancia 1.

4.2 Primera fase experimental, aplicación de los métodos *manifold*

Para la primera fase experimental el objetivo es averiguar si aplicando las transformaciones al conjunto de datos de alta dimensionalidad de la planta, se genera un nuevo conjunto en dos dimensiones que permita distinguir entre los modos de funcionamiento de la planta, es decir, si existe un fallo, cual es y que intensidad tiene. Por ejemplo, en la Figura 25, donde se ven dos series de datos sin fallos y vemos que siguen un patrón similar, si cada modo de funcionamiento generara un patrón distinguible del resto podría lograrse una clasificación.

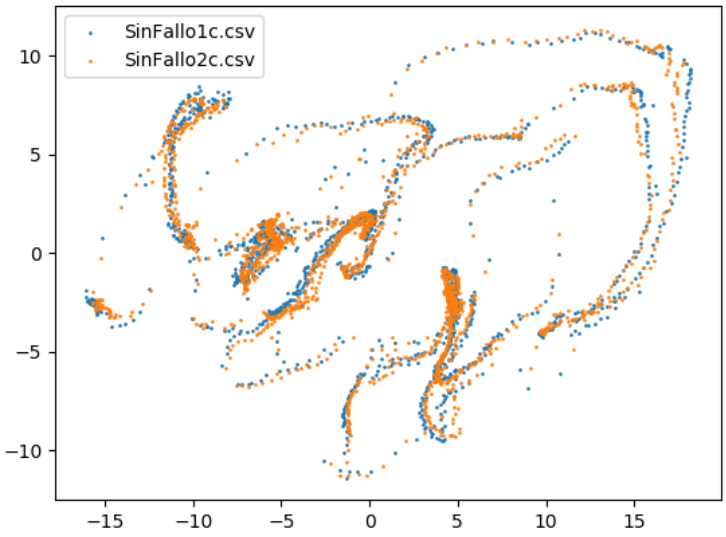


Figura 25 Ejemplo de transformación manifold de dos series de datos sin fallos.

Además, lo ideal sería que cada fallo e intensidad tuvieran un patrón bien definido, pudiendo identificar de forma sencilla que fallo y con qué intensidad ha ocurrido en la planta. Además, se realizan pruebas para intentar lograr una clasificación como la de la Figura 26.

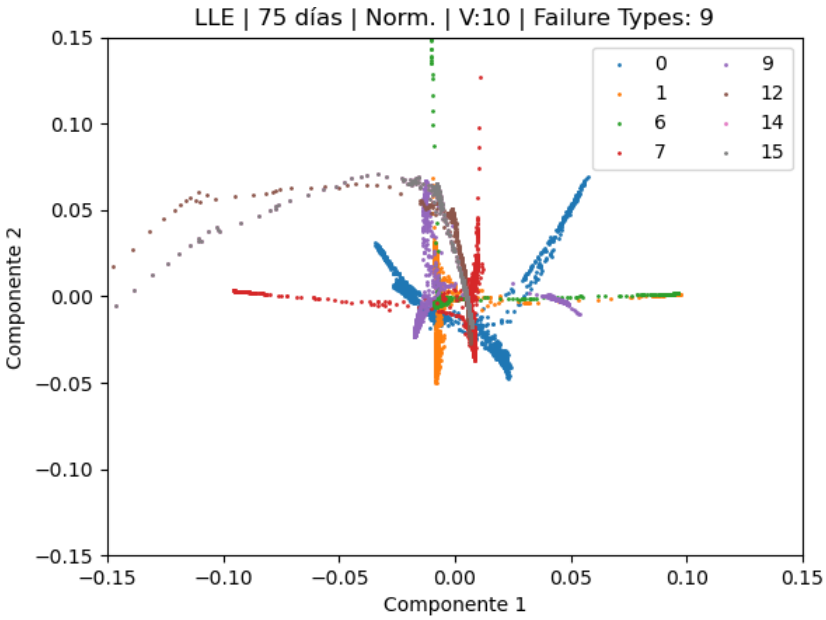


Figura 26 Ejemplo con LLE para cada serie de datos por separado. Cada color indica una serie con diferentes fallos (la serie 0 es la serie sin fallo).

En la primera fase experimental se intenta dar respuesta a los siguientes objetivos:

1. Cuál es el número óptimo de vecinos para aplicar los métodos *manifold* (4.4.1.1).

2. Que evolución temporal tiene el fallo desde que ha ocurrido (4.4.1.2).
3. La posibilidad de identificar un fallo comparándolo con los demás fallos de forma individual (4.4.1.3).
4. Cómo funciona la aplicación del método PCA para eliminar componentes combinada con los métodos *manifold* (4.4.1.4).

4.3 Segunda fase experimental, algoritmos de agrupación basados en densidad

Para la segunda fase de experimentos se aplican los algoritmos de agrupamiento basados en densidad tratando de crear agrupaciones dentro de las transformaciones que permitan algún tipo de clasificación. Además, la combinación de los métodos *manifold* y los algoritmos de densidad permiten obtener una vista intuitiva de cómo se comportan los diferentes fallos y como se realizan las agrupaciones.

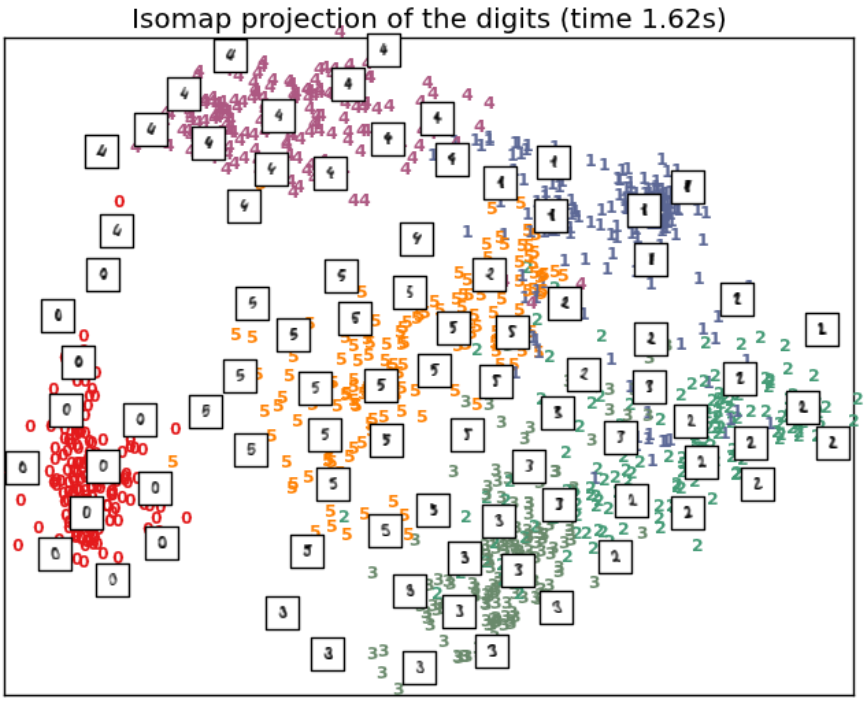


Figura 27 Ejemplo de aplicación ISOMAP para la identificación de números [34].

Aunque la identificación de números de la Figura 27 difiere de la planta que se trata en este trabajo, se intenta extrapolar su resolución y lo ideal sería que cada fallo se pudiera agrupar de la misma forma que hacen los números. De esta manera al tratar de identificar una nueva serie de datos sería automáticamente clasificada en uno de los grupos.

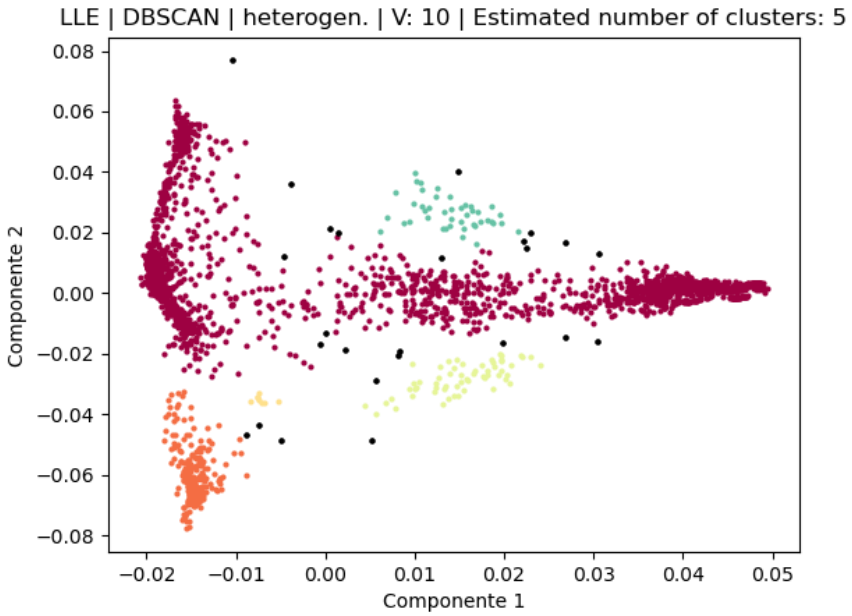


Figura 28 Ejemplo de aplicación de LLE con DBSCAN para los datos de fallo con 10 vecinos.

Una de las ventajas de utilizar algoritmos de *clustering* basados en densidad es que permiten realizar evaluaciones con métodos matemáticos, obteniendo ciertos coeficientes que indican la calidad de las agrupaciones. Además de las observaciones que se puedan hacer examinando los resultados. Combinando ambas formas de análisis se tratará de lograr la clasificación de fallos.

En la segunda fase experimental se realizan los siguientes análisis:

1. Se analizan los fallos de forma individual a través de las agrupaciones creadas por los algoritmos, tratando de lograr clasificarlos.
2. Se analizan todos los fallos concatenados en una misma matriz aplicándole a toda ella tanto los métodos *manifold* como los algoritmos de agrupación.

4.4 Análisis de los datos experimentales

Como se ha expresado en los apartados anteriores, los datos experimentales pertenecen a una planta EDAR, es decir, que se puede prever cierto comportamiento cíclico, tanto a lo largo del año como en las variaciones diarias. El primer pase es realizar un análisis del comportamiento de varias variables. Aun así, en este informe no se incluirán todas las variables analizadas pues estaríamos hablando de un mínimo de 282 figuras que aportarían la misma información.

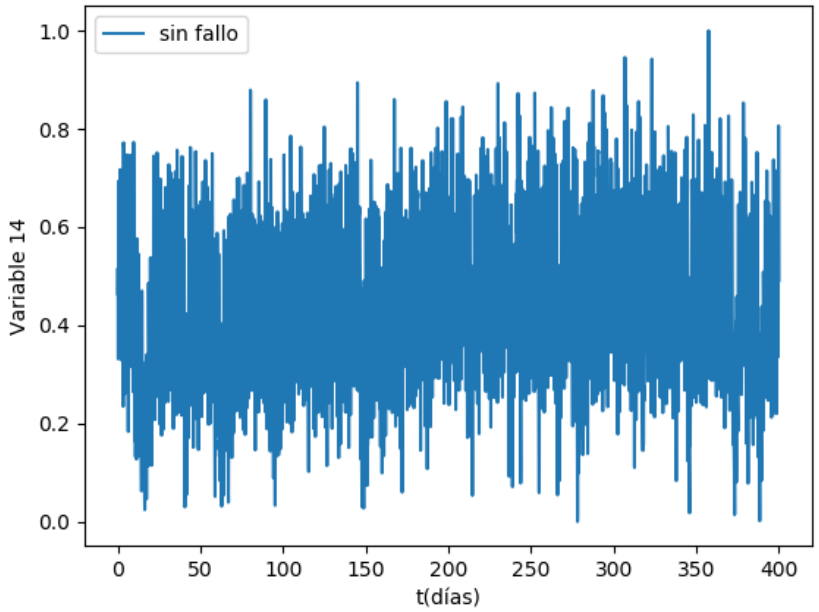


Figura 29 Serie de datos temporal sin fallo.

Aunque a simple vista parezca una serie de datos ruidos si ampliamos podemos apreciar la evolución de la serie (Figura 30).

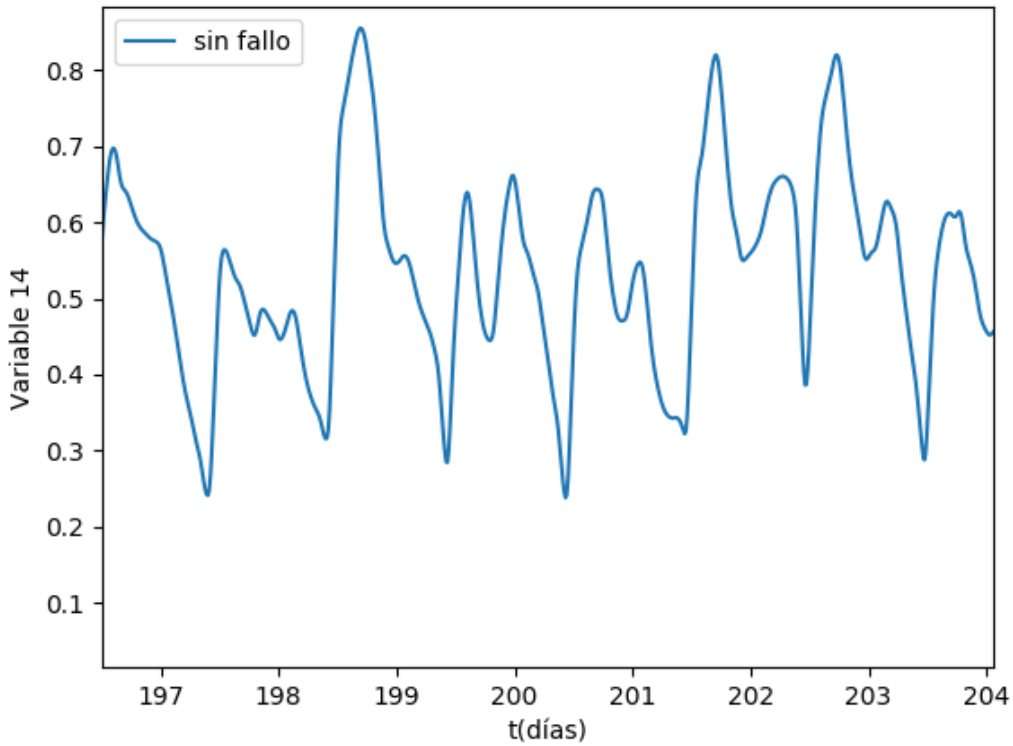


Figura 30 Serie de datos temporal sin fallo ampliada.

Si ahora realizamos una comparación con una serie de datos con fallo con la misma variable (Figura 31), está claro que la variable se ve afectada por el fallo y cambia su comportamiento, y estos cambios serán los que intentemos detectar con nuestro sistema.

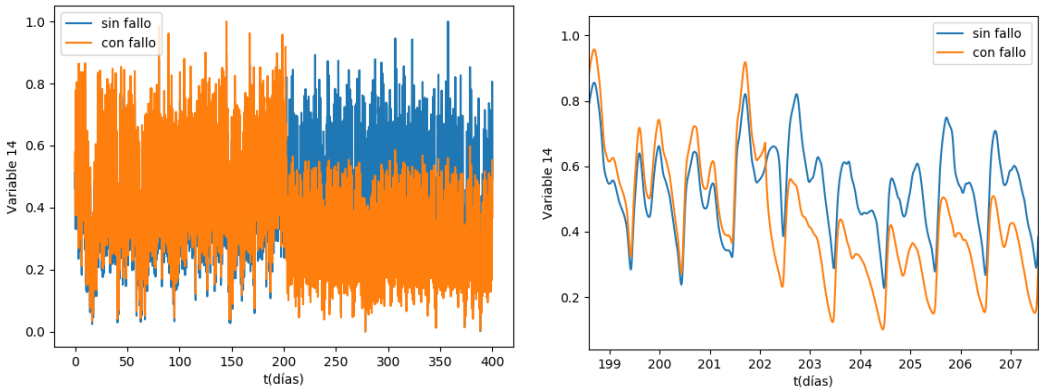


Figura 31 Comparativa sin fallo y con fallo (en el día 200).

Una vez hemos analizado las muestras podemos empezar a aplicar las transformaciones. El método para realizarlas será agrupar las series de datos que tengan el mismo fallo, aunque diferentes intensidades, y de esta forma ver si podemos clasificar las diferentes intensidades. De la misma forma actuaremos, agrupando diferentes fallos para ver si logramos clasificar cada fallo correctamente. Durante las transformaciones variaremos el número de vecinos para intentar encontrar el óptimo que nos dé el mejor resultado, entre otras pruebas que se especifican en los siguientes apartados.

Al utilizar varias series de datos, las muestras que tenemos disponibles se multiplican y nos hacen requerir un tiempo de computación mucho mayor, por ejemplo, si utilizamos el fallo del sensor de oxígeno del 4º reactor, trabajaremos con 5 series de datos, especificando un número de vecinos más bien pequeño como 8 y procesamos 232 días es probable que necesitemos muchas horas de computación y una cantidad elevada de memoria para poder realizar la transformación. La solución pasa por reducir el número de muestras, pero sin llegar a alterar los resultados, a continuación, se muestra una comparación entre diferentes transformaciones y el tiempo de computación.

- 1 muestra por cada 4, $t_{computación}$= 4661.62s = 1h 18minutos.
- 1 muestra de cada 5, $t_{computación}$= 1165.95s = 19,43minutos.
- 1 muestra de cada 6, $t_{computación}$= 651.12s = 10,85minutos.
- 1 muestra de cada 10 $t_{computación}$= 5.96minutos
- 1 muestra de cada 16, $t_{computación}$=41,50s
- 1 muestra de cada 32, $t_{computación}$=7.61s

Como puede observarse, el coste computacional aumenta drásticamente mientras más muestras analizamos, para simplificar el proceso realizaremos todas las

transformaciones con el mismo muestreo de una muestra cada 16, es decir, una muestra cada 4 horas.

4.4.1 Aplicación de los métodos *manifold*

4.4.1.1 Análisis según el número de vecinos escogidos

4.4.1.1.1 Entrenamiento aplicado a su propio fallo

En este apartado se estudia cómo afecta la elección del número de vecinos a las transformaciones. Recordemos como se ha comentado en el apartado 2.3.2 que el número óptimo suele encontrarse entre 8 y 12. El procedimiento utilizado se muestra en el Algoritmo 1, y en la Figura 32 donde se representa el diagrama de flujo utilizado.

tipoTransformacion = "LLE" o "ISOMAP"

Para *n_neighbors* 8,10,12,20,50,100 Hacer

Para *tipoFallo* con cada *intensidadFallo*

X = 1 de cada 16 muestras a partir de la muestra 19200

Normalizar *X* con media 0 y varianza 1

Si *tipoTransformacion LLE* Entonces

Entrenar Transformación LLE con X

Aplicar Transformación LLE a X

Sino Entonces

Entrenar Transformación ISOMAP con X

Aplicar Transformación ISOMAP a X

Fin Para

Realizar scatter Plot

Fin Para

Algoritmo 1 Algoritmo para calcular el número de vecinos a utilizar.

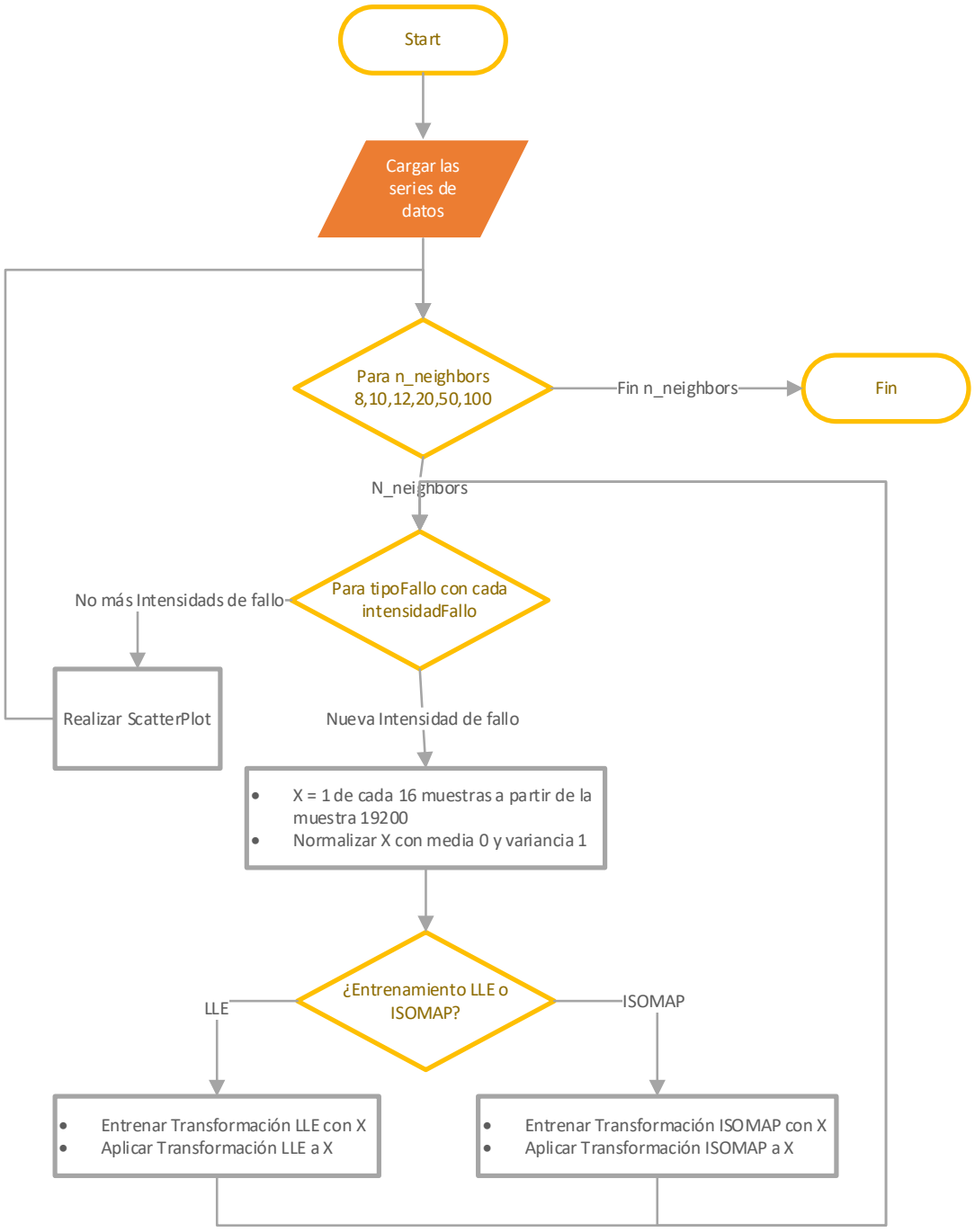


Figura 32 Proceso para comparar número de vecinos.

Considerando el número de fallos, el número de vecinos calculados y las dos transformaciones, se tiene un elevado número de gráficas, que es imposible poner en esta memoria. Como resumen se muestra un ejemplo para la transformación LEE para los datos de comportamiento normal con 8 vecinos y con 12 vecinos, en la Figura 33 y la Figura 34, y para la transformación ISOMAP con datos de los fallos 9, 10 y 11 para 10 vecinos y para 50 vecinos, en la Figura 35 y la Figura 36.

Realizando el análisis de estas figuras se obtienen que los resultados son muy parecidos usando diferentes números de vecinos, en particular para la transformada ISOMAP, el resultado es prácticamente igual usando 10 que 50 vecinos, por lo que usar más vecinos simplemente aumenta la carga computacional, y no parece que los resultados sean mejores. Se harán más experimentos, pero esto nos permite considerar que un número de 10 vecinos es el más adecuado para estos experimentos.

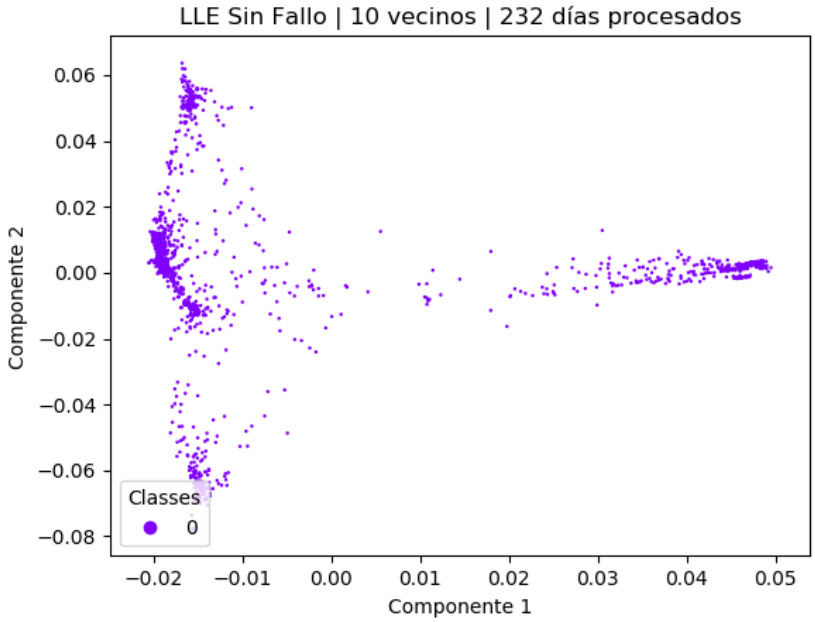


Figura 33 Resultado para comparar el número de vecinos con 10 vecinos.

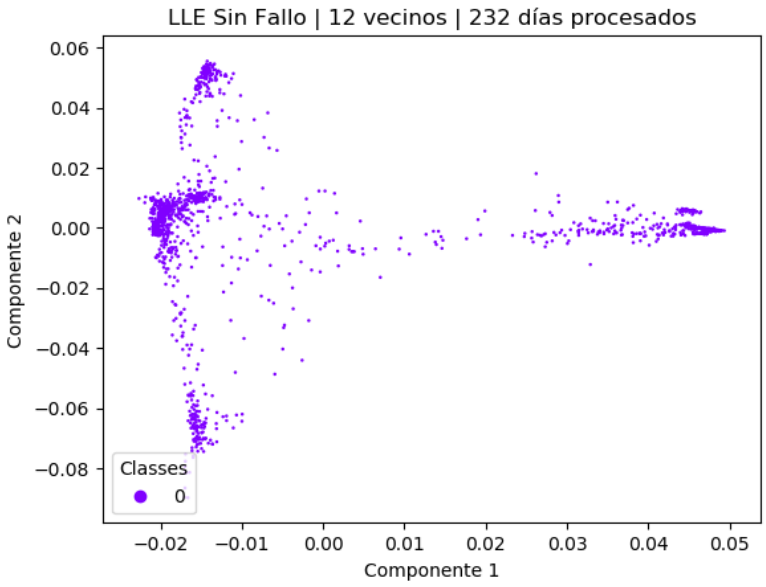


Figura 34 Resultado para comparar el número de vecinos con 12 vecinos.

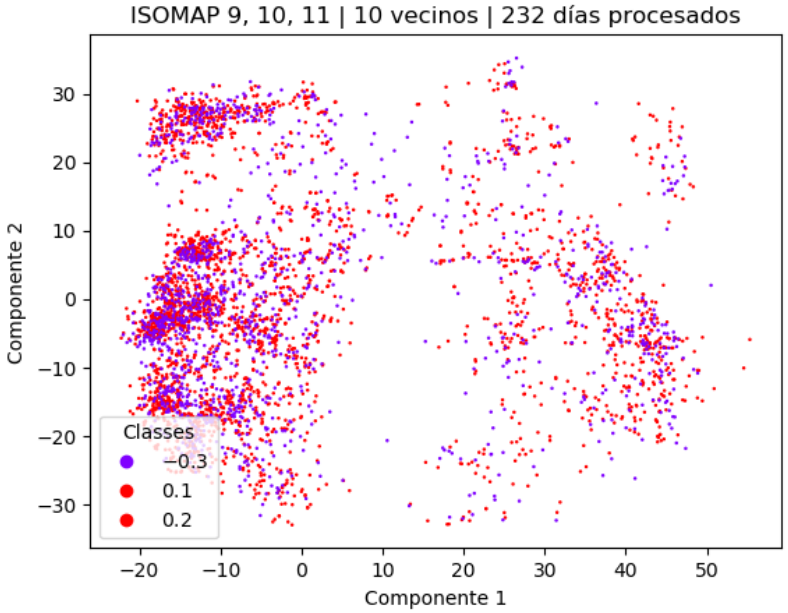


Figura 35 Resultado para comparar el número de vecinos para la serie sin fallo con 10 vecinos ISOMAP.

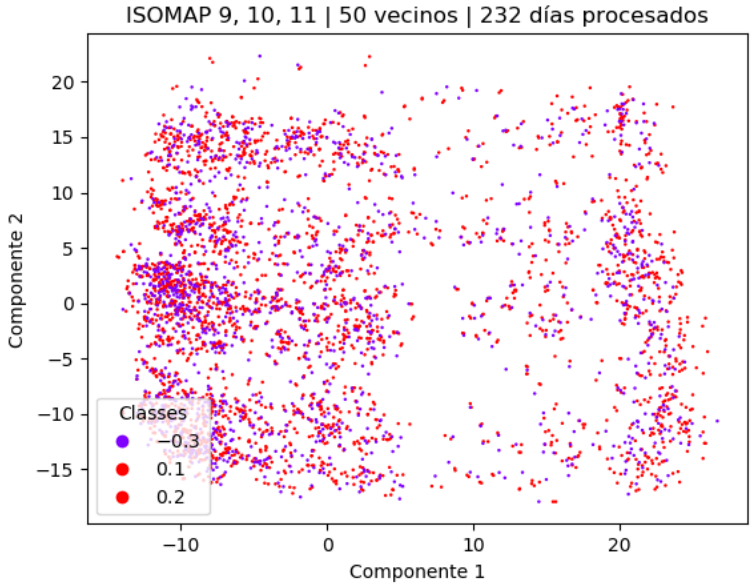


Figura 36 Resultado para comparar el número de vecinos para la serie sin fallo con 50 vecinos ISOMAP.

4.4.1.1.2 Entrenamiento utilizando la serie sin fallos como entrenamiento para la comparación del número de vecinos

Es necesario recalcar que al realizar una reducción de dimensionalidad o transformación se pueden utilizar las transformaciones de una serie de datos y guardarla para aplicar la transformación sobre otra serie de datos. De esta forma se puede realizar una transformación con la serie que deseemos y aplicar la configuración del método escogido al resto de series. El procedimiento se explica tanto en el Algoritmo 2, como en

el diagrama de flujo correspondiente de la Figura 37. Este experimento nos puede servir, para ver si procesando todos los datos de fallos por una referencia común (la transformación de los datos sin fallo, para ambas transformaciones LLE e ISOMAP), los datos se comportan de forma diferente y se pueden distinguir entre ellos y sobre todo si se pueden diferenciar del comportamiento normal.

tipoTransformacion = “LLE” o “ISOMAP”

S = 1 de cada 16 muestras de los datos sin fallo a partir de la muestra 19200

Normalizar *S* con media 0 y varianza 1

Para *n_neighbors* 8,10,12,20,50,100 Hacer

Para *tipoFallo* con cada *intensidadFallo*

X = 1 de cada 16 muestras a partir de la muestra 19200 de cada intensidadFallo

Normalizar *X* con media 0 y varianza 1

Si *tipoTransformacion LLE* Entonces

Entrenar Transformación LLE con S

Aplicar Transformación LLE a X

Sino Entonces

Entrenar Transformación ISOMAP con S

Aplicar Transformación ISOMAP a X

Fin Para

Realizar scatter Plot

Fin Para

Algoritmo 2 Algoritmo para la aplicación de la reducción de dimensionalidad a partir de otra transformación para comparar el número de vecinos.

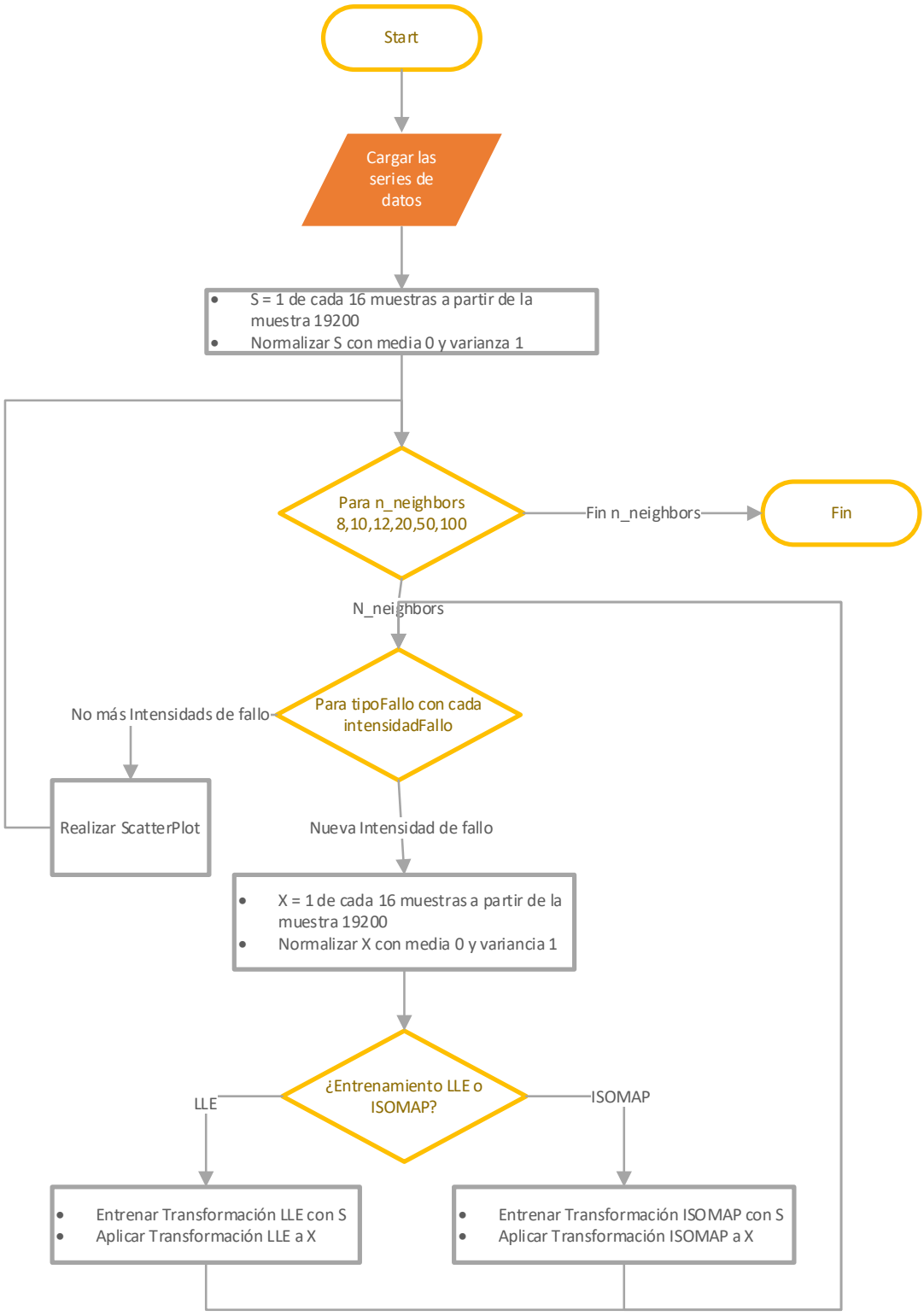


Figura 37 Proceso de la reducción de dimensionalidad con la opción de partir de otra transformación para comparar el número de vecinos.

En general, este experimento muestra que el comportamiento de los datos de fallo procesados por la transformación de comportamiento normal, cambia bastante su forma y sobre todo la densidad de puntos que se hace mucho más densa y uniforme en comparación con el comportamiento de los datos con su propia transformación,

independientemente del número de vecinos, como muestra en la Figura 38, donde se representa el resultado comparativo cuando los datos de fallo 15 y 16 se pasan por la transformación (tanto LSE como ISOMAP) entrenada con datos de comportamiento normal (gráficas de la izquierda) y cuando se pasan por su propia transformación (gráficas de la derecha).

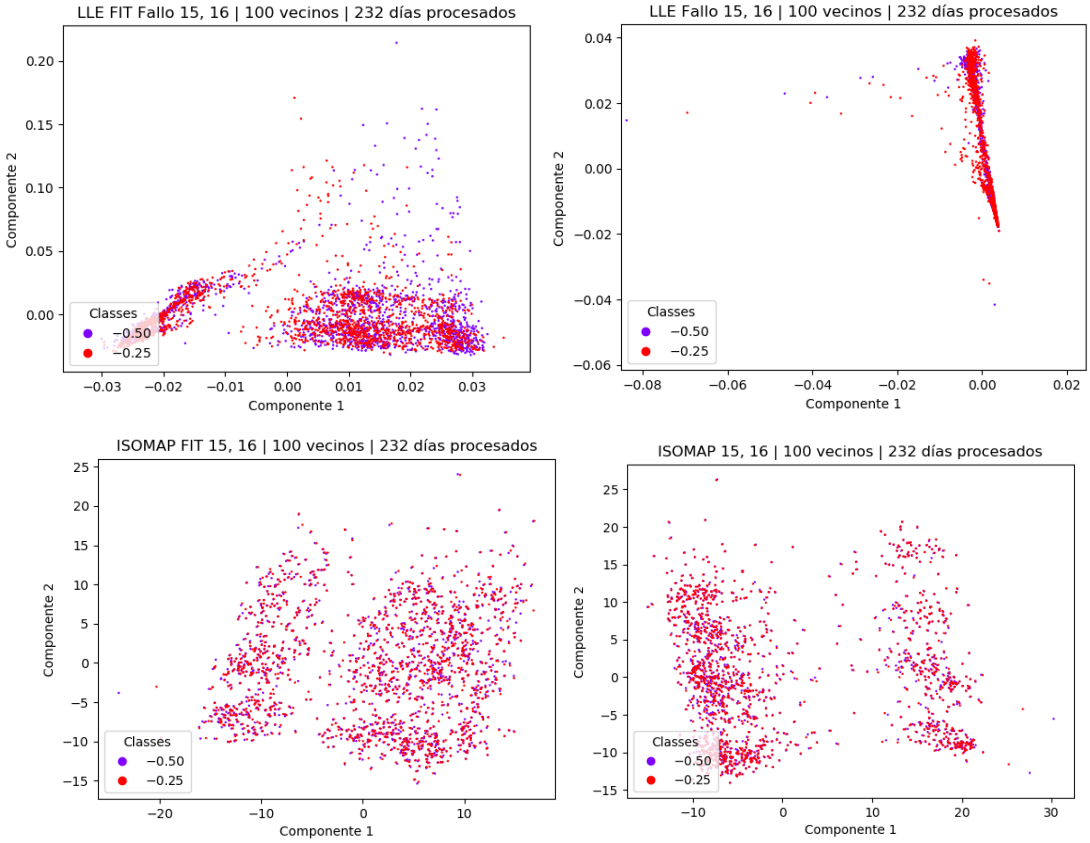


Figura 38 Transformación LLE (arriba) e ISOMAP (abajo) de los fallos 15 y 16 aplicándoles una transformación de la serie sin fallos con 100 vecinos (izquierda) y su propia transformación (derecha).

Por otro lado, si queremos comparar estas transformaciones en función del número de vecinos usados para realizarlas, este resultado se puede ver en la Figura 39, donde se ve que nuevamente para ISOMAP el número de vecinos no es muy significativo, ya que los resultados de ambas trasformaciones son muy similares., aunque es menos densa con 10 vecinos que con 100. Para la transformación LLE si cambia la forma como en el caso anterior.

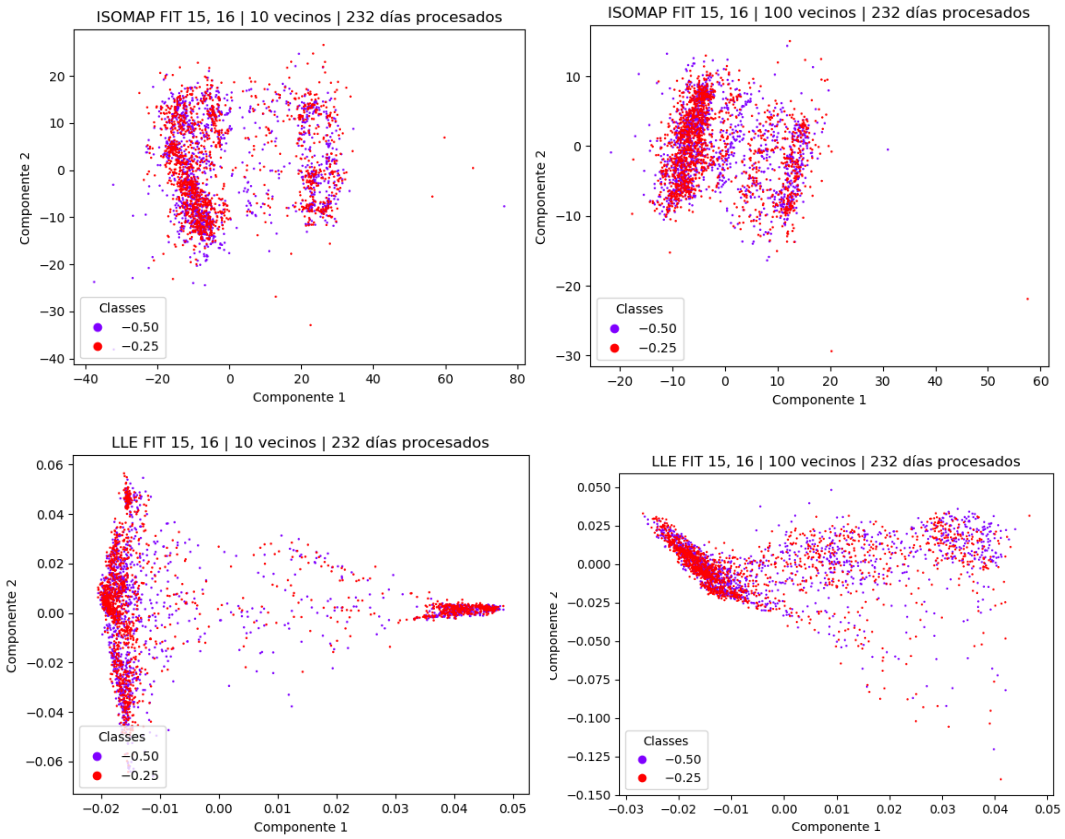


Figura 39 Comparación entre 10 y 100 vecinos para ISOMAP y LLE, aplicando la transformación de la serie sin fallo para los fallos 15 y 16.

Como uno de los objetivos principales, es la de poder clasificar los diferentes tipos de fallo, es necesario comparar, ya no únicamente entre las diferentes intensidades de fallo, sino también entre los diferentes tipos de fallos. Para ello se va a mostrar en una única gráfica, las transformaciones *LLE* e ISOMAP para cada tipo de fallo, las leyendas de las figuras indican el fallo escogido en referencia a la lista de fallos de los datos escogidos, mientras que la serie *0* indica la serie que no tiene fallos. Para realizar esta comparación se entrenará el método con la serie sin fallos y se aplicará la transformación a cada tipo de fallos.

Se puede apreciar en la Figura 40 y en la Figura 41 que las transformaciones mantienen la misma forma o silueta mientras que la densidad de los puntos cambia según el tipo de fallo. Cabe destacar que se obtienen densidades más uniformes cuando se utilizan 10 vecinos que cuando se utilizan 100. Además, se puede apreciar en la Figura 41 claramente como la serie de datos sin fallos (0 en la leyenda) tiene baja densidad de puntos en la zona central de la representación. Mientras que las series de datos con fallo tienen un comportamiento variado. Con estas observaciones es lógico pensar que uno de los pasos a seguir sea aplicar algún algoritmo que analice estas densidades con el objetivo de crear un clasificador, es aquí donde entran los algoritmos de agrupamiento basados en densidad de la segunda fase de pruebas.

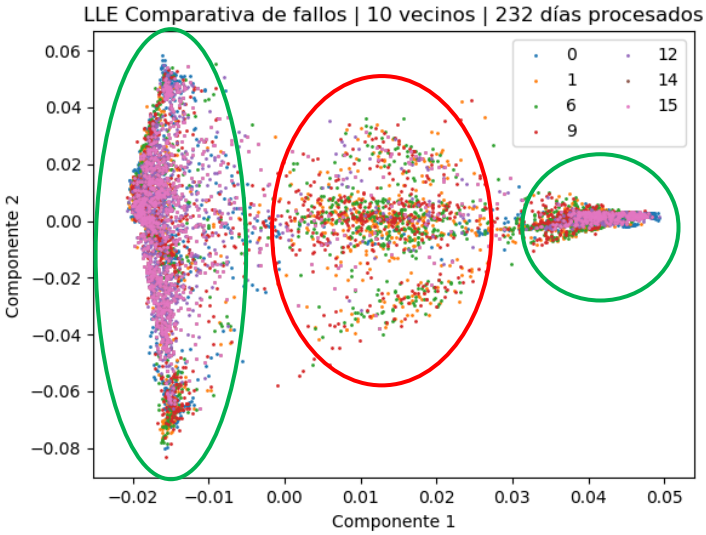


Figura 40 Comparativa de fallos para determinar el número óptimo de vecinos con 10 vecinos.

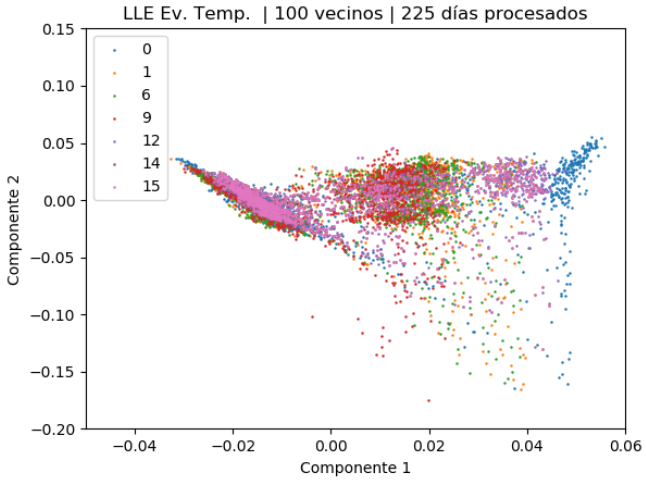


Figura 41 Comparativa de fallos para determinar el número óptimo de vecinos con 100 vecinos.

Como resultado de las pruebas realizadas para obtener un número de vecinos que sea válido para el resto de experimentos a realizar, se ha decidido escoger 10 vecinos como parámetro. Dado que al aplicar las transformaciones sin fallo sobre los datos con fallo se han obtenido densidades más uniformes las cuales deberían ofrecer mejores resultados en los siguientes experimentos.

4.4.1.2 *Evolución temporal*

El siguiente experimento trata de ver la evolución temporal de cada tipo de fallo para comparar cómo evoluciona junto a la serie sin fallo. A continuación, se muestra cada tipo de fallo con una única intensidad y la serie de datos sin fallo (*clase 0*), primero para 25 días procesados, y después añadiendo a la serie de datos 25 días más hasta llegar a 225 días procesados. De esta manera se puede apreciar la evolución acumulada que provoca el fallo y como se ha comentado en el apartado anterior ver si cambia la densidad de los puntos para poder crear alguna clasificación utilizando algoritmos de densidad. En este caso se estudiarán las dos posibilidades, haciendo las transformaciones LLE e ISOMAP para cada fallo por separado, y como se explicó en el apartado anterior, entrenando con los datos de comportamiento normal y aplicando después esta transformada a los datos de los distintos fallos. El procedimiento está explicado en el Algoritmo 3 y en la Figura 42 que muestra el diagrama de flujo del procedimiento seguido.

tipoTransformacion = “LLE” o “ISOMAP”

tipoEntrenamiento = “SinFallo” o “*”

S = 1 de cada 16 muestras de los datos sin fallo a partir de la muestra 19200

Normalizar *S* con media 0 y varianza 1

Para *días* 25 hasta 225 Con Paso 25 Hacer

Para cada *tipoFallo* con 1 de la *intensidadFallo*

X = 1 de cada 16 muestras a partir de la muestra 19200 de cada intensidadFallo

Normalizar *X* con media 0 y varianza 1

Si *tipoTransformacion LLE* Entonces

Si *tipoEntrenamiento sinFallo* Entonces

Entrenar Transformación LLE con S

Aplicar Transformación LLE a X

Sino Entonces

Entrenar Transformación LLE con X

Aplicar Transformación LLE a X

Sino Entonces

Si *tipoEntrenamiento sinFallo* Entonces

Entrenar Transformación ISOMAP con S

Aplicar Transformación ISOMAP a X

Sino Entonces

Entrenar Transformación ISOMAP con X

Aplicar Transformación ISOMAP a X

Fin Para

Realizar scatter Plot

Fin Para

Algoritmo 3 Algoritmo para estudiar la evolución temporal.

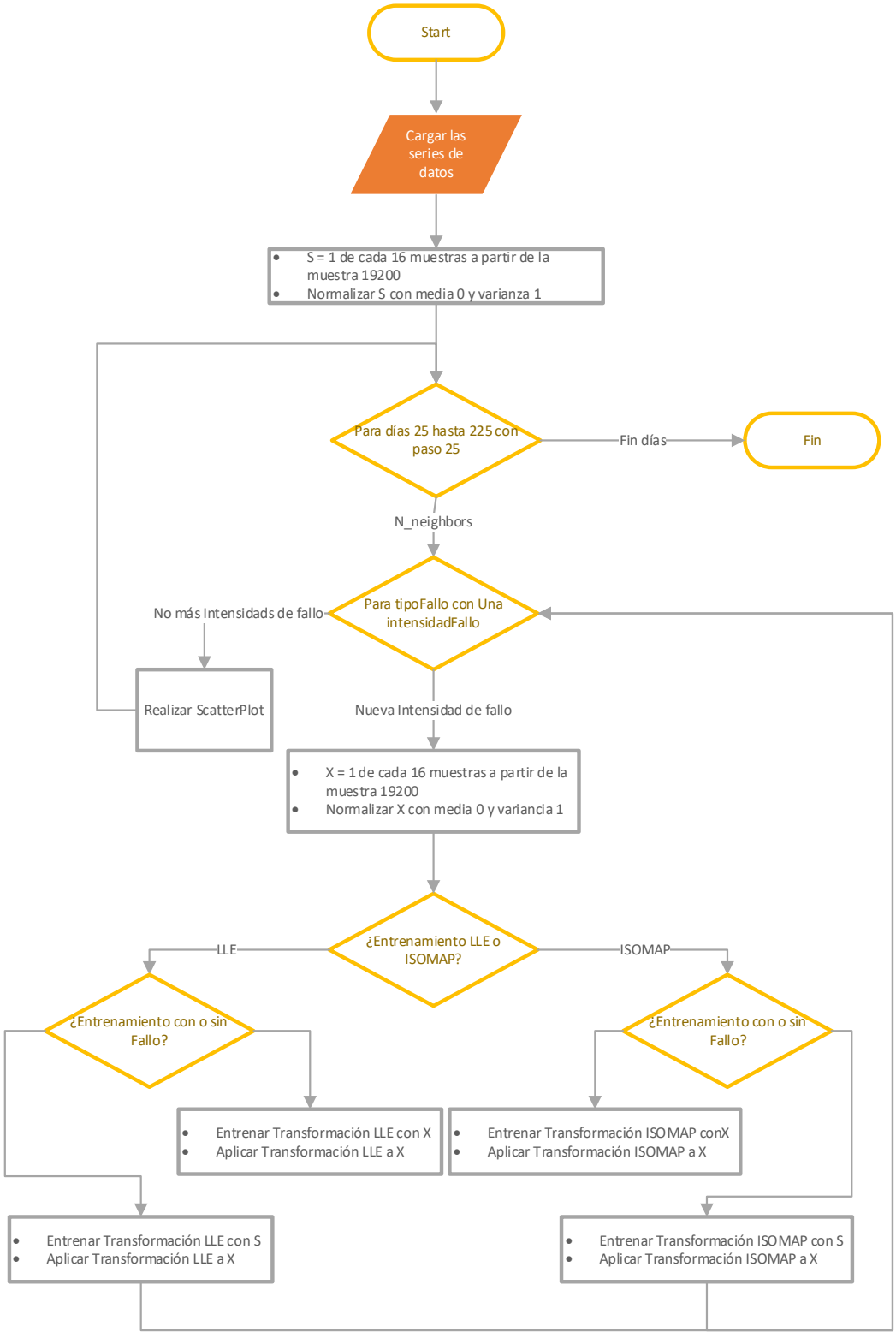
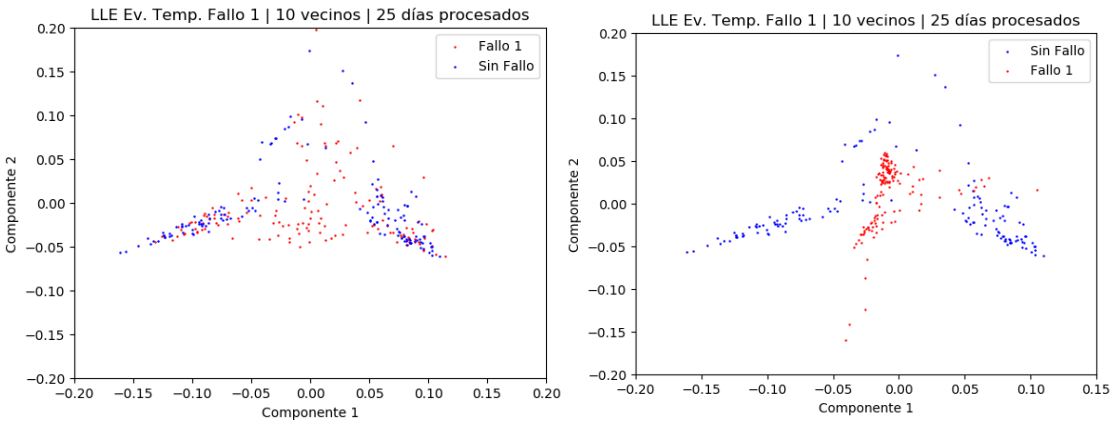


Figura 42 Proceso para estudiar la evolución temporal.

Como se ha dicho para este experimento se han probado tanto LLE como ISOMAP y se han realizado las transformaciones cada 25 días y todos los fallos. Aunque en los resultados se muestra únicamente el fallo en el sensor de oxígeno del 4º reactor (Fallo 1) y cada 50 días. Se ha decido mostrarlo de esta manera para no alargar el documento innecesariamente, dado que los demás fallos convergían a la misma conclusión. Como resultado, se puede apreciar en la Figura 43 en la columna de la izquierda la evolución del fallo 1 y que cuando se le aplica la transformación de la serie de datos sin fallo se vuelve difícil distinguir el fallo en comparación a cuando no existe fallo. En cambio, cuando se aplica al fallo su propia transformación (columna derecha) si se puede observar cómo existe diferencia entre la distribución de ambos grupos. Por lo tanto, se puede presuponer que aplicar una transformación a un fallo distinto llevara a que quede demasiado influido por está y hará imposible que sea reconocible.

Teniendo en cuenta esto y la evolución temporal se ha optado por realizar pruebas escogiendo el número máximo de muestras (225 días) que es cuando más información se tiene para realizar las transformaciones. Además, se ha escogido realizar parte de los experimentos eligiendo el número de muestras cuando se manifiesta más diferencia entre los fallos (75 días), debido a que los resultados parecen indicar que si se escoge un número muy elevado de días el fallo se va diluyendo a causa de que el sistema intenta compensarlo. Por último, en la Figura 44 se da una muestra de 75 y 225 días siguiendo la lógica de la Figura 43 donde se puede ver que con ISOMAP no existe una distinción tan clara como con LLE.



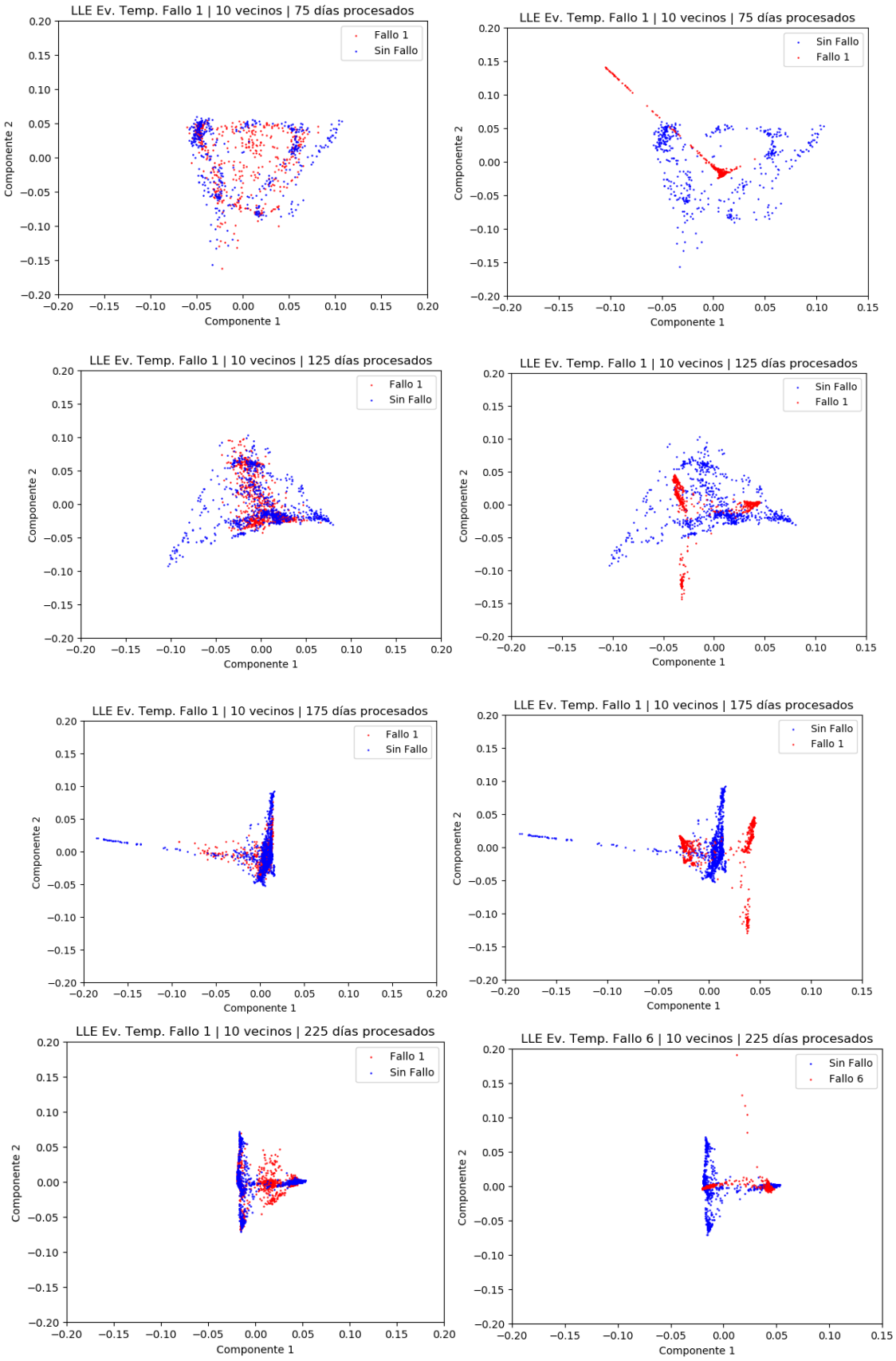


Figura 43 Las gráficas se presentan la evolución temporal cada 50 días del fallo 1 con transformación LLE. En la columna izquierda representan la transformación del fallo aplicándole la transformación sin fallo. Mientras que en la columna de la derecha se aplica a cada serie su propia transformación.

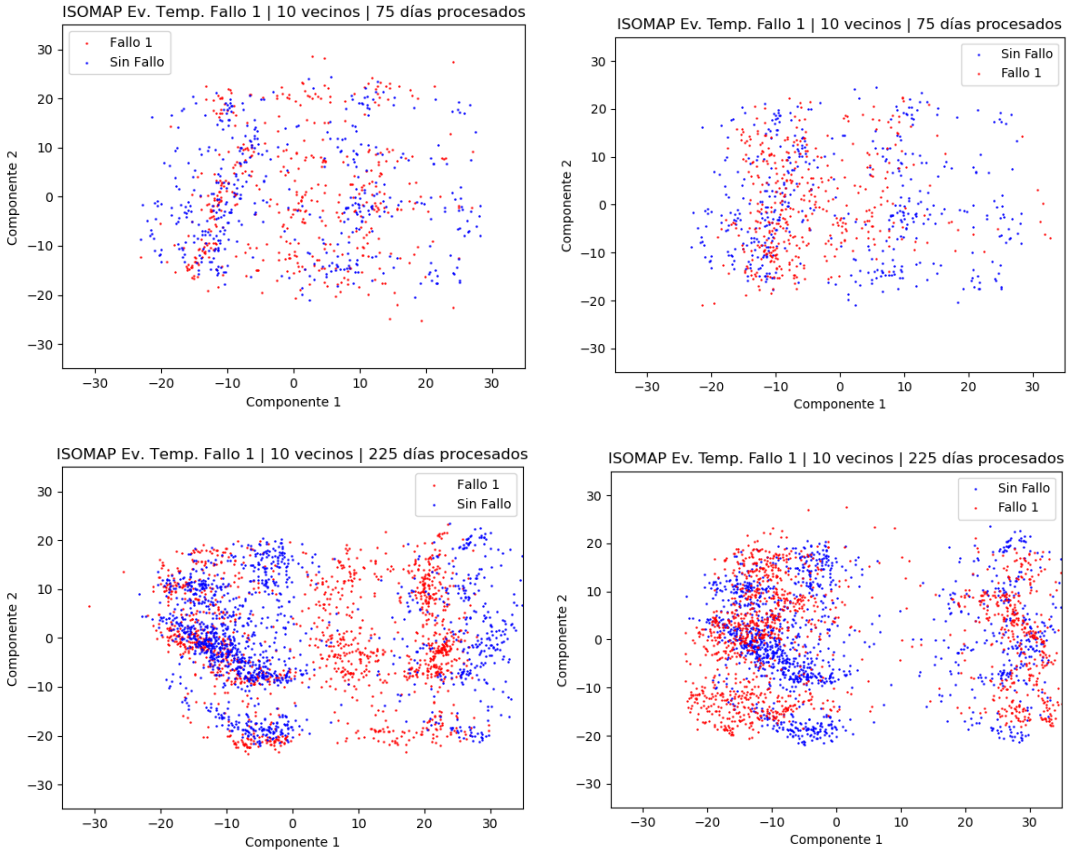


Figura 44 Las gráficas se presentan la evolución del fallo 1 con transformación ISOMAP para 75 y 225 días. En la columna izquierda representan la transformación del fallo aplicándole la transformación sin fallo. Mientras que en la columna de la derecha se aplica.

4.4.1.3 Comparación de los fallos de forma individual

Teniendo en cuenta las conclusiones de la sección anterior (4.4.1.2) es necesario comprobar si se puede lograr una clasificación de los fallos aplicándoles su propia transformación y comparándolos entre sí. De igual manera hay que tener presente las conclusiones y escoger el número de datos en el que mayor diferencia manifiesten los fallos que es 75 días.

Por lo tanto, se realizan las normalizaciones y las transformaciones para cada fallo por separado, se escoge un fallo de cada tipo para poder utilizarlos como base de datos. A continuación, se le aplican al fallo desconocido las transformaciones de la base de datos y se compara uno a uno para tratar de identificarlo. El proceso puede verse más detallado en el Algoritmo 4 y la Figura 45.

S = 1 de cada 16 muestras de los datos del fallo a comparar a partir de la muestra 19200

Normalizar *S* con media 0 y varianza 1

Entrenar Transformación ISOMAP con S

Aplicar Transformación ISOMAP a S

Para cada *tipoFallo* con 1 de la *intensidadFallo*

X = 1 de cada 16 muestras a partir de la muestra 19200 de cada intensidadFallo

Normalizar *X* con media 0 y varianza 1

Entrenar Transformación ISOMAP con X

Aplicar Transformación ISOMAP a X

Realizar Scatter Plot

Fin Para

Algoritmo 4 Algoritmo para la comparación de los fallos de forma individual con 75 días e ISOMAP.

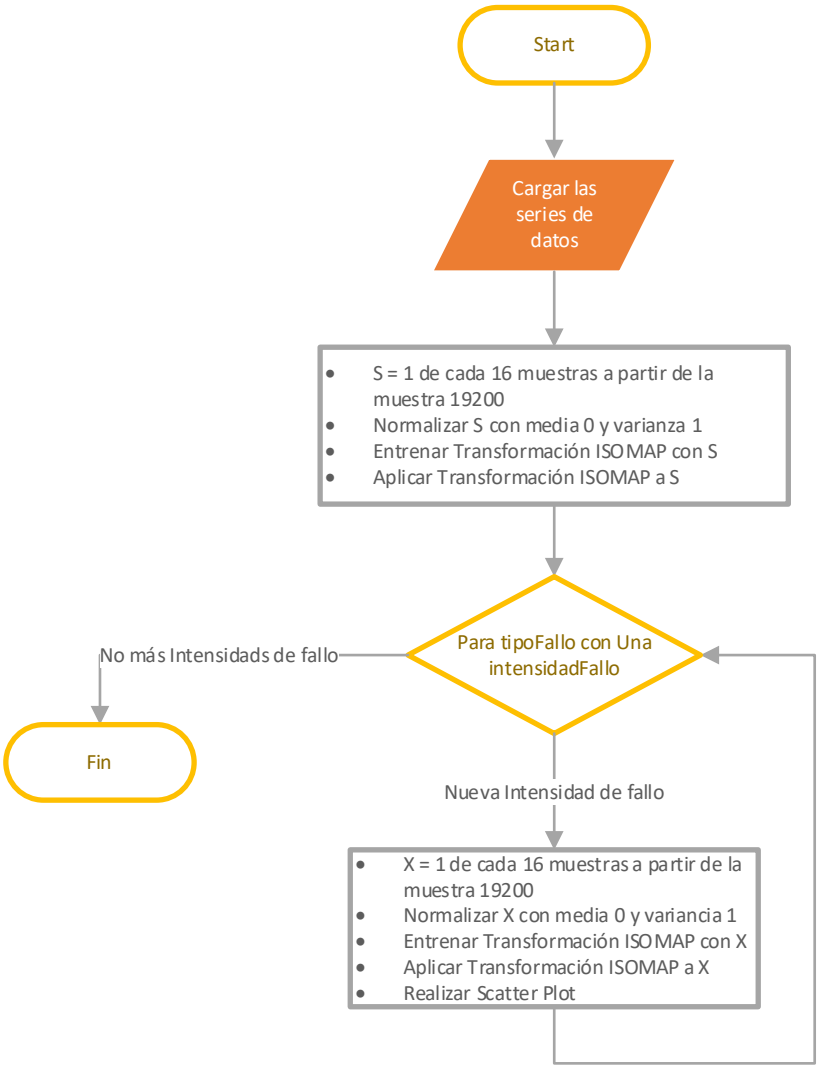


Figura 45 Proceso para la comparación de los fallos de forma individual con 75 días e ISOMAP.

Primero, se muestra en la Figura 46 el resultado de aplicar las transformaciones LEE (izquierda) e ISOMAP (derecha) a cada tipo de fallo distinto por separado usando sólo 75 días de datos y superponiendo el resultado para tener una visión global de las representaciones.

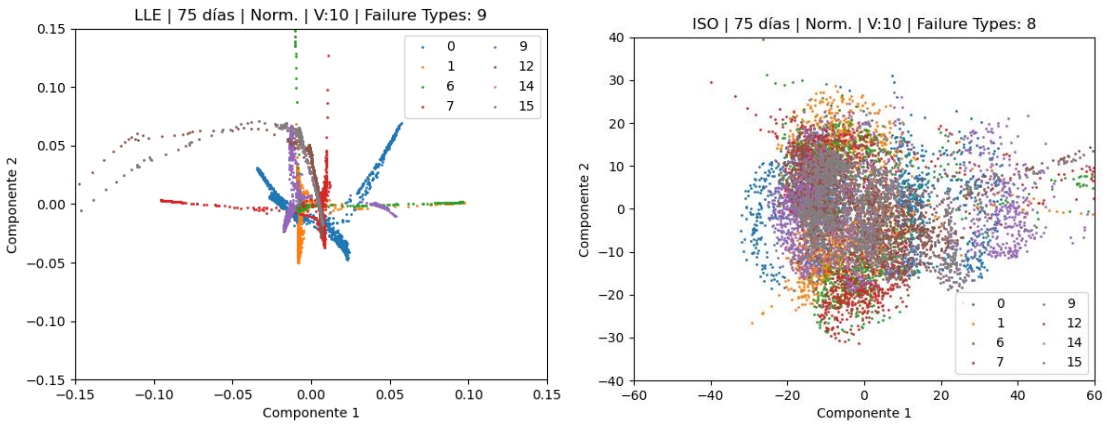


Figura 46 Representación de las transformaciones individuales de los fallos para 75 días para LLE (izquierda) e ISOMAP (derecha).

Puede apreciarse en la representación LLE como cada fallo tiene una silueta determinada, aunque es necesario subrayar que algunos de ellos son muy similares si se les aplica simetría, como por ejemplo el fallo 1 y 7 que si se les aplica simetría vertical son prácticamente idénticos. No obstante, parece ser que la transformación ISOMAP no es capaz de realizar ninguna distinción entre los fallos.

Ahora para ver visualmente si se puede identificar cada tipo de fallo, se muestra el resultado de pasar el fallo 3 (como fallo desconocido) sobre las transformaciones LLE de cada tipo de fallo, para ver a qué forma se parece más, para poder ser identificado. Los resultados se muestran en la Figura 47, donde queda claro que es imposible clasificar el Fallo 3 como un único y diferente fallo. No obstante, si existe una cierta clasificación clara dado que el Fallo 3 coincide con los fallos del 1 al 11 y no lo hace con los fallos del 12 al 16. Para confirmar esta hipótesis se ha repetido el proceso para el fallo 8 (Figura 48) comparando este fallo con el fallo 1 y el 12 para ver si entra dentro del mismo grupo (del 1 al 11) y para el fallo 13 (Figura 49) comparando con los mismos fallos, para ver si pertenece al otro grupo (12-16).

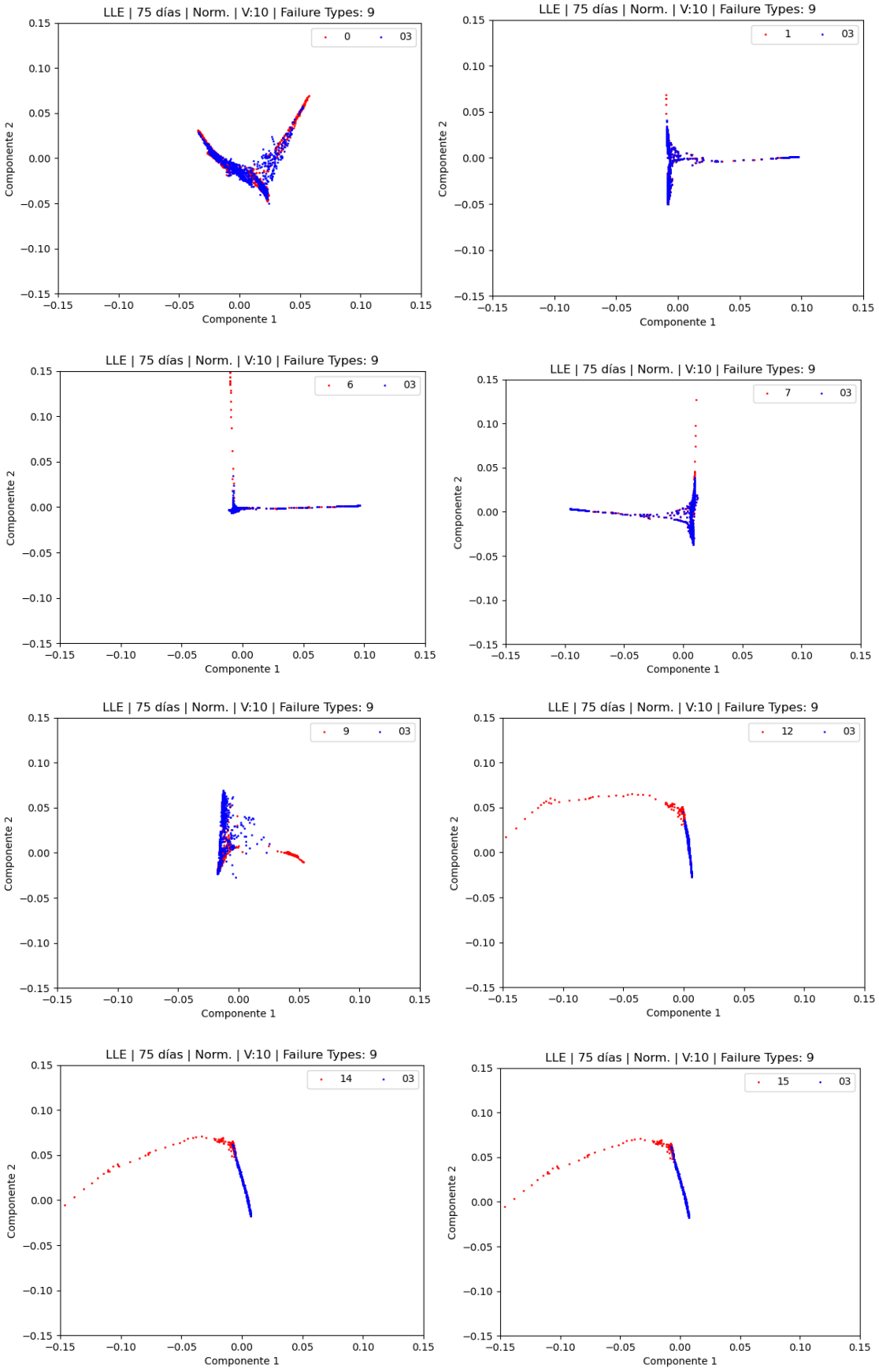


Figura 47 Comparación del Fallo 3 con los tipos de fallos con LLE y 75 días de datos.

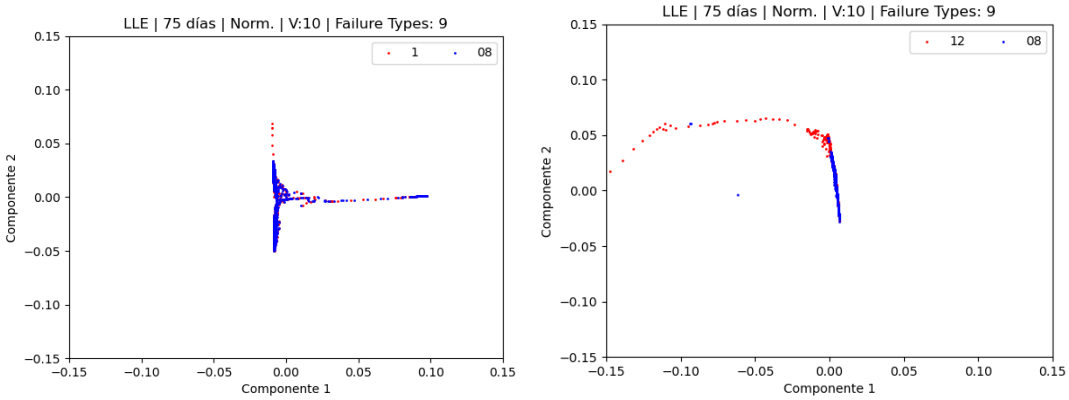


Figura 48 Comparativa del fallo 8 con los fallos 1 y 12 respectivamente con LLE y 75 días de datos.

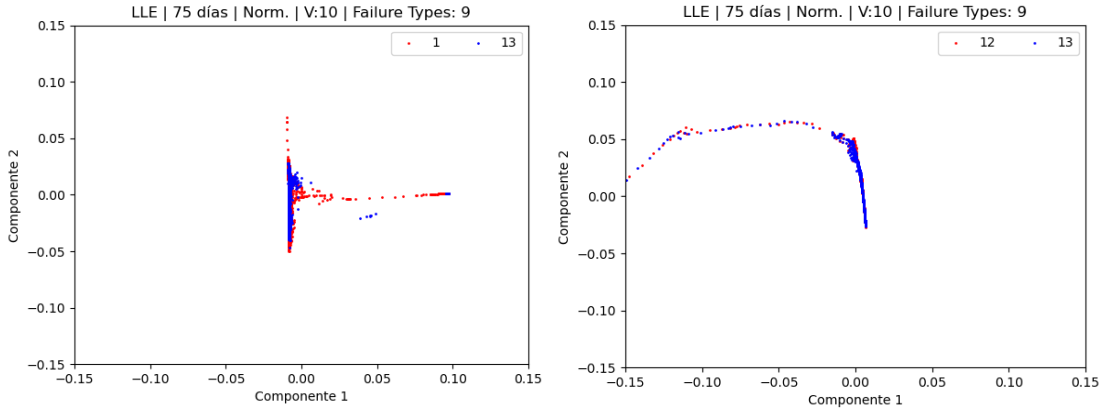


Figura 49 Comparativa del fallo 13 con el fallo 1 y 12 respectivamente con LLE y 75 días de datos.

Al estudiar la Figura 48 y la Figura 49 observamos que para el fallo 8 ocurre lo mismo que para el fallo 1 de la Figura 47 mientras que para el 13 coincide con todos los tipos de fallo. Teniendo en cuenta estas afirmaciones podría decirse que se puede realizar una identificación parcial de los fallos, es decir puede saberse si un fallo pertenece a los fallos del 1 al 11 o si pertenece a los fallos del 12 al 16. No obstante, hay que tener en cuenta que este método solo puede clasificar el fallo, no detectarlo, dado que todos los fallos han coincidido también con la serie de datos sin fallo.

4.4.1.4 Aplicación del método PCA para eliminar componentes

A continuación, utilizando el método PCA se puede averiguar cuantas de las variables tienen un peso real en el sistema, permitiendo eliminar el resto y obtener unos resultados próximos a los que se obtendrían utilizando todas las variables.

Tras realizar el cálculo de eliminación de variables, resulta que para todas las series de datos por separado (Figura 50) se obtiene un comportamiento similar. Por lo tanto, utilizando 20 variables en vez de las 141 que componen el sistema se obtendría

prácticamente la misma precisión, reduciendo el coste computacional y facilitando la comprensión de los datos, cabe destacar que se llega al 90% de variancia utilizando 6 variables aproximadamente.

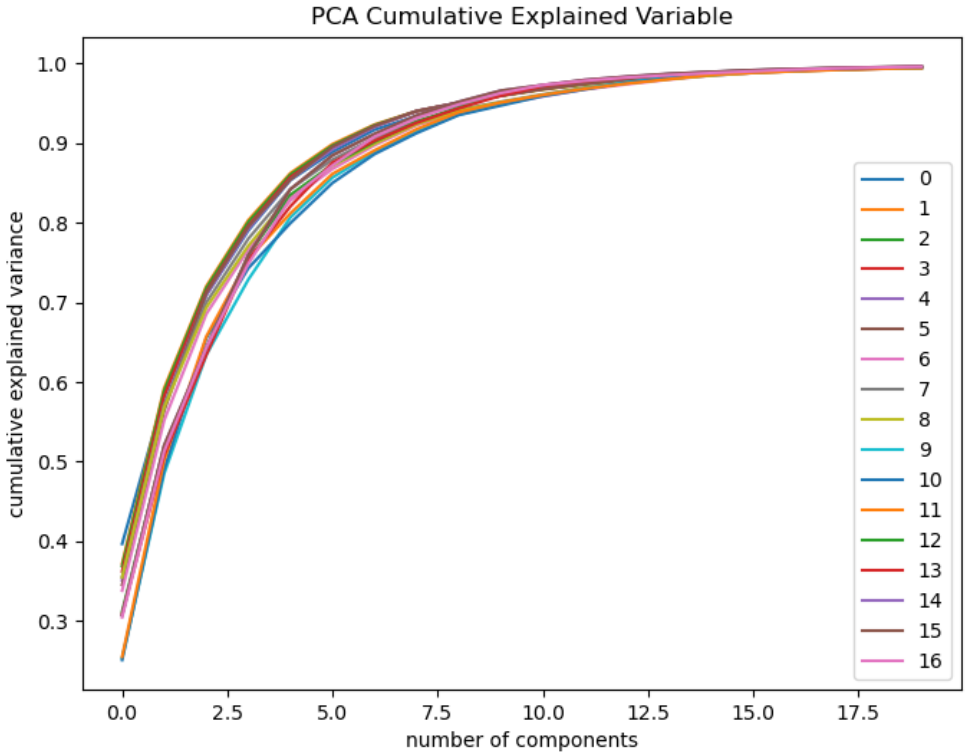


Figura 50 Resultado de la eliminación de variables con PCA.

Para comprobar el correcto funcionamiento del método PCA se va a repetir el experimento del apartado anterior, pero utilizando 20 variables, para asegurarnos de cubrir prácticamente el 100% de la variabilidad del sistema. Si se mantienen los anteriores resultados se podría aplicar sin ningún problema y utilizar este método. El procedimiento seguido ahora se muestra en el Algoritmo 5 y en el diagrama de flujo de la Figura 51.

tipoTransformacion = "LLE" o "ISOMAP"

tipoEntrenamiento = "SinFallo" o "cada uno el suyo"

S = 1 de cada 16 muestras de los datos sin fallo a partir de la muestra 19200

Normalizar *S* con media 0 y varianza 1

Para *días* 25 hasta 225 Con Paso 25 Hacer

Para cada *tipoFallo* con 1 de la *intensidadFallo*

X = 1 de cada 16 muestras a partir de la muestra 19200 de cada intensidadFallo

Normalizar *X* con media 0 y varianza 1

Si *tipoTransformacion LLE* Entonces

Si *tipoEntrenamiento sinFallo* Entonces

Aplicar PCA a S

Entrenar Transformación LLE con S

Aplicar Transformación LLE a X

Sino Entonces

Aplicar PCA a X

Entrenar Transformación LLE con X

Aplicar Transformación LLE a X

Sino Entonces

Si *tipoEntrenamiento sinFallo* Entonces

Aplicar PCA a S

Entrenar Transformación ISOMAP con S

Aplicar Transformación ISOMAP a X

Sino Entonces

Aplicar PCA a S

Entrenar Transformación ISOMAP con X

Aplicar Transformación ISOMAP a X

Fin Para

Realizar scatter Plot

Fin Para

Algoritmo 5 Algoritmo para la aplicación del método PCA para la eliminación de variables.

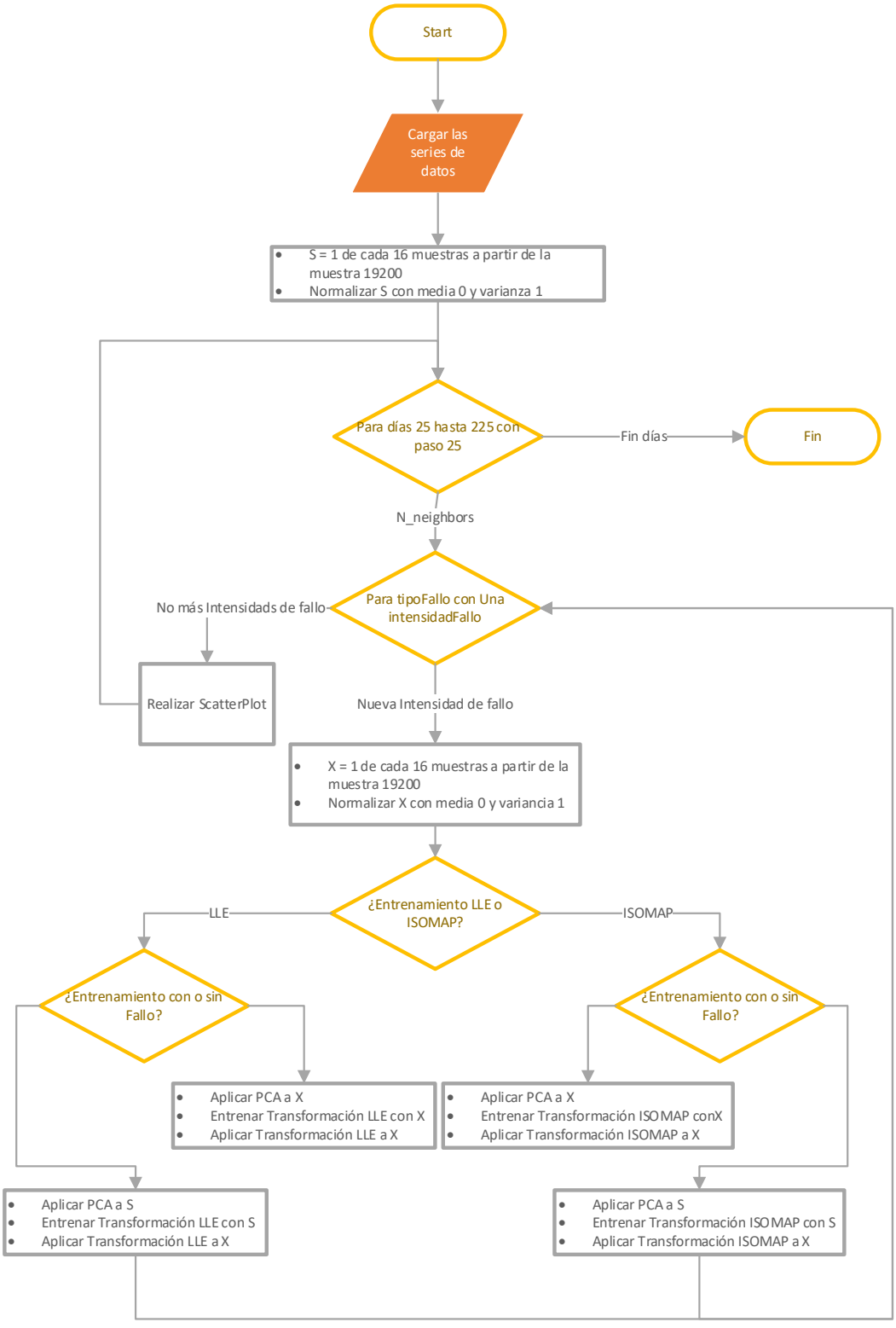


Figura 51 Proceso para aplicar el método PCA combinado con Manifold.

A continuación, se muestran los resultados de las pruebas realizadas con PCA (columna izquierda de las figuras) los cuales se comparan con los resultados del apartado 4.4.1.2 (columna derecha de las figuras). Como se puede observar en la Figura 52 y la Figura 53 se ha aplicado como entrenamiento la serie de datos sin fallo y se ha aplicado a uno de los fallos en concreto al fallo 1, y se compara el resultado tanto con el método LLE como con el método ISOMAP. Cabe destacar que no se aprecia una pérdida significativa de la información de los resultados, aunque tampoco se logra una distinción del tipo de fallo, e incluso se puede concluir que con PCA los datos de fallo se asemejan mucho más a los datos de comportamiento normal que cuando no se usa PCA, por lo que no parece que esto ayude a la identificación de los fallos.

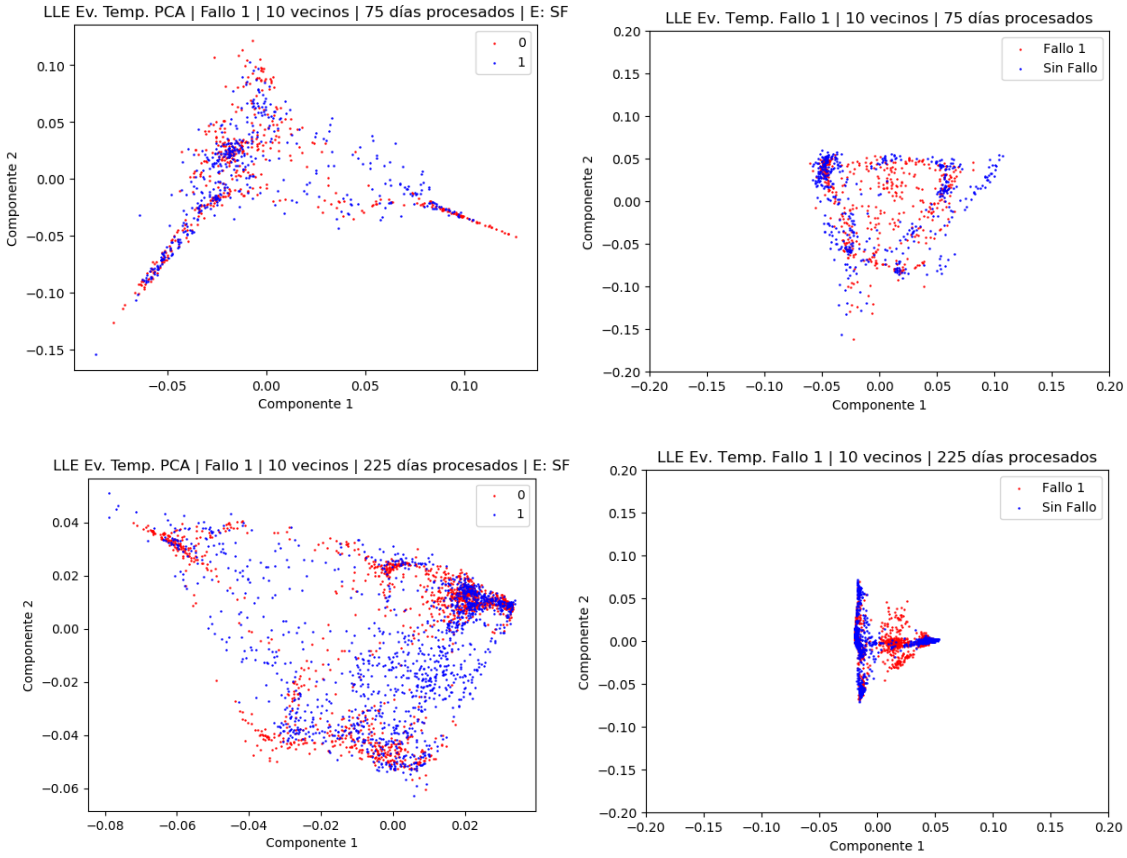


Figura 52 Comparativa entre las transformaciones con PCA (columna izquierda) y sin PCA (columna derecha) para LLE, aplicándose el entrenamiento con la serie sin fallos.

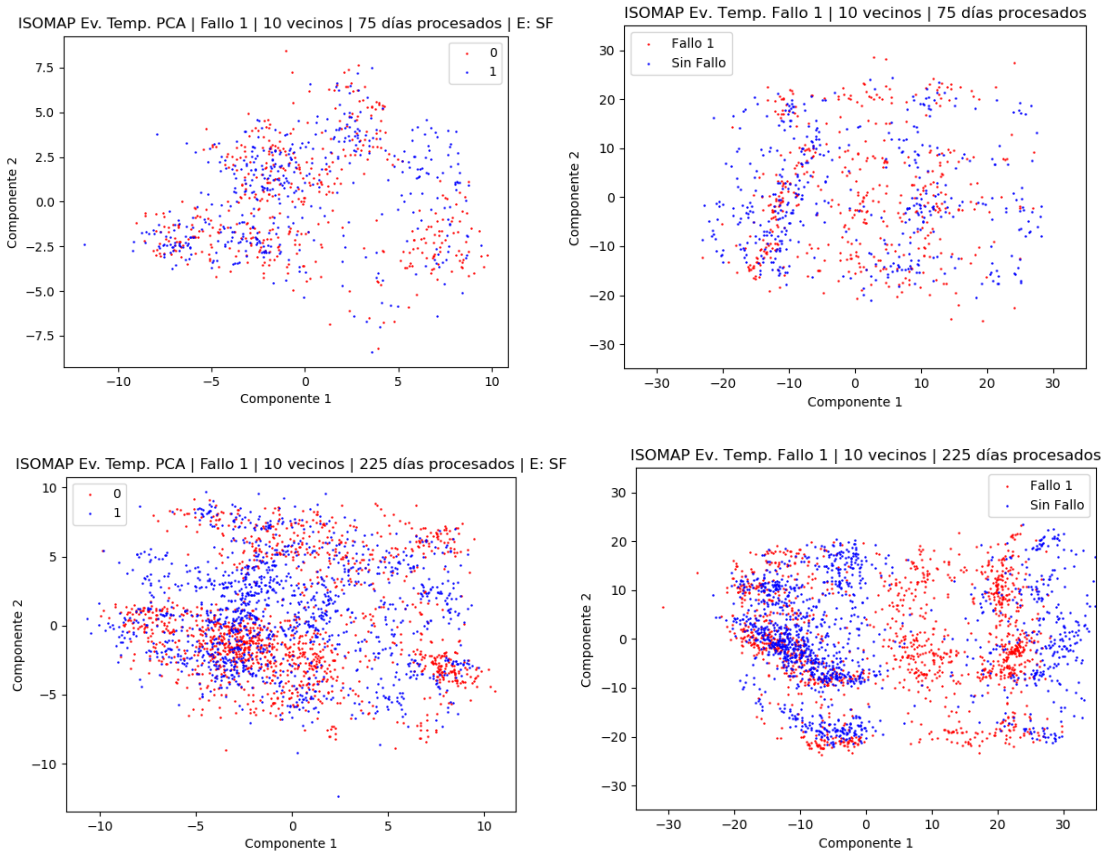


Figura 53 Comparativa entre las transformaciones con PCA (columna izquierda) y sin PCA (columna derecha) para ISOMAP, aplicándose el entrenamiento con la serie sin fallos.

En segundo lugar, se muestran la Figura 54 y la Figura 55 que contienen las transformaciones realizadas con su propio fallo y nuevamente se comparan aquellas a las que se les ha aplicado el método PCA (columna izquierda), con las del apartado anterior (columna derecha). Puede apreciarse en los resultados como la pérdida de información ha causado que el fallo sea menos distinguible en comparación a cuando no existe fallo. Aun así, no implica que no sea posible aplicar el método PCA y lograr una clasificación adecuada, aunque posiblemente sea más dificultoso.

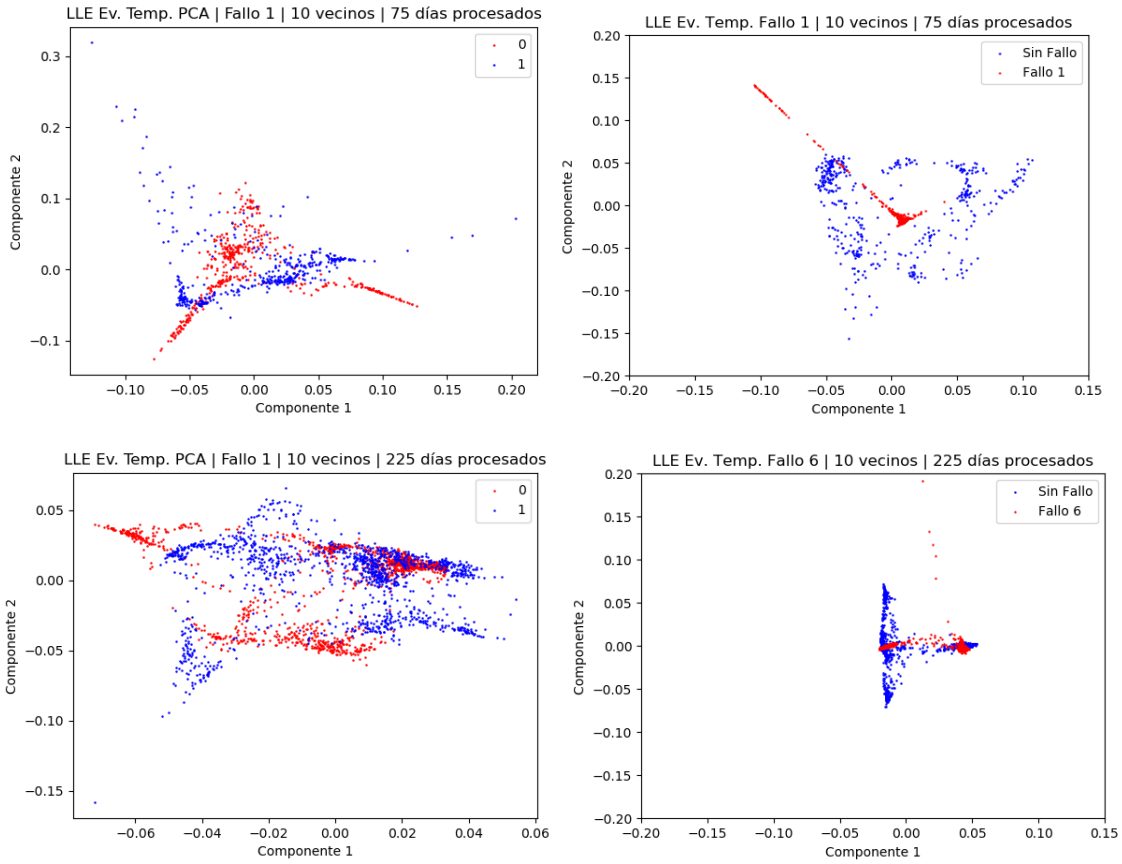


Figura 54 Comparativa entre las transformaciones con PCA (columna izquierda) y sin PCA (columna derecha) para LLE, aplicándose el entrenamiento con su propio fallo.

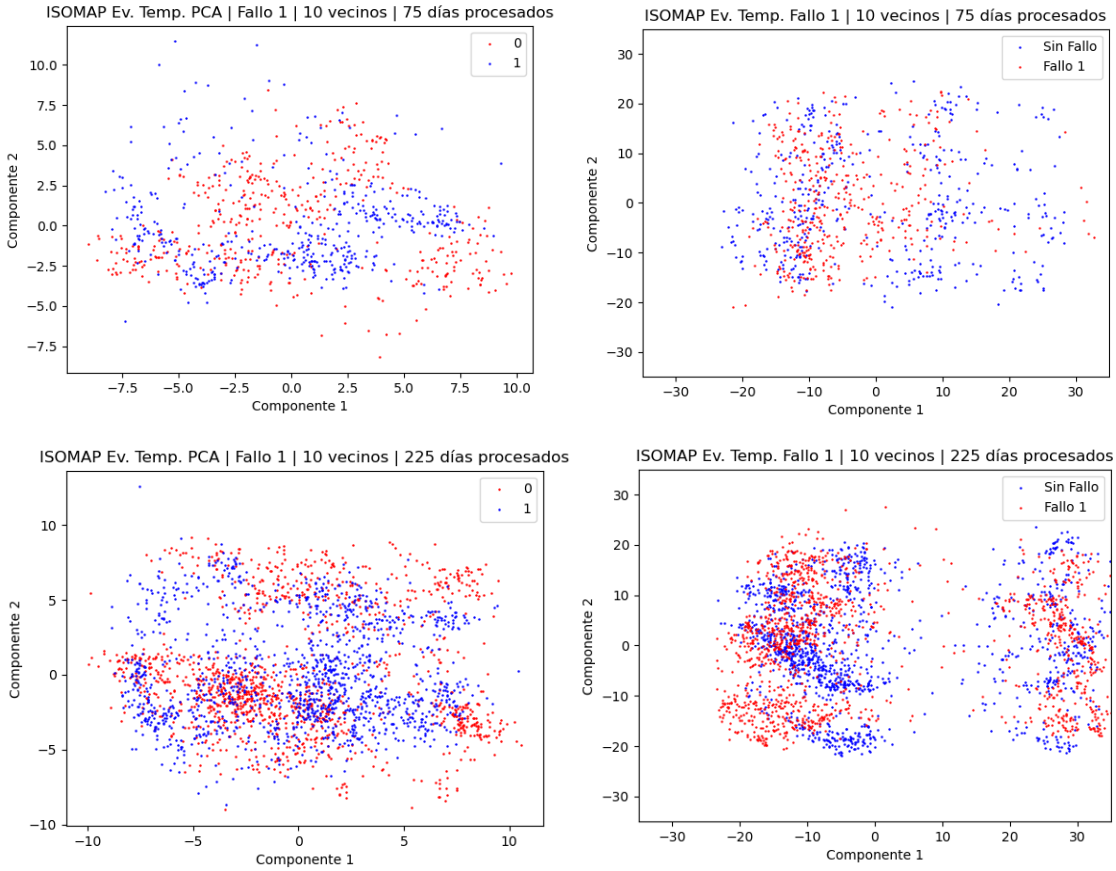


Figura 55 Comparativa entre las transformaciones con PCA (columna izquierda) y sin PCA (columna derecha) para ISOMAP, aplicándose el entrenamiento con su propio fallo.

4.4.2 Evaluación de los algoritmos de agrupación

En este segundo grupo de pruebas se analizan los algoritmos de agrupamiento o *clustering* basados en densidad explicados en el Capítulo 2.- junto a los métodos *manifold*. Es importante tener en cuenta que en las pruebas realizadas se mostrarán tanto los resultados obtenidos aplicando únicamente los métodos *manifold* como las agrupaciones generadas por los algoritmos. Además, se realizarán algunas pruebas distintas a las del apartado 4.4.1, donde la diferencia principal está en que se concatenarán los fallos creando un vector que contenga varios fallos y se aplicarán las transformaciones al conjunto del vector. De esta manera se explora desde otra aproximación la clasificación de los fallos y se pueden ampliar las conclusiones obtenidas. Así mismo se realizarán ciertas pruebas con los datos para 75 días siguiendo las conclusiones extraídas que indicaban la posibilidad de una mejor clasificación a pesar de tener menor número de muestras.

Durante las pruebas realizadas se variarán los parámetros principales del método escogido. Mientras que para HDBSCAN solo es necesario variar un parámetro, para OPTICS se variarán tres.

- *Eps*: Especifica la distancia máxima entre dos puntos para que puedan considerarse vecinos. No obstante, no implica que sea la distancia máxima que puede existir dentro de un grupo entre todos los vecinos. Este parámetro se variará en DBSCAN y OPTICS.
- *minPoints:* El número mínimo de puntos para formar un grupo. Este parámetro es el único que se variará en HDBSCAN además de hacerlo en DBSCAN y OPTICS
- ε: Permite especificar cuándo se deben dejar de seguir formando clústeres según la densidad de los puntos, como resultado permite acelerar el algoritmo a costa de no clasificar los puntos que no alcancen la densidad umbral. Es necesario subrayar que ε se especifica únicamente en OPTICS.

4.4.2.1 Análisis de los algoritmos de densidad tratando los fallos de forma individual

En primer lugar, se analiza cada fallo por separado tratando de buscar una buena clasificación que permita distinguir los fallos y una buena diferenciación entre cada uno de ellos. Pues si resultara que cada fallo tiene unas características muy diferenciadas podría generarse una base de datos y sólo sería necesario comparar el nuevo fallo que se quisiera clasificar para ver a cuál de los fallos de la base de datos se asemeja más. Con este proceso se pueden crear tablas por cada fallo, mostrando cuál es la mejor elección de parámetros para cada método de agrupamiento y para cada tipo de transformación que queramos realizar, tanto para PCA como para el tipo de entrenamiento. El procedimiento seguido se explica en el Algoritmo 6 y en el diagrama de flujo de la Figura 56.

tipoTransformacion = "LLE" o "ISOMAP"

tipoEntrenamiento = "Sin fallo" o "*"

sinPCA = True o False

S = 1 de cada 16 muestras de los datos sin fallo a partir de la muestra 19200

Normalizar *S* con media 0 y varianza 1

Entrenamiento con serie sin fallo (EsinFallo) = True o False

Para cada *tipoFallo* para todo *intensidadFallo*

X = 1 de cada 16 muestras a partir de la muestra 19200 de cada intensidadFallo

Normalizar *X* con media 0 y varianza 1

Si *tipoEntrenamiento "Sin fallo"*

Si *tipoTransformacion LLE* Entonces

Entrenar los datos atendiendo a si deben procesarse con PCA

Entrenar Transformación LLE con S

Aplicar Transformación LLE a X

Sino Entonces

Entrenar los datos atendiendo a si deben procesarse con PCA

Entrenar Transformación ISOMAP con S

Aplicar Transformación ISOMAP a X

Sino Entonces

Si *tipoTransformacion LLE* Entonces

Entrenar los datos atendiendo a si deben procesarse con PCA

Entrenar Transformación LLE con X

Aplicar Transformación LLE a X

Sino Entonces

Entrenar los datos atendiendo a si deben procesarse con PCA

Entrenar Transformación ISOMAP con X

Aplicar Transformación ISOMAP a X

Si DBSCAN

Para los parámetros Eps y min_samples

Realizar DBSCAN

Analizar DBSCAN

Realizar Cluster Plot

Fin Para

Si HDBSCAN

Para los parámetros min_cluster

Realizar HDBSCAN

Analizar HDBSCAN

Realizar Cluster Plot

Fin Para

Si OPTICS

Para los parámetros min_samples, xi y min_cluster

Realizar OPTICS

Analizar OPTICS

Realizar Cluster Plot

Fin Para

Fin Para

Realizar scatter Plot

Algoritmo 6 Algoritmo para el análisis con algoritmos de densidad para cada fallo de forma individual.

El primer paso es aplicar la transformación (LLE o ISOMAP) y el algoritmo de densidad (DBSCAN, HDBSCAN u OPTICS) con el objetivo de obtener una tabla como la Tabla 1 para cada fallo, buscando cual es la mejor opción de los parámetros de los métodos de agrupamiento, variando dichos parámetros entre intervalos adecuados. Para a continuación, escoger los parámetros óptimos según el valor de los coeficientes ya mencionados en el Capítulo 2, usados para evaluar los métodos de agrupamiento.

Tabla 1 Fallo 0 con PCA y entrenamiento con el propio fallo.

CON PCA	EconFallo				
Fallo 0	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
DBSCAN(0,0008,5)	18,00	357,00	-0,03	1,22	215,80
DBSCAN(0,0008,10)	10,00	546,00	0,01	1,12	190,90
DBSCAN(0,0008,15)	5,00	645,00	0,09	1,47	228,74
DBSCAN(0,004,5)	8,00	22,00	0,55	1,31	3446,28
DBSCAN(0,004,10)	6,00	77,00	0,58	11,73	3747,09
DBSCAN(0,004,15)	6,00	98,00	0,56	4,25	3137,77
DBSCAN(0,008,5)	1,00	2,00	-0,05	2,04	0,73
DBSCAN(0,008,10)	1,00	2,00	-0,05	2,04	0,73
DBSCAN(0,008,15)	3,00	5,00	0,56	1,04	1385,49

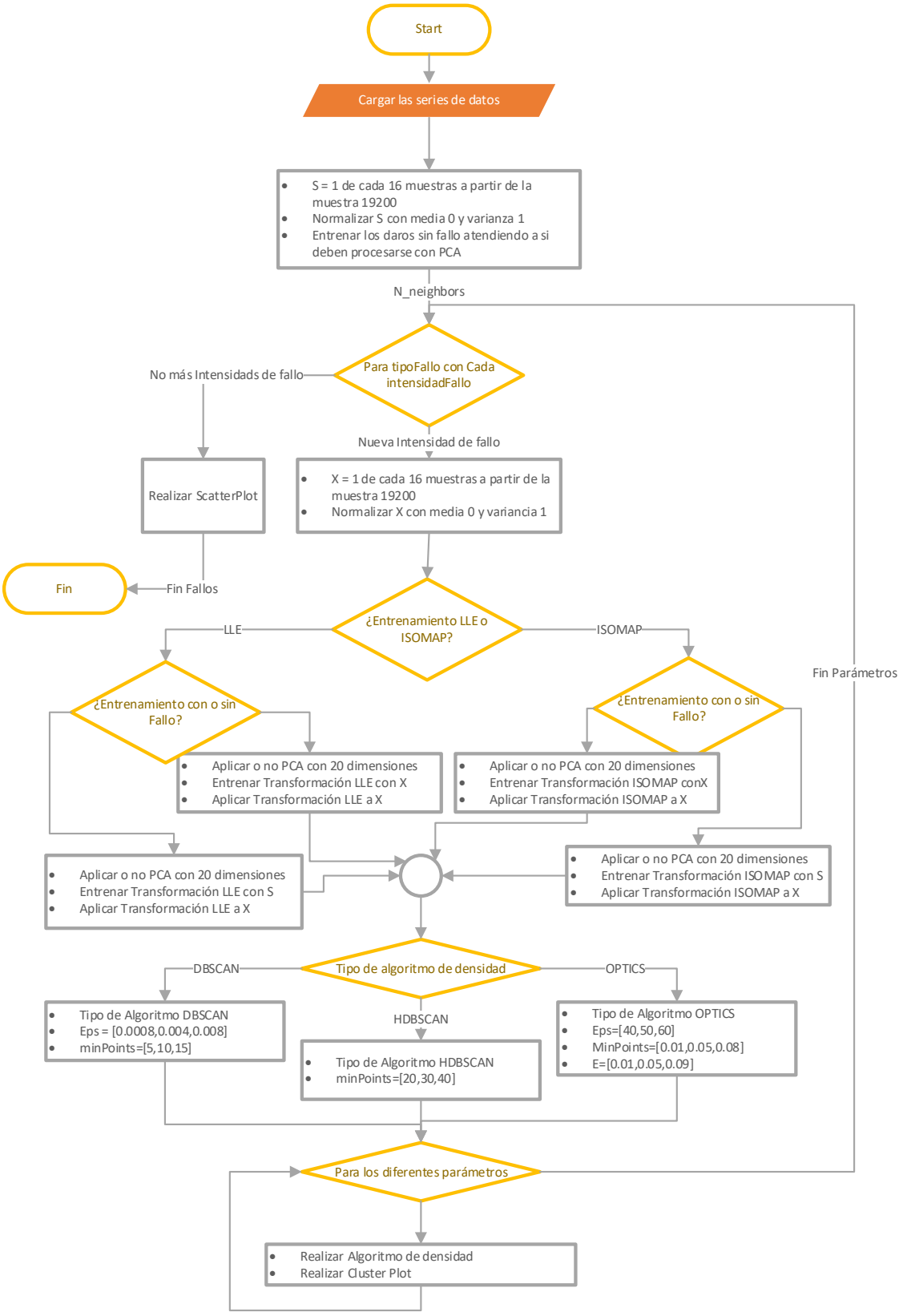


Figura 56 Diagrama de Flujo para el análisis con algoritmos de densidad para cada fallo de forma individual.

Una vez se ha realizado una tabla similar a la Tabla 1 para cada tipo de fallo se escoge aquellos parámetros que reflejan un mejor comportamiento de los algoritmos y se generan las tablas correspondientes. En concreto, la Tabla 1, contiene el mejor resultado del clustering para el caso de entrenamiento con el propio fallo, método PCA, transformación LLE y usando DBSCAN, la Tabla 2 y la Tabla 3 con las mismas condiciones pero usando los algoritmos de agrupamiento HDBSCAN y OPTICS respectivamente. La Tabla 5, Tabla 6 y

Tabla 7 con las mismas condiciones que las tablas Tabla 2, Tabla 3 y Tabla 4 pero usando la transformación ISOMAP.

Tabla 2 Todos los fallos con PCA y entrenamiento con el propio fallo utilizando LLE y DBSCAN.

Fallo	Parámetros	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
Fallo 0	DBSCAN(0,004,5)	8,00	22,00	0,55	1,31	3446,28
Fallo 1	DBSCAN(0,004,10)	3,00	50,00	0,65	0,74	3634,15
Fallo 2	DBSCAN(0,004,10)	3,00	47,00	0,70	0,70	5111,29
Fallo 3	DBSCAN(0,008,10)	3,00	0,00	0,81	0,21	7490,22
Fallo 4	DBSCAN(0,004,10)	3,00	58,00	0,73	0,72	5423,93
Fallo 5	DBSCAN(0,004,10)	3,00	58,00	0,72	0,73	4480,67
Fallo 6	DBSCAN(0,004,10)	2,00	44,00	0,90	1,37	925,12
Fallo 7	DBSCAN(0,008,15)	3,00	22,00	0,76	0,67	7351,26
Fallo 8	DBSCAN(0,008,15)	2,00	21,00	0,90	0,99	1434,19
Fallo 9	DBSCAN(0,004,15)	6,00	174,00	0,51	1,76	1777,00
Fallo 10	DBSCAN(0,004,15)	3,00	203,00	0,51	0,76	1020,71
Fallo 11	DBSCAN(0,004,10)	7,00	76,00	0,59	0,94	1600,10
Fallo 12	DBSCAN(0,008,5)	1,00	19,00	0,84	0,73	659,77
Fallo 13	DBSCAN(0,008,15)	1,00	22,00	0,82	0,81	584,60
Fallo 14	DBSCAN(0,004,15)	2,00	25,00	0,82	0,80	1762,12
Fallo 15	DBSCAN(0,004,15)	2,00	25,00	0,82	0,80	1762,12
Fallo 16	DBSCAN(0,008,15)	1,00	34,00	0,79	0,73	730,61

Tabla 3 Todos los fallos con PCA y entrenamiento con el propio fallo utilizando LLE y HDBSCAN

Fallo	Parámetros	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
Fallo 0	HDBSCAN(40)	5,00	219,00	0,59	1,36	4251,39
Fallo 1	HDBSCAN(40)	5,00	128,00	0,61	0,99	1899,89
Fallo 2	HDBSCAN(40)	4,00	50,00	0,68	0,80	3504,64
Fallo 3	HDBSCAN(40)	4,00	70,00	0,68	0,93	3872,61
Fallo 4	HDBSCAN(40)	3,00	20,00	0,73	0,73	5056,18

Fallo 5	HDBSCAN(40)	4,00	114,00	0,63	1,02	2691,82
Fallo 6	HDBSCAN(40)	2,00	26,00	0,91	1,22	1129,95
Fallo 7	HDBSCAN(40)	3,00	6,00	0,79	0,23	7078,41
Fallo 8	HDBSCAN(40)	4,00	166,00	0,57	1,29	335,77
Fallo 9	HDBSCAN(40)	3,00	110,00	0,61	0,93	3051,22
Fallo 10	HDBSCAN(20)	4,00	206,00	0,60	0,76	3344,82
Fallo 11	HDBSCAN(40)	3,00	100,00	0,64	0,74	2934,08
Fallo 12	HDBSCAN(30)	3,00	127,00	0,73	1,38	497,99
Fallo 13	HDBSCAN(20)	3,00	70,00	0,76	1,28	626,22
Fallo 14	HDBSCAN(20)	3,00	69,00	0,71	1,39	562,18
Fallo 15	HDBSCAN(20)	3,00	69,00	0,71	1,39	562,18
Fallo 16	HDBSCAN(30)	3,00	286,00	0,47	1,43	491,42

Tabla 4 Todos los fallos con PCA y entrenamiento con el propio fallo utilizando LLE y OPTICS.

Fallo	Parámetros	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
Fallo 0	OPTICS(60,0,05,0,09)	4,00	154,00	0,66	1,39	4632,90
Fallo 1	OPTICS(40,0,08,0,05)	3,00	476,00	0,53	0,79	1528,23
Fallo 2	OPTICS(40,0,08,0,05)	2,00	859,00	0,51	0,53	1323,83
Fallo 3	OPTICS(50,0,08,0,05)	2,00	861,00	0,50	0,54	1258,36
Fallo 4	OPTICS(60,0,08,0,05)	3,00	502,00	0,50	0,84	1168,83
Fallo 5	OPTICS(60,0,01,0,05)	5,00	156,00	0,60	4,05	1698,89
Fallo 6	OPTICS(40,0,08,0,05)	3,00	491,00	0,52	0,83	1233,81
Fallo 7	OPTICS(50,0,08,0,05)	3,00	489,00	0,53	0,79	1334,58
Fallo 8	OPTICS(50,0,08,0,05)	3,00	514,00	0,51	0,79	1304,68
Fallo 9	OPTICS(50,0,08,0,05)	2,00	967,00	0,46	0,53	1045,00
Fallo 10	OPTICS(60,0,08,0,01)	2,00	980,00	0,44	0,54	952,94
Fallo 11	OPTICS(60,0,08,0,05)	3,00	556,00	0,43	0,94	979,85
Fallo 12	OPTICS(60,0,08,0,05)	2,00	961,00	0,58	0,40	1582,92
Fallo 13	OPTICS(40,0,08,0,05)	2,00	951,00	0,58	0,41	1554,60
Fallo 14	OPTICS(60,0,05,0,05)	2,00	950,00	0,56	0,43	1424,18
Fallo 15	OPTICS(60,0,05,0,05)	2,00	950,00	0,56	0,43	1424,18
Fallo 16	OPTICS(50,0,08,0,05)	2,00	962,00	0,57	0,40	1533,18

Tabla 5 Todos los fallos con PCA y entrenamiento con el propio fallo utilizando ISOMAP y DBSCAN.

Fallo	Parámetros	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
Fallo 0	DBSCAN(3,10)	3,00	43,00	0,42	1,31	856,82
Fallo 1	DBSCAN(0,008,15)	3,00	22,00	0,62	0,54	3699,84
Fallo 2	DBSCAN(0,008,15)	3,00	28,00	0,70	0,65	5118,28

Fallo 3	DBSCAN(0,008,10)	3,00	0,00	0,81	0,21	7490,22
Fallo 4	DBSCAN(0,004,10)	3,00	58,00	0,73	0,72	5423,93
Fallo 5	DBSCAN(0,004,10)	3,00	58,00	0,72	0,73	4480,67
Fallo 6	DBSCAN(0,008,15)	2,00	19,00	0,92	1,05	1368,20
Fallo 7	DBSCAN(0,008,15)	3,00	22,00	0,76	0,67	7351,26
Fallo 8	DBSCAN(0,008,15)	2,00	21,00	0,90	0,99	1434,19
Fallo 9	DBSCAN(0,004,15)	6,00	174,00	0,51	1,76	1777,00
Fallo 10	DBSCAN(0,004,15)	3,00	203,00	0,51	0,76	1020,71
Fallo 11	DBSCAN(0,004,15)	5,00	150,00	0,59	1,34	2552,05
Fallo 12	DBSCAN(0,004,10)	1,00	24,00	0,81	0,91	508,88
Fallo 13	DBSCAN(0,008,10)	1,00	22,00	0,82	0,81	584,60
Fallo 14	DBSCAN(0,004,15)	2,00	25,00	0,82	0,80	1762,12
Fallo 15	DBSCAN(0,004,15)	2,00	25,00	0,82	0,80	1762,12
Fallo 16	DBSCAN(0,008,10)	1,00	34,00	0,79	0,73	730,61

Tabla 6 Todos los fallos con PCA y entrenamiento con el propio fallo utilizando ISOMAP y HDBSCAN.

Fallo	Parámetros	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
Fallo 0	HDBSCAN(20)	2,00	0,00	0,60	0,62	2269,27
Fallo 1	HDBSCAN(40)	2,00	1,00	0,48	0,48	1512,55
Fallo 2	HDBSCAN(50)	2,00	13,00	0,51	0,67	1827,37
Fallo 3	HDBSCAN(20)	2,00	0,00	0,65	0,53	3143,94
Fallo 4	HDBSCAN(30)	3,00	130,00	0,51	0,98	2119,08
Fallo 5	HDBSCAN(30)	3,00	53,00	0,58	1,00	1206,63
Fallo 6	HDBSCAN(40)	2,00	0,00	0,64	0,54	3081,50
Fallo 7	HDBSCAN(30)	2,00	0,00	0,64	0,56	2876,03
Fallo 8	HDBSCAN(30)	2,00	1,00	0,62	0,45	1495,84
Fallo 9	HDBSCAN(20)	4,00	143,00	0,42	1,40	873,56
Fallo 10	HDBSCAN(40)	2,00	185,00	0,43	2,01	808,54
Fallo 11	HDBSCAN(40)	3,00	201,00	0,45	1,22	1170,03
Fallo 12	HDBSCAN(50)	2,00	124,00	0,52	2,36	538,09
Fallo 13	HDBSCAN(30)	2,00	205,00	0,41	2,39	412,94
Fallo 14	HDBSCAN(30)	2,00	146,00	0,52	2,59	519,91
Fallo 15	HDBSCAN(10)	3,00	121,00	0,41	2,33	389,58
Fallo 16	HDBSCAN(30)	2,00	96,00	0,56	2,09	683,68

Tabla 7 Todos los fallos con PCA y entrenamiento con el propio fallo utilizando ISOMAP y OPTICS.

Fallo	Parámetros	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
Fallo 0	OPTICS(60,0,08,0,01)	1,00	387,00	0,61	0,61	2536,29
Fallo 1	OPTICS(50,0,08,0,01)	2,00	305,00	0,58	0,56	1879,44
Fallo 2	OPTICS(60,0,08,0,05)	1,00	410,00	0,66	0,50	3424,25
Fallo 3	OPTICS(60,0,08,0,05)	1,00	411,00	0,65	0,53	3121,82
Fallo 4	OPTICS(50,0,05,0,05)	2,00	836,00	0,55	0,52	2555,21
Fallo 5	OPTICS(50,0,08,0,05)	1,00	411,00	0,65	0,52	2931,90
Fallo 6	OPTICS(60,0,05,0,05)	3,00	107,00	0,53	1,28	1277,79
Fallo 7	OPTICS(40,0,05,0,09)	2,00	162,00	0,57	0,78	1733,01
Fallo 8	OPTICS(40,0,05,0,09)	2,00	836,00	0,50	0,55	2250,43
Fallo 9	OPTICS(60,0,01,0,09)	4,00	532,00	0,27	1,01	728,46
Fallo 10	OPTICS(50,0,05,0,09)	1,00	302,00	0,54	0,71	1526,01
Fallo 11	OPTICS(40,0,08,0,09)	1,00	1199,00	0,29	0,89	320,76
Fallo 12	OPTICS(50,0,05,0,05)	1,00	1022,00	0,57	0,60	1398,62
Fallo 13	OPTICS(60,0,05,0,01)	1,00	1290,00	0,24	0,82	161,60
Fallo 14	OPTICS(60,0,05,0,05)	2,00	103,00	0,56	1,60	796,27
Fallo 15	OPTICS(60,0,05,0,05)	2,00	103,00	0,56	1,60	796,27
Fallo 16	OPTICS(40,0,05,0,05)	2,00	38,00	0,60	0,76	1235,31

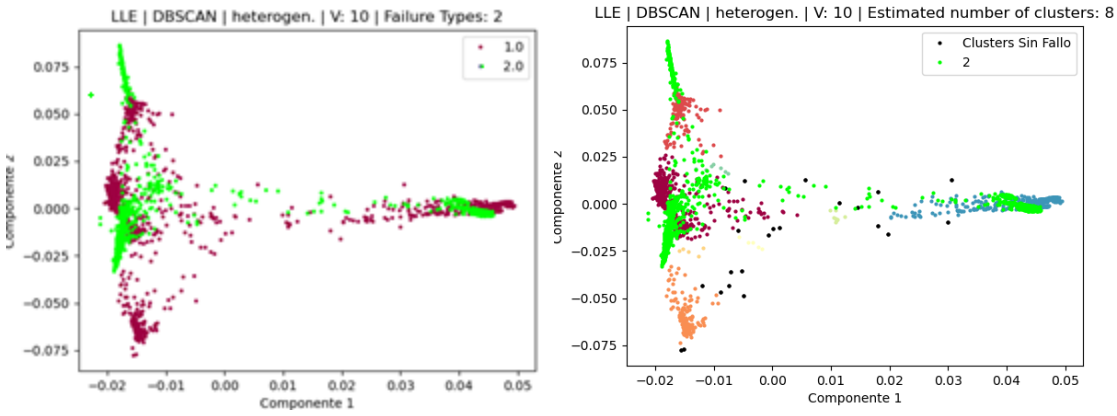
A continuación, se supone un caso práctico en el que se eliminan los datos de algunos de los fallos para poder utilizar una de las eliminaciones como un fallo sin clasificar.

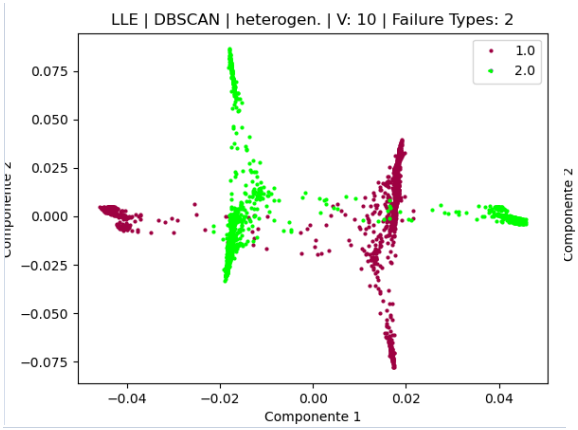
Tabla 8 Todos fallos eliminados para comprobación con PCA y entrenamiento con el propio fallo con LLE y DBSCAN.

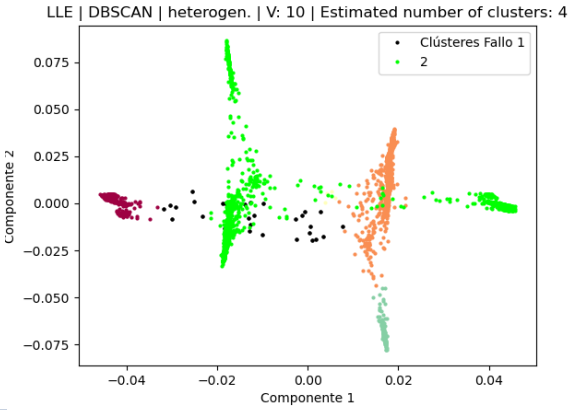
Fallo	Parámetros	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
Fallo 0	DBSCAN(0,004,5)	8,00	22,00	0,55	1,31	3446,28
Fallo 1	DBSCAN(0,004,10)	3,00	50,00	0,65	0,74	3634,15
Fallo 5	DBSCAN(0,004,10)	4,00	58,00	0,72	0,73	4480,67
Fallo 6	DBSCAN(0,004,10)	2,00	44,00	0,90	1,37	925,12
Fallo 8	DBSCAN(0,008,15)	2,00	21,00	0,90	0,99	1434,19

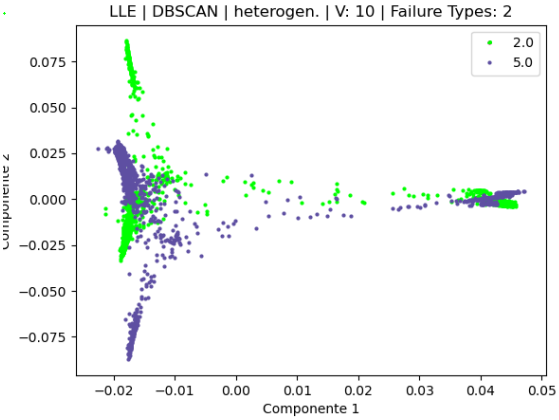
Fallo 9	DBSCAN(0,004,15)	6,00	174,00	0,51	1,76	1777,00
Fallo 12	DBSCAN(0,008,5)	1,00	19,00	0,84	0,73	659,77
Fallo 14	DBSCAN(0,004,15)	2,00	25,00	0,82	0,80	1762,12
Fallo 15	DBSCAN(0,004,15)	2,00	25,00	0,82	0,80	1762,12

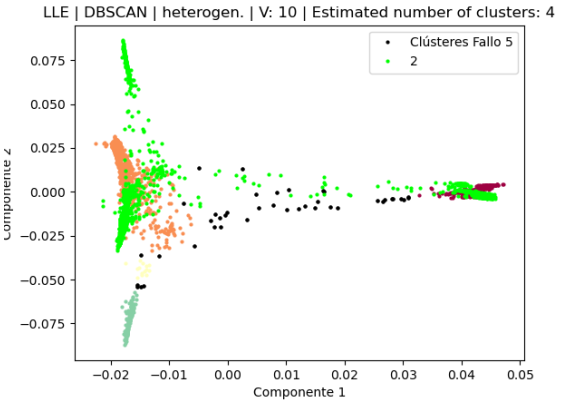
Ahora para clasificar el Fallo X (concretamente, es el fallo 2, fallo en el sensor de oxígeno 4º reactor, magnitud del 50%), se pasan los datos de este fallo por los clústeres definidos para cada tipo de fallo entrenado. Si los datos se encuentran dentro de alguno de los clústeres de fallo 1 o fallo 5, estará bien clasificado, si están fuera no se clasifica bien. Esto se puede ver en la Figura 57, que al analizarla se ve a la izquierda donde se sitúan los puntos de fallo X en comparación con los fallos de la Tabla 8. En la columna de la derecha se ven los clústeres definido por DBSCAN y donde se colocan los puntos del fallo X sobre los clústeres del fallo con el que se compara. En la Figura 58, se ve el mismo resultado comparando los datos de fallo Y (fallo 10, alcalinidad del primer reactor con una magnitud del 20%, con los datos (izquierda) y con los clústeres definidos por DBSCAN (derecha) para los mismos fallos que para el fallo X. Se puede ver en ambos resultados que los clústeres creados no son capaces de indicar a qué tipo de fallo pertenece. Aunque se puede apreciar, que el fallo X con quien más puntos en común tiene de todos los resultados, es con el fallo 5.

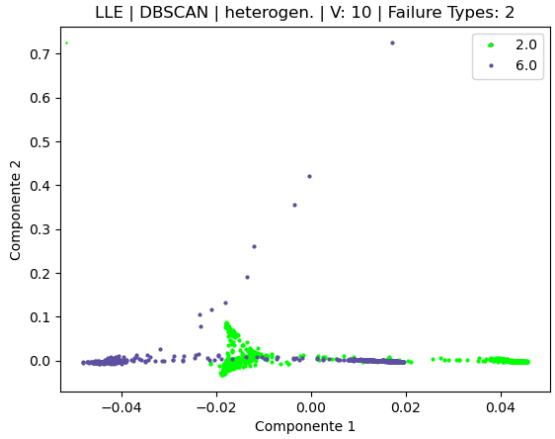


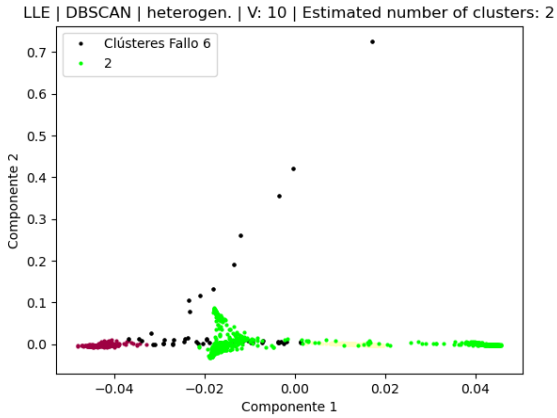

LLE | DBSCAN | heterogen. | V: 10 | Failure Types: 2
1.0
2.0
Componente 1

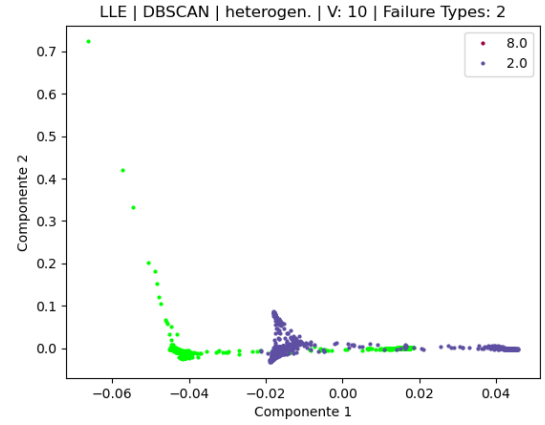

LLE | DBSCAN | heterogen. | V: 10 | Estimated number of clusters: 4
Clústeres Fallo 1
2
Componente 2
Componente 1

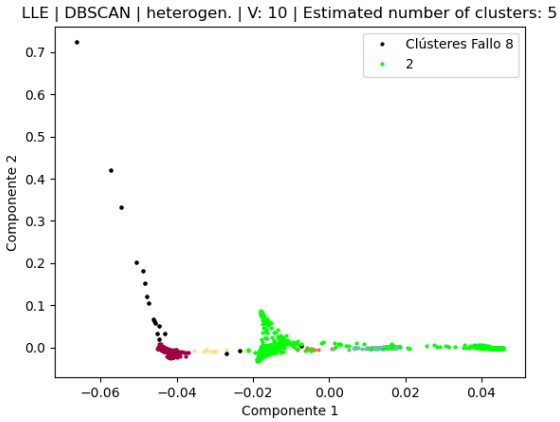

LLE | DBSCAN | heterogen. | V: 10 | Failure Types: 2
2.0
5.0
Componente 1


LLE | DBSCAN | heterogen. | V: 10 | Estimated number of clusters: 4
Clústeres Fallo 5
2
Componente 2
Componente 1


LLE | DBSCAN | heterogen. | V: 10 | Failure Types: 2
2.0
6.0
Componente 2
Componente 1


LLE | DBSCAN | heterogen. | V: 10 | Estimated number of clusters: 2
Clústeres Fallo 6
2
Componente 2
Componente 1


LLE | DBSCAN | heterogen. | V: 10 | Failure Types: 2
8.0
2.0
Componente 2
Componente 1


LLE | DBSCAN | heterogen. | V: 10 | Estimated number of clusters: 5
Clústeres Fallo 8
2
Componente 2
Componente 1

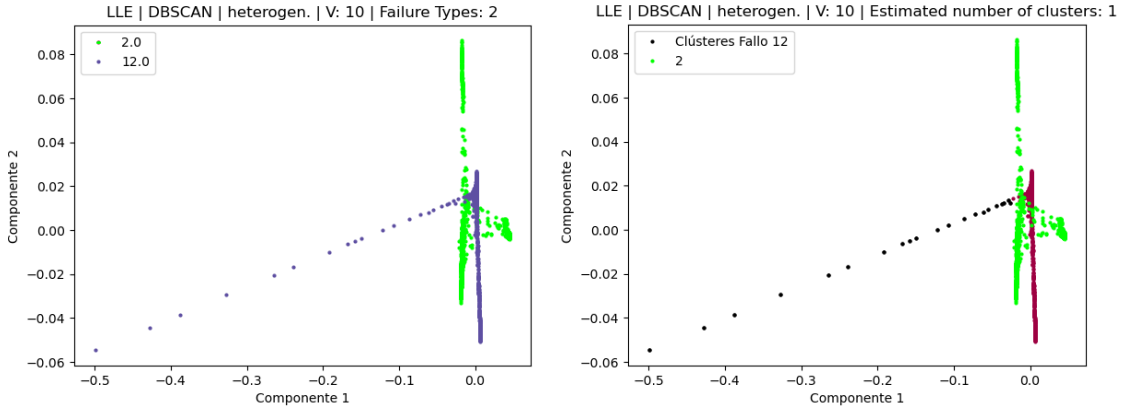
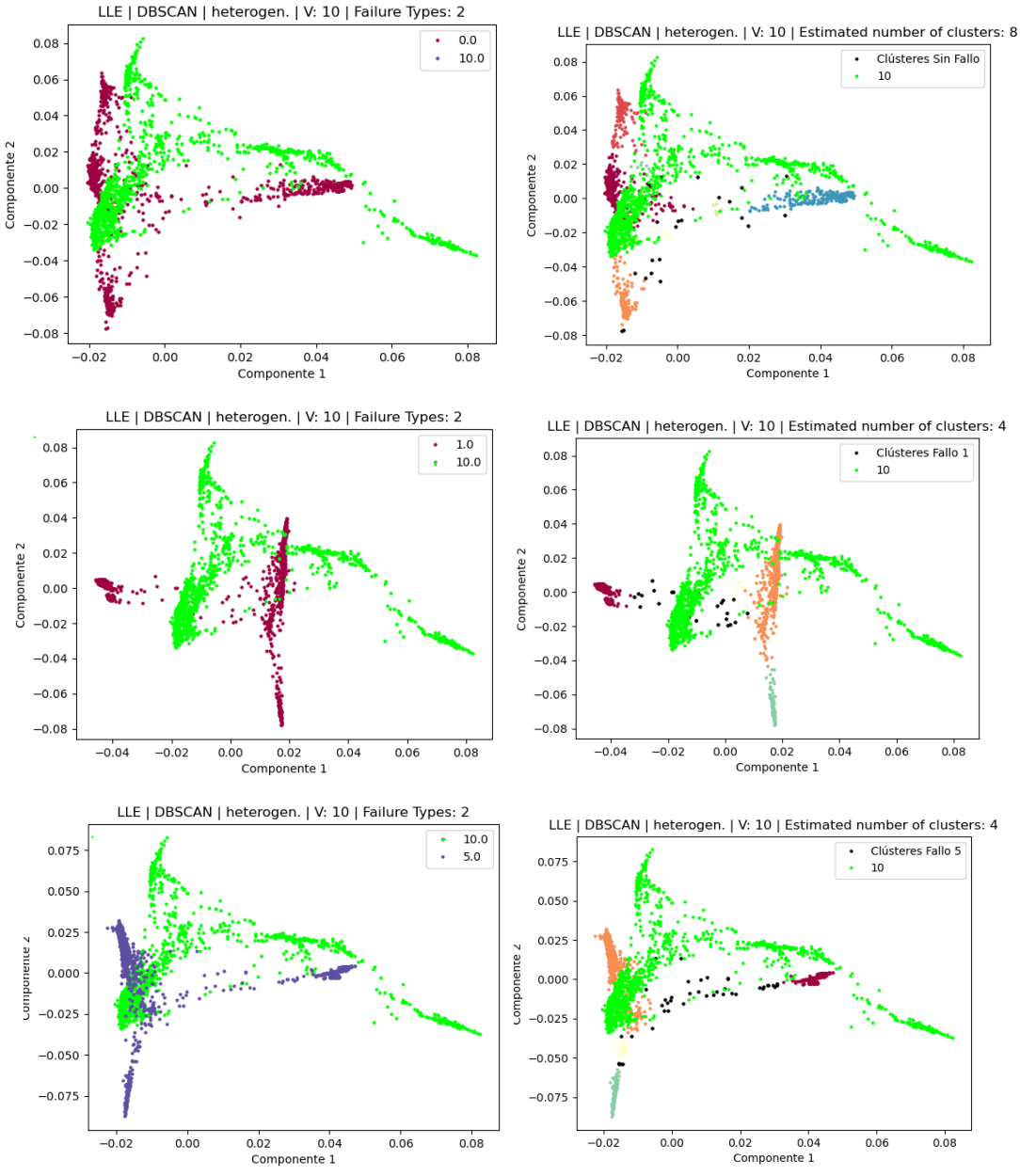


Figura 57 Comparativa entre la transformación de los fallos para el análisis del Fallo X y los fallos de la Tabla 8 (Sin fallo, Fallo 1, Fallo 5, Fallo 6, Fallo 8 y el Fallo 12). En la gráfica de la izquierda dónde se sitúa cada tipo de fallo. En la gráfica de la derecha los grupos formados por el método DBSCAN.



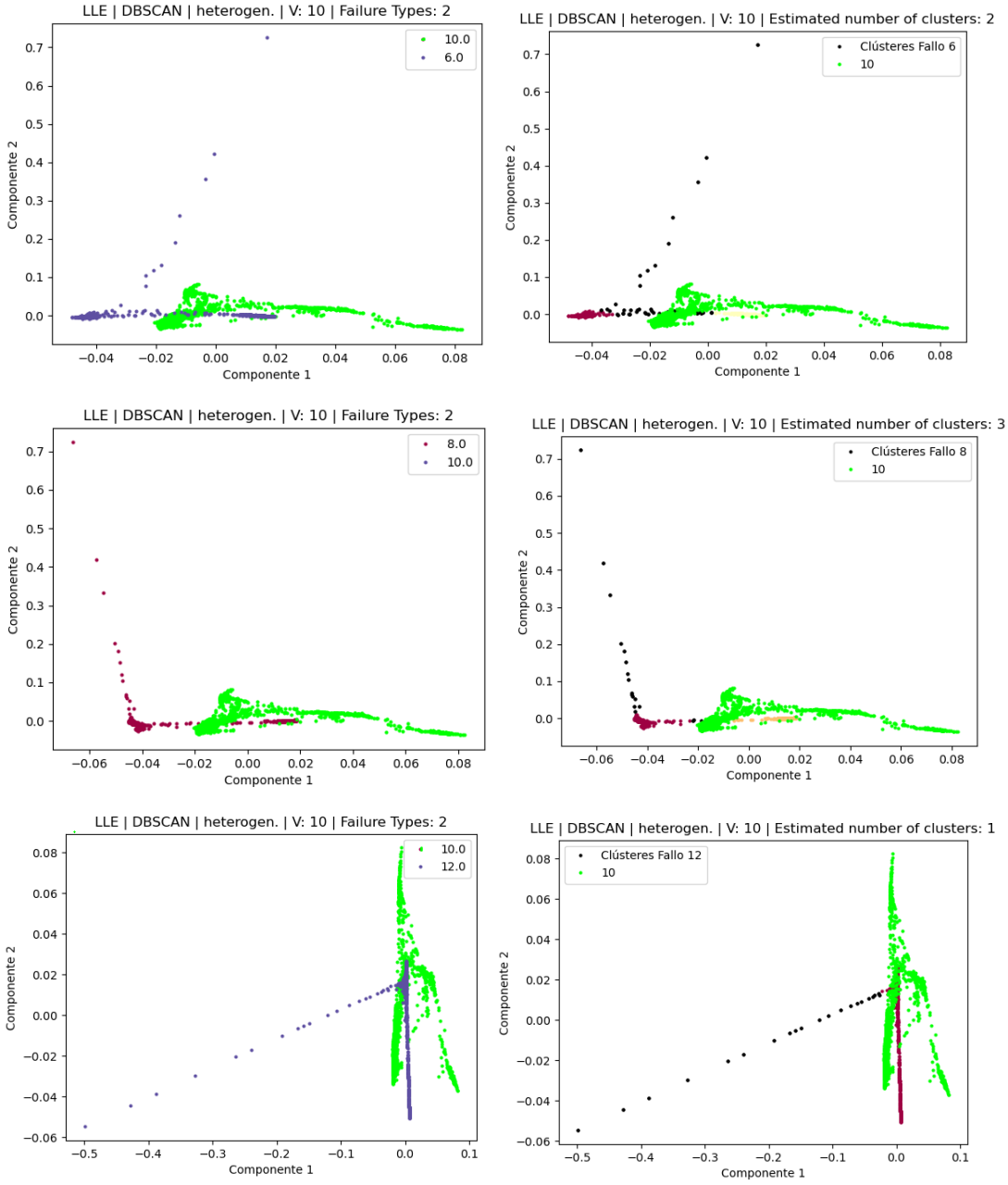


Figura 58 Comparativa entre la transformación de los fallos para el análisis del Fallo Y y los fallos de la Tabla 8 (Sin fallo, Fallo 1, Fallo 5, Fallo 6, Fallo 8 y el Fallo 12). En la gráfica de la izquierda dónde se sitúa cada tipo de fallo. En la gráfica de la derecha los grupos formados por el método DBSCAN.

A continuación, se aplica el mismo ejemplo realizado con el Fallo X, con la diferencia de que ahora los algoritmos *manifold* han sido entrenados con la serie de datos sin fallo. No se incluyen las tablas con todos los parámetros ni todas las comparaciones en la Figura 59 para no alargar demasiado la documentación. Como puede observarse en dicha figura en la columna derecha se ve como los grupos creados no distinguen entre fallos. Además, como se ha concluido en los análisis *manifold* los fallos son influenciados por el entrenamiento y es más complicado realizar cualquier distinción y clasificarlos.

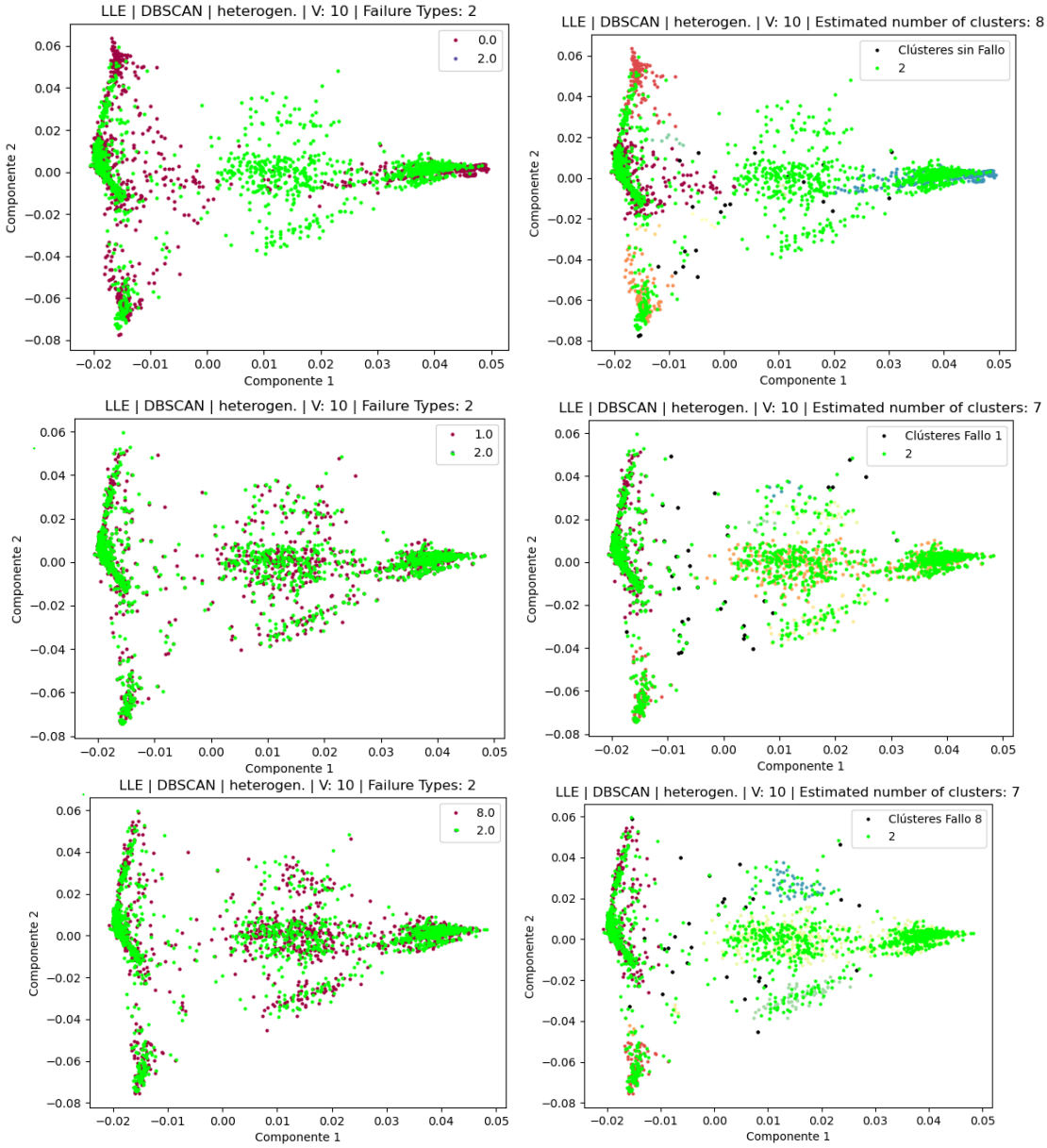


Figura 59 Comparativa entre la transformación de los fallos para el análisis del Fallo X y los fallos de la Tabla 8 (Sin fallo, Fallo 1 y Fallo 8) entrenados con la serie de datos sin fallos. En la gráfica de la izquierda dónde se sitúa cada tipo de fallo. En la gráfica de la derecha los grupos formados por el método DBSCAN.

4.4.2.2 *Análisis de los algoritmos de densidad concatenando los fallos en un mismo vector*

Como se ha concluido en el apartado 4.4.2.1 el análisis de los fallos por separado no ha otorgado buenos resultados aplicando los algoritmos de agrupamiento basados en densidad. Por lo consiguiente en este apartado se ha optado por realizar una clasificación tratando todos los tipos de fallo (únicamente se concatena un tipo de fallo en el vector) como una única serie de datos agrupados. Por lo tanto, se han concatenado todos los datos de los fallos y aplicando los algoritmos a la nueva serie completa se tratará de relacionar las agrupaciones que se crean con cada tipo de fallo en concreto.

Además, se realizarán los experimentos tanto con la reducción de dimensionalidad utilizando PCA (a excepción de cuando se entrenan los métodos con la serie sin fallo), que recordemos reduce de 141 dimensiones a 20, como sin ella. Destacar que al agrupar todos los fallos se ha multiplicado por 16 el número de muestras y ha aumentado de forma drástica el tiempo de computación, tanto en el algoritmo *manifold* (LLE o ISOMAP) como en los algoritmos de agrupación, añadiendo que con estos últimos se está realizando una iteración de diferentes parámetros.

Para esta tanda de pruebas se priorizará la heterogeneidad de las agrupaciones, dato el cual indica el índice de Calinski-Harabasz [35]. Por lo que se realizarán varias simulaciones variando su aproximación, con el objetivo de tener más posibilidades de obtener una solución adecuada.

4.4.2.2.1 Entrenamiento y aplicación con la serie de datos sin fallos

La primera prueba utiliza la serie de datos sin fallos para entrenar los métodos a pesar de que no ha arrojado buenos resultados en los experimentos anteriores, se realiza de esta manera para mantener la continuidad y la metodología empleada hasta ahora. Se crea un vector que incluye los datos de todos los tipos de fallo con una única intensidad *sin Fallo, Fallo 1, Fallo 6, Fallo 9, Fallo 12, Fallo 14 y Fallo 15.* Una vez se han obtenido la matriz en cuestión, se realiza el entrenamiento con la serie de datos sin fallos, utilizando los métodos LLE o ISOMAP, y al resultado de esta transformación se realizan los algoritmos de densidad. A su vez puede verse el proceso utilizado en el Algoritmo 7 y la Figura 60.

Hay que subrayar que debido a que se espera una mala transformación de los métodos por la influencia de la serie sin fallo, únicamente se mostrará para este apartado una de las pruebas realizadas en comparación a cuando los fallos se entrenan de forma distinta, en cuyo caso se muestran todas las pruebas. Concretamente las pruebas se realizarán sin PCA para tener el mayor número de datos posibles y como método de selección de los parámetros óptimos se ha escogido fijar el número de grupos lo más cercano a 6 y que tenga el índice de heterogeneidad más alto. Huelga decir que se han realizado todas las pruebas pero que para evitar mostrar todos los resultados que no funcionan no se han incluido en este documento.

Además, el rango de variación de los parámetros de los algoritmos de agrupamiento utilizados en esta prueba se muestra en la Tabla 9, esto se hace para buscar los valores óptimos de dichos parámetros que mejor agrupamiento resulte.

tipoTransformacion = “LLE” o “ISOMAP”

sinPCA = True o False

S = Serie sin fallo

V = Todos los fallos incluido la serie sin fallo

Normalizar *S* y V con media 0 y varianza 1

Si *tipoTransformacion LLE* Entonces

    Entrenar Transformación LLE con S

    Aplicar Transformación LLE a V

Sino Entonces

    Entrenar Transformación ISOMAP con S

    Aplicar Transformación ISOMAP a V

Si DBSCAN

    Para los parámetros eps y min_samples

        Realizar DBSCAN

        Analizar DBSCAN

        Realizar Cluster Plot

    Fin Para

Si HDBSCAN

    Para los parámetros min_cluster

        Realizar HDBSCAN

        Analizar HDBSCAN

        Realizar Cluster Plot

    Fin Para

Si OPTICS

    Para los parámetros min_samples, xi y min_cluster

        Realizar HDBSCAN

        Analizar HDBSCAN

        Realizar Cluster Plot

    Fin Para

Fin Para

Realizar scatter Plot

Algoritmo 7 Algoritmo para el análisis de los algoritmos de densidad concatenando los fallos entrenando los métodos con la serie sin fallos.

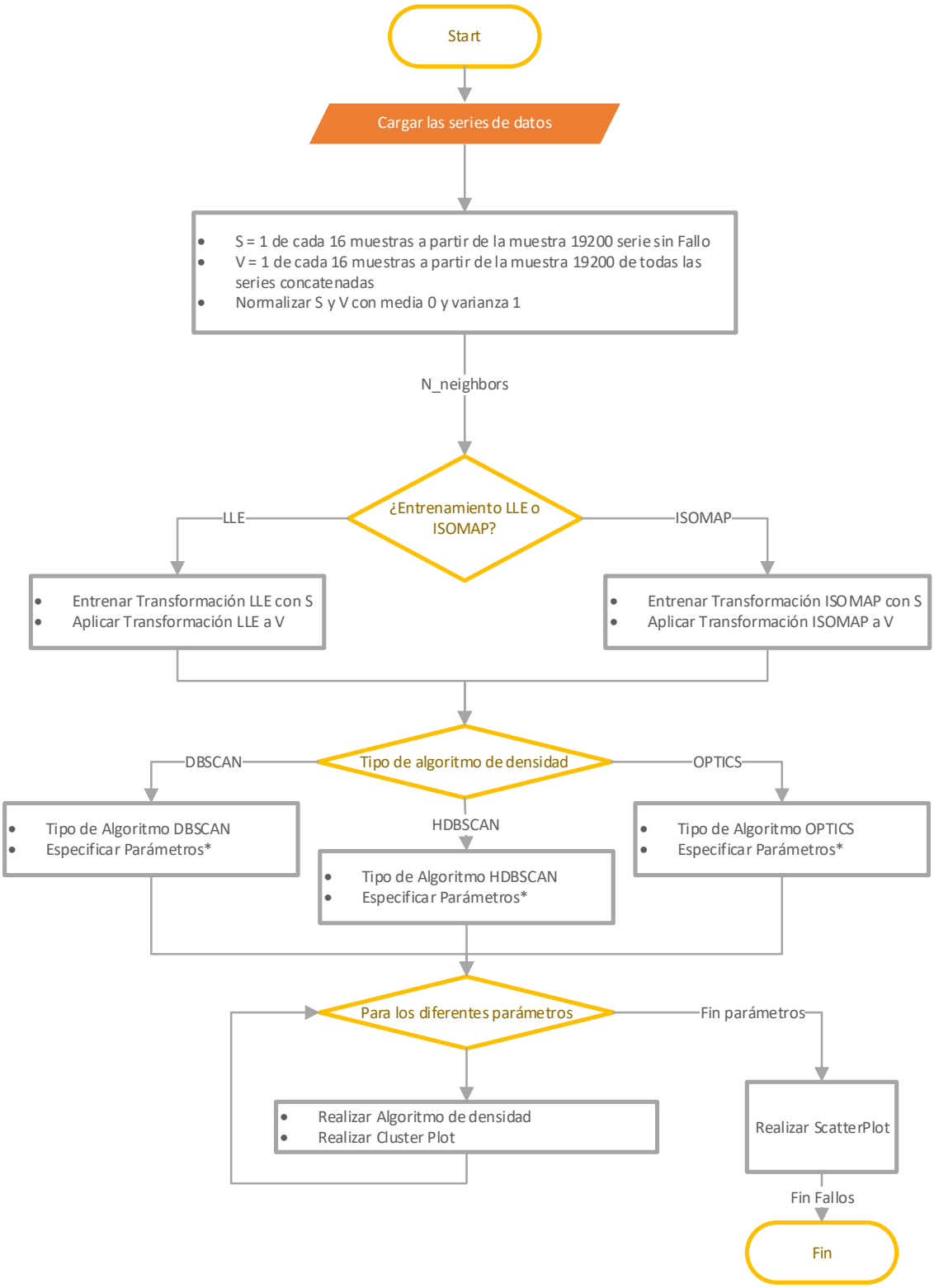


Figura 60 Proceso para análisis de los algoritmos de densidad concatenando los fallos entrenando los métodos con la serie sin fallos.

Tabla 9 Parámetros de los algoritmos de agrupamiento.

	manifold	Eps	minPoints	ε
DBSCAN	LLE	0.0008,0.001,0.002	30,32,34	-
	ISOMAP	0.4,0.5,0.6	51,53,55	-
HDBSCAN	LLE	-	60,65,70	-
	ISOMAP	-	28,33,38	-
OPTICS	LLE	50,60,70	0.01,0.005,0.001	0.01,0.25,0.05
	ISOMAP	45,50,55	0.007,0.01,0.03	0.008,0.01,0.03

A continuación, se muestran en las siguientes tablas algunos de los resultados de búsqueda de los parámetros más óptimos para los algoritmos de agrupamiento, para los distintos fallos y las distintas transformaciones. Así como gráficas donde se ven los clústeres obtenidos por cada método.

Tabla 10 Resultado del análisis concatenando los fallos entrenando los métodos con la serie sin fallos para LLE y DBSCAN por tipo de fallo.

	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
DBSCAN(0,0008,30)	17,00	4477,00	-0,04	1,14	823,90
DBSCAN(0,0008,32)	13,00	4848,00	-0,10	1,26	971,40
DBSCAN(0,0008,34)	11,00	5063,00	-0,14	1,39	1068,50
DBSCAN(0,001,30)	7,00	3402,00	0,09	0,99	2747,29
DBSCAN(0,001,32)	6,00	3586,00	0,11	1,07	2909,88
DBSCAN(0,001,34)	6,00	3669,00	0,10	1,07	2845,58
DBSCAN(0,002,30)	2,00	1418,00	0,41	1,51	8576,72
DBSCAN(0,002,32)	2,00	1461,00	0,41	1,51	8572,37
DBSCAN(0,002,34)	2,00	1512,00	0,41	1,49	8584,99

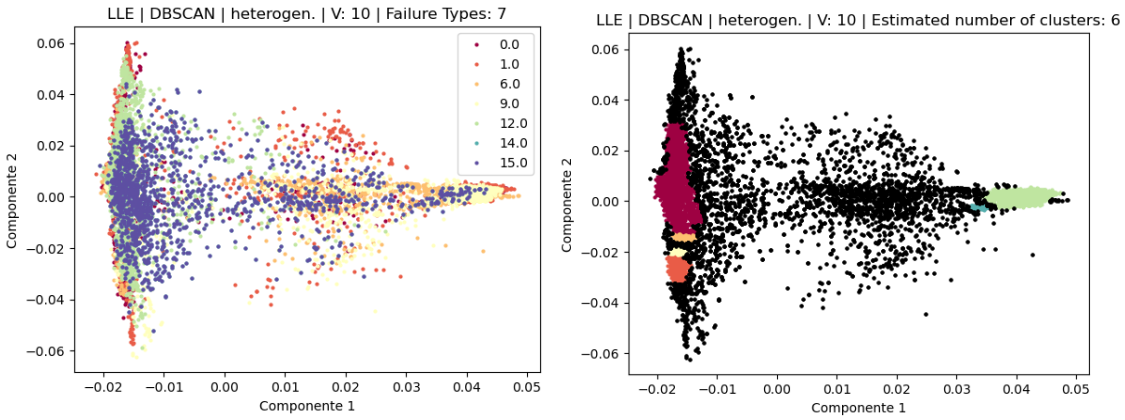


Figura 61 Resultado del análisis concatenando los fallos entrenando los métodos con la serie sin fallos para LLE y DBSCAN por tipo de fallo.

Tabla 11 Resultado del análisis concatenando los fallos entrenando los métodos con la serie sin fallos para ISOMAP y DBSCAN por tipo de fallo.

	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
DBSCAN(0,4,51)	2,00	9628,00	-0,51	1,10	78,76
DBSCAN(0,4,53)	1,00	9682,00	0,08	0,68	123,53
DBSCAN(0,4,55)	1,00	9682,00	0,08	0,68	123,53
DBSCAN(0,5,51)	4,00	9306,00	-0,50	0,99	162,50
DBSCAN(0,5,53)	4,00	9311,00	-0,50	0,99	160,90
DBSCAN(0,5,55)	4,00	9320,00	-0,50	0,99	156,59
DBSCAN(0,6,51)	8,00	8509,00	-0,44	1,04	218,51
DBSCAN(0,6,53)	6,00	8792,00	-0,44	1,10	226,31
DBSCAN(0,6,55)	5,00	8870,00	-0,45	1,09	252,87

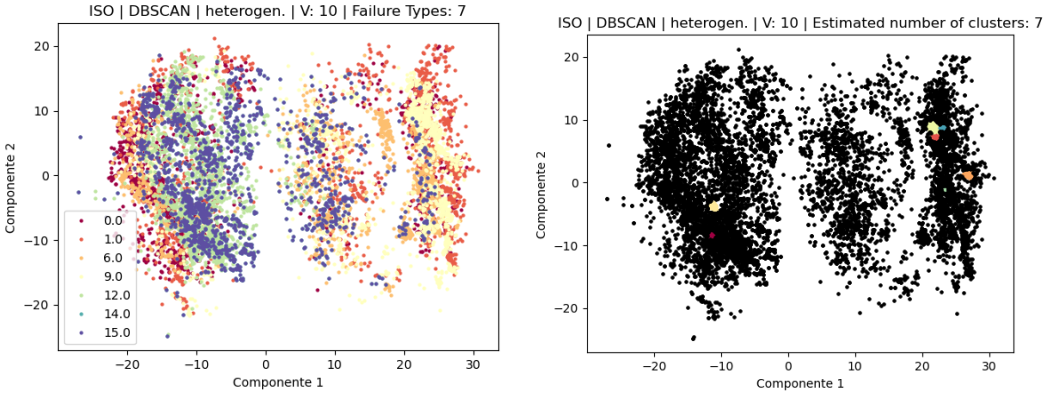


Figura 62 Resultado del análisis concatenando los fallos entrenando los métodos con la serie sin fallos para ISOMAP y DBSCAN por tipo de fallo.

Tabla 12 Resultado del análisis concatenando los fallos entrenando los métodos con la serie sin fallos para LLE y HDBSCAN por tipo de fallo.

	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
HDBSCAN(60)	7,00	5090,00	0,02	1,19	2585,37
HDBSCAN(65)	7,00	4424,00	0,10	1,34	3384,04
HDBSCAN(70)	4,00	2159,00	0,36	1,47	5103,54

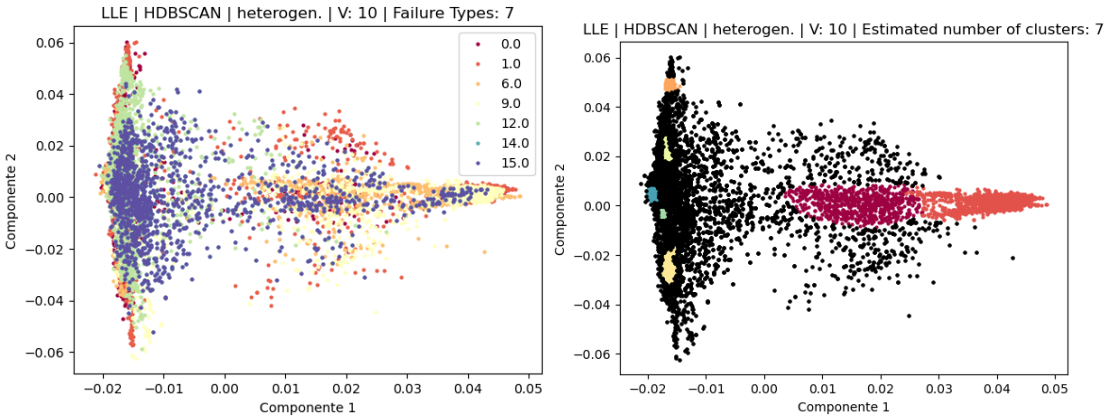


Figura 63 Resultado del análisis concatenando los fallos entrenando los métodos con la serie sin fallos para LLE y HDBSCAN por tipo de fallo.

Tabla 13 Resultado del análisis concatenando los fallos entrenando los métodos con la serie sin fallos para ISOMAP y HDBSCAN por tipo de fallo.

	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
HDBSCAN(28)	6,00	667,00	0,22	3,56	3721,38
HDBSCAN(33)	6,00	589,00	0,22	1,88	4200,99
HDBSCAN(38)	10,00	1217,00	0,27	1,62	2624,60

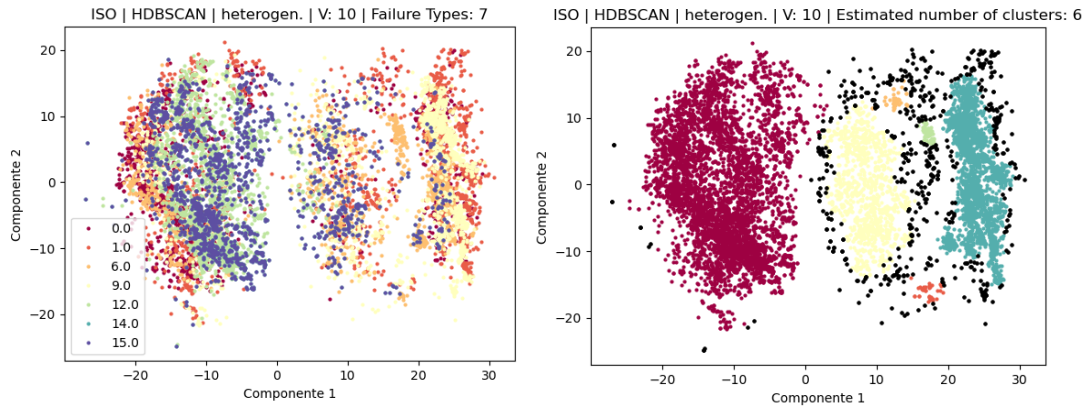


Figura 64 Resultado del análisis concatenando los fallos entrenando los métodos con la serie sin fallos para ISOMAP y HDBSCAN por tipo de fallo.

Tabla 14 Resultado del análisis concatenando los fallos entrenando los métodos con la serie sin fallos para LLE y OPTICS por tipo de fallo.

	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
OPTICS(50,0,01,0,05)	2,00	6923,00	-0,02	1,28	4617,82
OPTICS(50,0,01,0,025)	5,00	6160,00	-0,06	1,20	2343,67
OPTICS(50,0,01,0,01)	14,00	7858,00	-0,53	1,11	199,17
OPTICS(50,0,005,0,05)	3,00	8060,00	-0,36	1,76	566,73
OPTICS(50,0,005,0,025)	7,00	7434,00	-0,42	1,34	439,41
OPTICS(50,0,005,0,01)	16,00	7556,00	-0,50	1,11	203,76
OPTICS(50,0,001,0,05)	3,00	7962,00	-0,36	1,75	553,47
OPTICS(50,0,001,0,025)	7,00	7387,00	-0,42	1,35	450,96
OPTICS(50,0,001,0,01)	18,00	7106,00	-0,45	1,29	230,05
OPTICS(60,0,01,0,05)	3,00	6466,00	-0,01	1,12	3732,97
OPTICS(60,0,01,0,025)	4,00	6345,00	-0,03	1,24	2716,99
OPTICS(60,0,01,0,01)	10,00	8178,00	-0,54	1,17	227,56
OPTICS(60,0,005,0,05)	5,00	7056,00	-0,31	1,68	710,77
OPTICS(60,0,005,0,025)	7,00	7256,00	-0,39	1,30	499,63
OPTICS(60,0,005,0,01)	14,00	7499,00	-0,47	1,10	245,62
OPTICS(60,0,001,0,05)	5,00	6806,00	-0,27	1,60	789,21
OPTICS(60,0,001,0,025)	7,00	7421,00	-0,42	1,27	452,14
OPTICS(60,0,001,0,01)	16,00	6995,00	-0,41	1,05	311,39
OPTICS(70,0,01,0,05)	3,00	6389,00	0,00	1,12	3819,06
OPTICS(70,0,01,0,025)	4,00	6225,00	-0,01	1,23	2790,02
OPTICS(70,0,01,0,01)	10,00	8133,00	-0,49	0,88	285,70
OPTICS(70,0,005,0,05)	4,00	7575,00	-0,34	1,65	661,06
OPTICS(70,0,005,0,025)	7,00	7333,00	-0,40	1,30	460,76
OPTICS(70,0,005,0,01)	14,00	7491,00	-0,46	1,12	236,59
OPTICS(70,0,001,0,05)	5,00	6686,00	-0,25	1,63	817,92
OPTICS(70,0,001,0,025)	7,00	7324,00	-0,40	1,29	464,30
OPTICS(70,0,001,0,01)	15,00	7027,00	-0,40	1,06	314,61

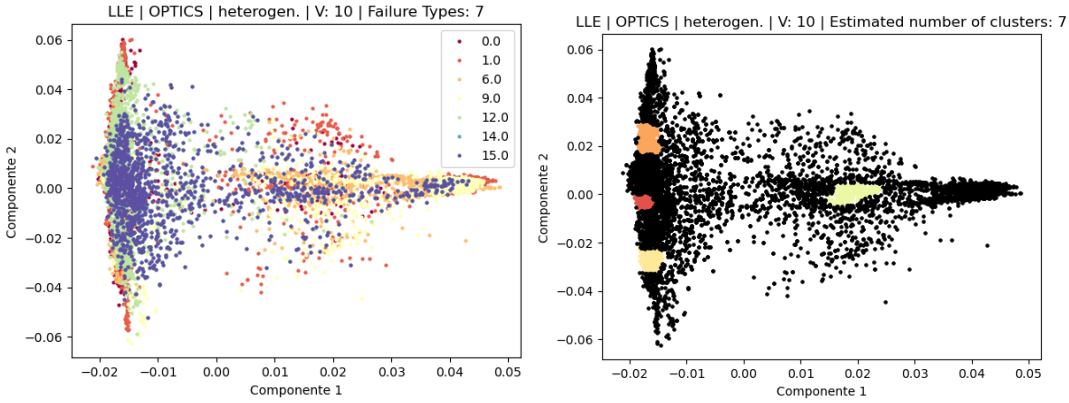


Figura 65 Resultado del análisis concatenando los fallos entrenando los métodos con la serie sin fallos para LLE y OPTICS por tipo de fallo.

Tabla 15 Resultado del análisis concatenando los fallos entrenando los métodos con la serie sin fallos para ISOMAP y OPTICS por tipo de fallo.

	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
OPTICS(45,0,03,0,03)	1,00	4264,00	0,56	0,68	17778,33
OPTICS(45,0,03,0,01)	6,00	8902,00	-0,53	1,39	158,54
OPTICS(45,0,03,0,008)	9,00	8856,00	-0,57	1,23	110,72
OPTICS(45,0,01,0,03)	10,00	5762,00	-0,20	1,63	477,95
OPTICS(45,0,01,0,01)	21,00	6224,00	-0,30	1,28	234,09
OPTICS(45,0,01,0,008)	29,00	5544,00	-0,23	1,28	242,75
OPTICS(45,0,007,0,03)	10,00	5947,00	-0,22	1,65	418,54
OPTICS(45,0,007,0,01)	21,00	5819,00	-0,25	1,30	277,51
OPTICS(45,0,007,0,008)	29,00	5298,00	-0,20	1,28	270,74
OPTICS(50,0,03,0,03)	1,00	4263,00	0,56	0,68	17800,23
OPTICS(50,0,03,0,01)	5,00	8993,00	-0,52	1,06	202,99
OPTICS(50,0,03,0,008)	7,00	8837,00	-0,54	1,02	179,81
OPTICS(50,0,01,0,03)	8,00	6575,00	-0,25	1,50	492,18
OPTICS(50,0,01,0,01)	21,00	6444,00	-0,32	1,28	213,64
OPTICS(50,0,01,0,008)	28,00	5826,00	-0,26	1,29	223,34
OPTICS(50,0,007,0,03)	10,00	5736,00	-0,18	1,53	525,16
OPTICS(50,0,007,0,01)	23,00	6297,00	-0,31	1,48	224,46
OPTICS(50,0,007,0,008)	29,00	5780,00	-0,25	1,29	236,76
OPTICS(55,0,03,0,03)	1,00	9452,00	-0,09	1,88	92,83
OPTICS(55,0,03,0,01)	7,00	8655,00	-0,51	1,44	158,87
OPTICS(55,0,03,0,008)	8,00	8576,00	-0,53	1,39	151,17
OPTICS(55,0,01,0,03)	9,00	6311,00	-0,23	1,50	475,43
OPTICS(55,0,01,0,01)	21,00	6003,00	-0,26	1,23	265,42
OPTICS(55,0,01,0,008)	27,00	6318,00	-0,32	1,32	195,92
OPTICS(55,0,007,0,03)	10,00	5706,00	-0,18	1,53	530,26
OPTICS(55,0,007,0,01)	21,00	5805,00	-0,24	1,23	295,53

OPTICS(55,0,007,0,008)	27,00	6033,00	-0,28	1,27	234,06

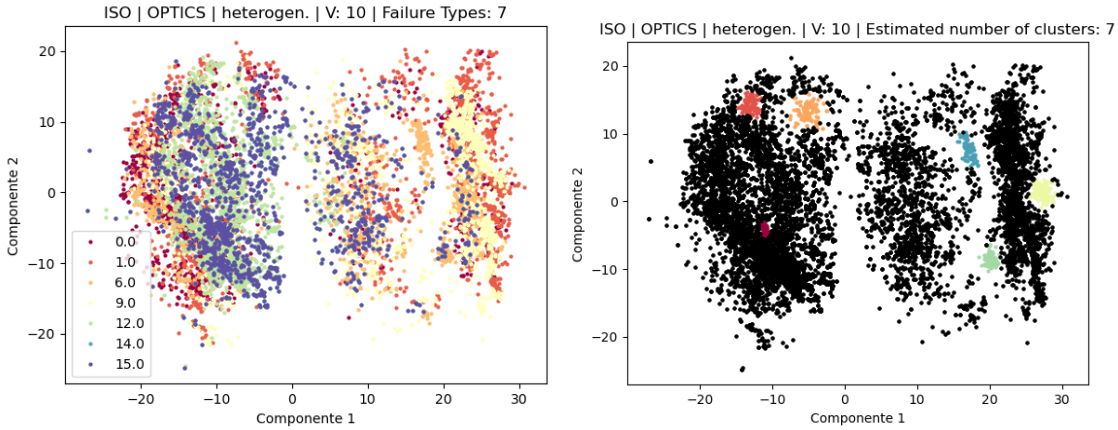


Figura 66 Resultado del análisis concatenando los fallos entrenando los métodos con la serie sin fallos para ISOMAP y OPTICS por tipo de fallo.

Como se ha comentado en la explicación de la prueba todos los fallos se parecen demasiado al verse influidos por la serie sin fallo, por lo que es imposible llevar a cabo cualquier clasificación. Además, los algoritmos de agrupamiento no han sido capaces de generar grupos que permitan alguna distinción.

4.4.2.2.2 Entrenamiento y aplicación con todas las series de datos

Las siguientes pruebas que se realizarán se dividen en tres apartados. En el primero y en el segundo se crea un vector que incluye los datos de todos los tipos de fallo, es decir una matriz que como columnas tiene las variables de sistema, y que las filas son los datos de *sin Fallo, Fallo 1, Fallo 2, etc.* La diferencia entre ambas pruebas es que en la primera los parámetros de los algoritmos de agrupación se eligen con el índice máximo absoluto de heterogeneidad, mientras que para la segunda prueba se elige el índice máximo con la condición de que se tienen que formar entre 5 y 7 grupos. Para la última prueba se aplica la condición del índice de la segunda, pero en este caso se analizará creando la matriz por tipo de fallo con una única intensidad, es decir una matriz que contiene las series *sin Fallo, Fallo 1, Fallo 6, Fallo 9, Fallo 12, Fallo 14 y Fallo 15.* Además, se utilizarán los fallos que han quedado fuera de la matriz para comprobar si se pueden clasificar. Una vez se han obtenido la matriz en cuestión, se realiza el entrenamiento, aplicando o no PCA para la reducción de variables según se escoja, utilizando los métodos LLE o ISOMAP, y al resultado de esta transformación se realizan los algoritmos de densidad. El proceso seguido puede analizarse en el Algoritmo 8 y la Figura 67.

Con el objetivo de evitar la repetición de resultados se mostrará únicamente los resultados de DBSCAN para LLE e ISOMAP con y sin PCA (Tablas 17 a 20 respectivamente y sus respectivas gráficas). Debido a que las transformaciones serán las

mismas y únicamente cambia la forma en que se crean las agrupaciones las cuales no varía demasiado entre los algoritmos. No obstante, se añadirá una tabla con los parámetros de todas las pruebas.

tipoTransformacion = “LLE” o “ISOMAP”

sinPCA = True o False

S = Todos los fallos incluido la serie sin fallo

Normalizar *S* con media 0 y varianza 1

Entrenar los datos atendiendo a si deben procesarse con PCA

Si *tipoTransformacion LLE* Entonces

    Entrenar Transformación LLE con S

    Aplicar Transformación LLE a S

Sino Entonces

    Entrenar Transformación ISOMAP con S

    Aplicar Transformación ISOMAP a S

Si DBSCAN

    Para los parámetros eps y min_samples

        Realizar DBSCAN

        Analizar DBSCAN

        Realizar Cluster Plot

    Fin Para

Si HDBSCAN

    Para los parámetros min_cluster

        Realizar HDBSCAN

        Analizar HDBSCAN

        Realizar Cluster Plot

    Fin Para

Si OPTICS

    Para los parámetros min_samples, xi y min_cluster

Realizar HDBSCAN

Analizar HDBSCAN

Realizar Cluster Plot

Fin Para

Fin Para

Algoritmo 8 Algoritmo para el análisis de los fallos concatenados.

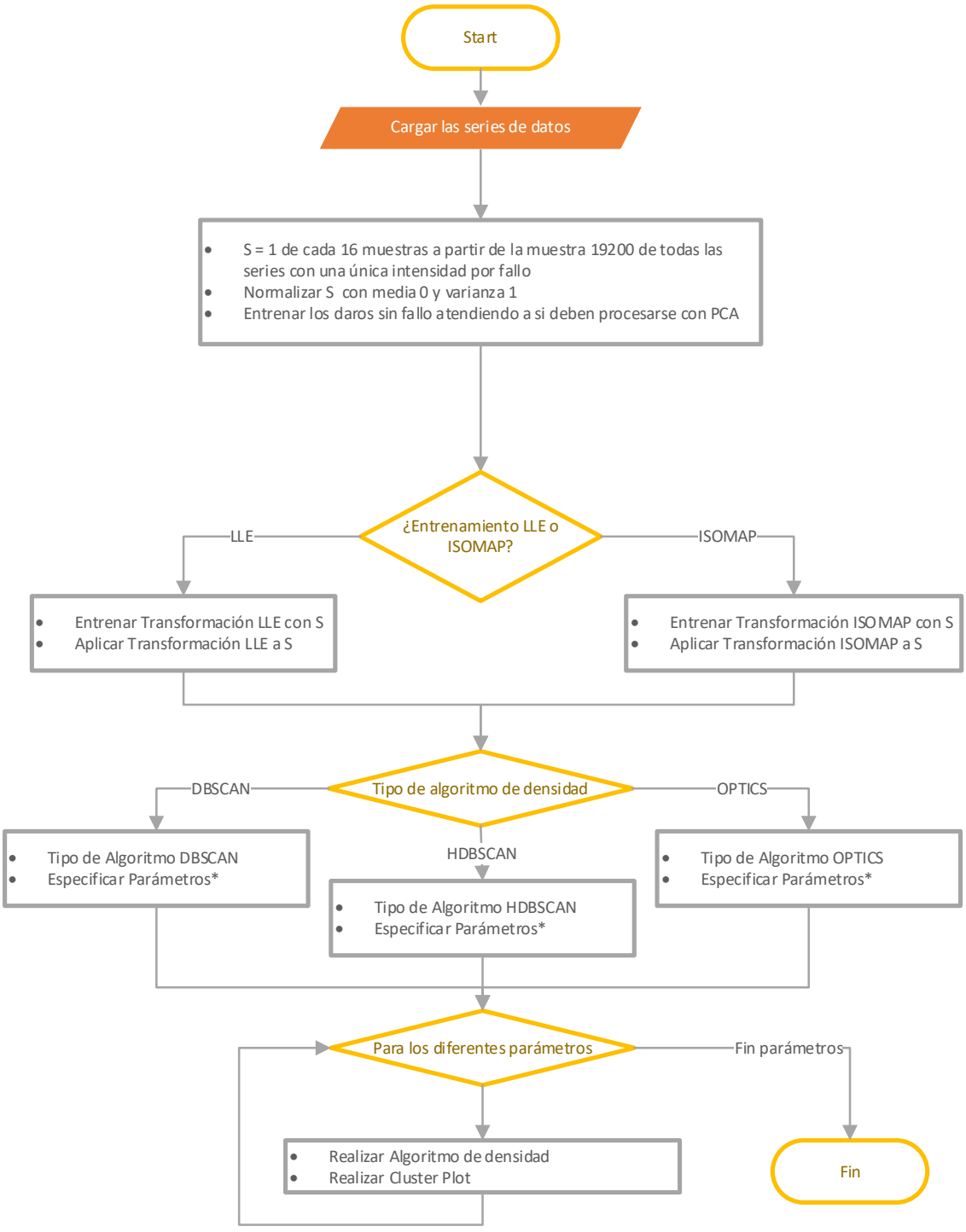


Figura 67 Proceso para el análisis de los fallos concatenados.

En el siguiente caso la matriz que se forma para la simulación contiene todos los datos de todos los tipos de fallo. Además, se han escogido los parámetros de los algoritmos de densidad buscando el índice de heterogeneidad más alto, independientemente del resto de indicadores, y donde el rango de valores donde se van a buscar los parámetros óptimos de los algoritmos de agrupamiento aparece en la Tabla 16.

Tabla 16 Parámetros para los resultados con el índice máximo de heterogeneidad.

	PCA	*manifold*	Eps	minPoints	ε
DBSCAN	No	LLE	0.0004,0.0008,0.004	15,20,25	-
		ISOMAP	0.5,0.25,0.1	45,50,55	-
	Sí	LLE	0.0004,0.0008,0.004	15,20,25	-
		ISOMAP	0.5,0.25,0.1	50,55,60	-
HDBSCAN	No	LLE	-	2000,2500,3000	-
		ISOMAP	-	30,35,40	-
	Sí	LLE	-	3500,4000,4500	-
		ISOMAP	-	50,100,150	-
OPTICS	No	LLE	30,40,50	12,15,18	0.002,0.005,0.008
		ISOMAP	30,40,50	0.1,0.01,0.001	0.1,0.01,0.001
	Sí	LLE	30,40,50	12,15,18	0.002,0.005,0.008
		ISOMAP	30,40,50	0.1,0.01,0.001	0.1,0.01,0.001

Tabla 17 Resultado de la transformación de todos los fallos concatenados para LLE, DBSCAN y sin PCA.

	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
DBSCAN(0,0004,15)	3,00	127,00	0,96	0,49	879745,66
DBSCAN(0,0004,20)	3,00	133,00	0,96	0,50	857165,04
DBSCAN(0,0004,25)	3,00	133,00	0,96	0,50	857165,04
DBSCAN(0,0008,15)	4,00	91,00	0,95	0,47	798571,21
DBSCAN(0,0008,20)	3,00	107,00	0,96	0,44	973497,54
DBSCAN(0,0008,25)	3,00	108,00	0,96	0,44	969308,12
DBSCAN(0,004,15)	2,00	0,00	0,82	0,30	23639,88
DBSCAN(0,004,20)	2,00	0,00	0,82	0,30	23639,88
DBSCAN(0,004,25)	2,00	0,00	0,82	0,30	23639,88

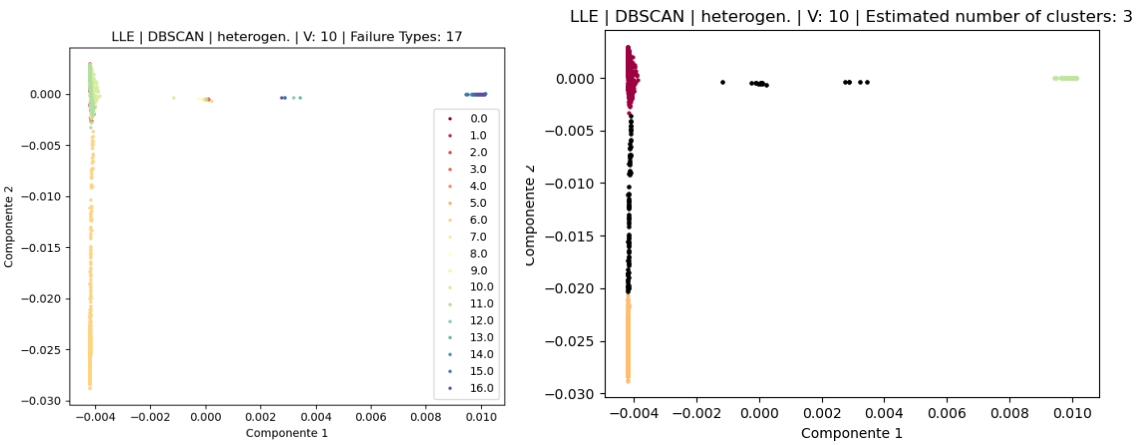
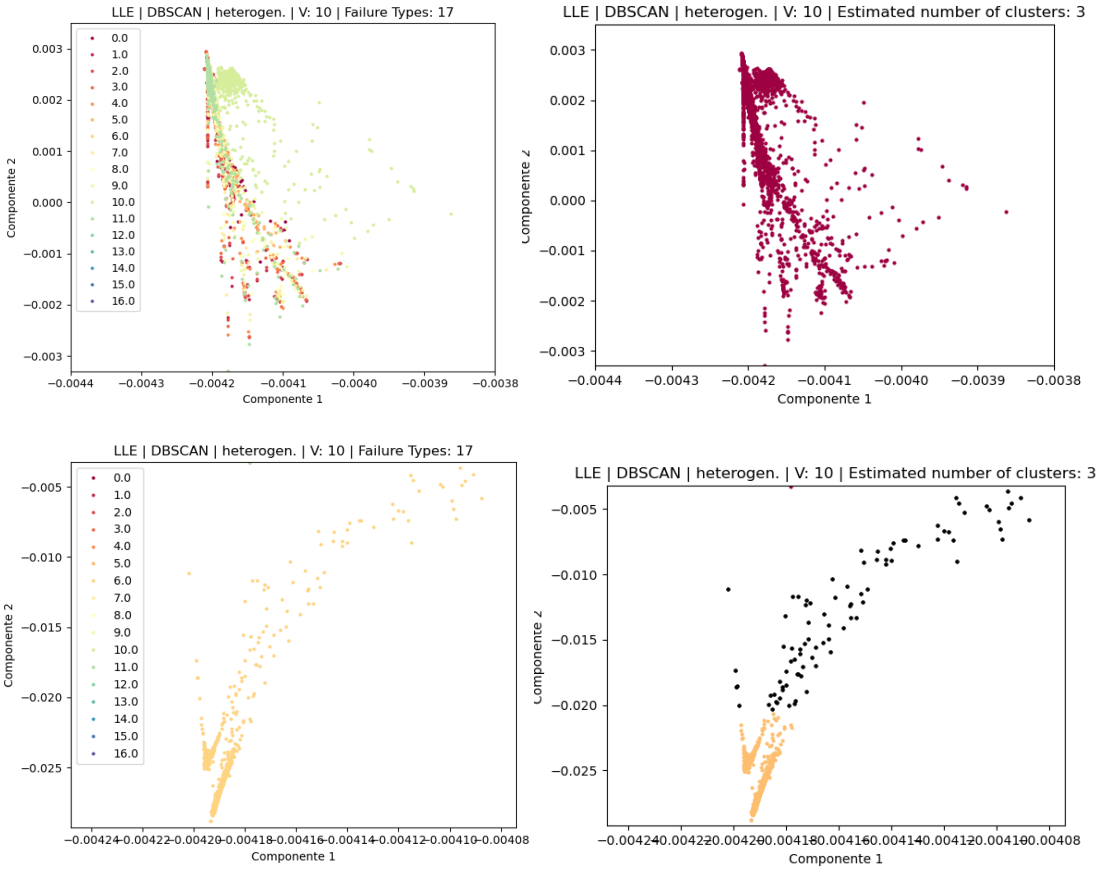


Figura 68 Resultado de la transformación de todos los fallos concatenados para LLE, DBSCAN y sin PCA.



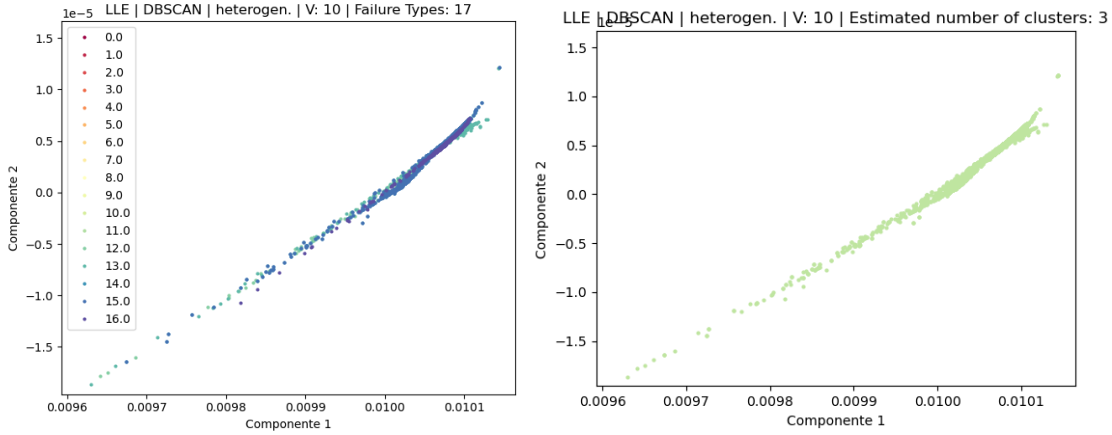


Figura 69 Ampliaciones de la Figura 68.

Tabla 18 Resultado de la transformación de todos los fallos concatenados para LLE, DBSCAN y con PCA.

	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
DBSCAN(0,0004,15)	3,00	102,00	0,96	0,39	1170872,58
DBSCAN(0,0004,20)	3,00	108,00	0,96	0,41	1133301,85
DBSCAN(0,0004,25)	3,00	111,00	0,96	0,41	1119134,04
DBSCAN(0,0008,15)	4,00	71,00	0,96	0,41	1002868,73
DBSCAN(0,0008,20)	3,00	89,00	0,96	0,36	1239683,68
DBSCAN(0,0008,25)	3,00	92,00	0,96	0,36	1226134,71
DBSCAN(0,004,15)	2,00	0,00	0,80	0,36	16704,18
DBSCAN(0,004,20)	2,00	0,00	0,80	0,36	16704,18
DBSCAN(0,004,25)	2,00	0,00	0,80	0,36	16704,18

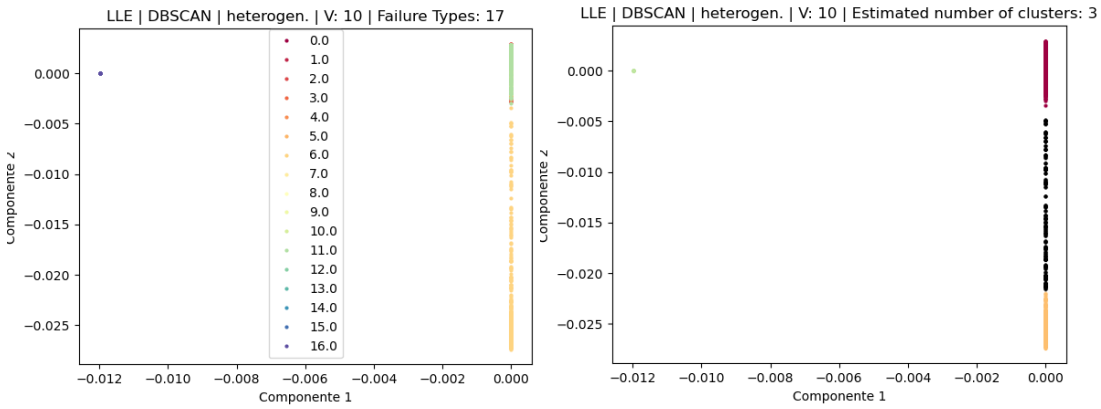


Figura 70 Resultado de la transformación de todos los fallos concatenados para LLE, DBSCAN y con PCA.

Tabla 19 Resultado de la transformación de todos los fallos concatenados para ISOMAP, DBSCAN y sin PCA.

	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
DBSCAN(0,5,45)	12,00	10549,00	-0,08	1,00	22833,16
DBSCAN(0,5,50)	11,00	11473,00	-0,04	1,10	21680,30
DBSCAN(0,5,55)	9,00	12288,00	-0,04	1,08	23024,25
DBSCAN(0,25,45)	2,00	17689,00	0,61	0,35	39227,78
DBSCAN(0,25,50)	1,00	17909,00	0,74	0,24	62203,70
DBSCAN(0,25,55)	2,00	18027,00	0,57	0,34	28547,58
DBSCAN(0,1,45)	13,00	21449,00	0,30	0,48	533,80
DBSCAN(0,1,50)	20,00	21842,00	0,27	0,53	261,83
DBSCAN(0,1,55)	10,00	22646,00	0,22	0,56	254,54

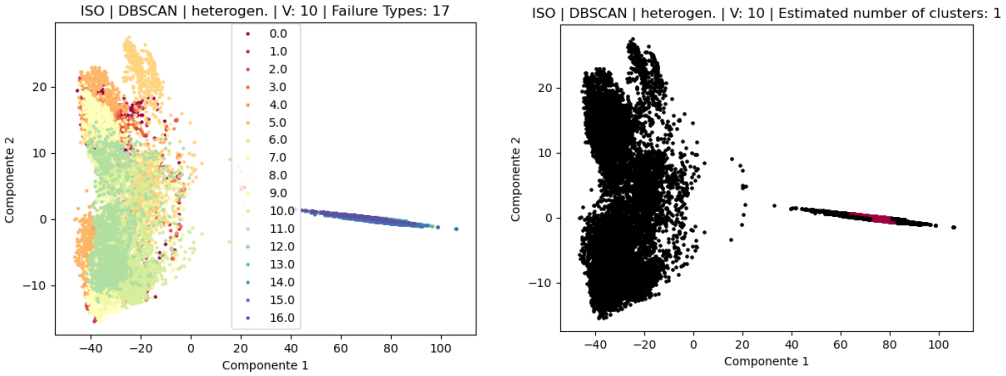


Figura 71 Resultado de la transformación de todos los fallos concatenados para ISOMAP, DBSCAN y sin PCA.

Tabla 20 Resultado de la transformación de todos los fallos concatenados para ISOMAP, DBSCAN y con PCA.

Fallo 0	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
DBSCAN(0,5,50)	5,00	9441,00	0,05	3,12	1817,23
DBSCAN(0,5,55)	6,00	10034,00	0,00	2,81	1323,27
DBSCAN(0,5,60)	4,00	10738,00	0,16	3,16	1685,85
DBSCAN(0,25,50)	2,00	16758,00	-0,08	2,76	266,19
DBSCAN(0,25,55)	1,00	16836,00	0,14	3,42	487,21
DBSCAN(0,25,60)	1,00	16844,00	0,14	3,41	487,71
DBSCAN(0,1,50)	2,00	16956,00	-0,07	6,41	246,28
DBSCAN(0,1,55)	1,00	17012,00	0,13	3,33	492,36
DBSCAN(0,1,60)	1,00	17014,00	0,13	3,33	492,17

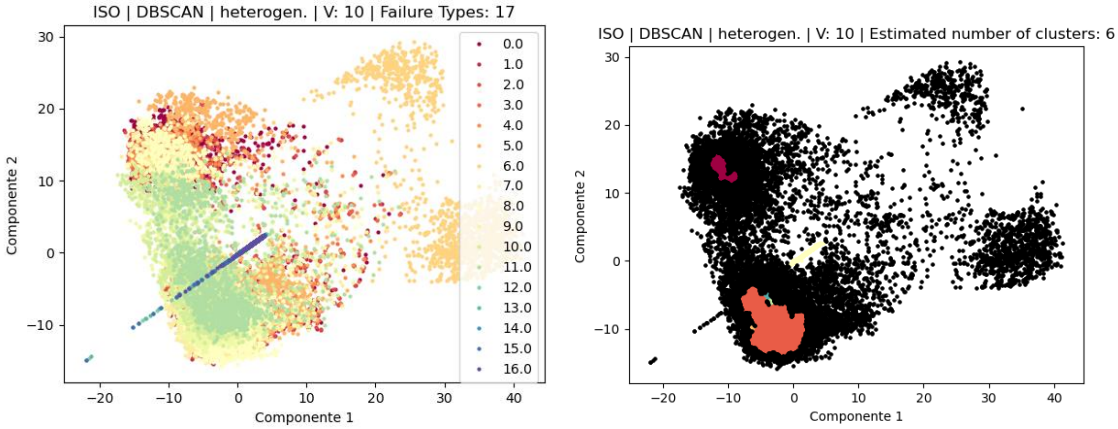


Figura 72 Resultado de la transformación de todos los fallos concatenados para ISOMAP, DBSCAN y con PCA.

Se observa en todos los métodos de agrupación que un mayor índice de heterogeneidad no implica un buen resultado para la clasificación de fallos. Dado que la mayoría de los resultados convergen a una solución de pocos grupos, con una heterogeneidad alta, pero que dejan sin clasificar la mayoría de los puntos. Por lo tanto, es necesario encontrar cierto equilibrio entre la heterogeneidad y el número de grupos seleccionado. En cuanto a las transformaciones *manifold* puede apreciarse una posible clasificación entre dos grupos de fallos que se discute más adelante

Para mejorar esta solución, ahora se realiza una segunda prueba donde la matriz de datos que se forma para la simulación contiene todos los datos de todos los fallos al igual que en la primera. No obstante, se han escogido los parámetros de los algoritmos de densidad buscando un índice de heterogeneidad más alto, teniendo en cuenta el número de grupos formado intentando que cada grupo identifique un fallo distinto, es decir 6 grupos (7 si incluimos la serie sin fallo). Como antes el rango de variación de los parámetros de cada algoritmo de clúster se ve en la Tabla 21.

Tabla 21 Parámetros para los resultados con el índice máximo de heterogeneidad condicionado.

	PCA	*manifold*	Eps	minPoints	ε
DBSCAN	No	LLE	0.0008,0.0009,0.001	11,12,13	-
		ISOMAP	0.059,0.06,0.061	34,35,36	-
	Sí	LLE	0.0002,0.0003,0.0004	6,7,8	-
		ISOMAP	0.015,0.02,0.025	20,25,30	-
HDBSCAN	No	LLE	-	700,705,710	-
		ISOMAP	-	40,50,60	-
	Sí	LLE	-	810,820,830	-
		ISOMAP	-	55,60,65	-
OPTICS	No	LLE	40,50,60	0.1,0.12,0.14	0.001,0.002,0.003
		ISOMAP	38,40,42	0.1,0.105,0.11	0.0001,0.0004,0.0007
	Sí	LLE	40,50,60	0.1,0.12,0.14	0.001,0.002,0.003
		ISOMAP	23,25,27	0.1,0.105,0.11	0.0001,0.0005,0.001

Como en la prueba anterior y con el objetivo de evitar la repetición de resultados se mostrará únicamente los resultados de DBSCAN para LLE e ISOMAP con y sin PCA (Tablas 22 a 25).

Tabla 22 Resultados con el índice máximo de heterogeneidad condicionado con LLE y sin PCA.

	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
DBSCAN(0,0008,11)	7,00	27,00	0,91	1,04	500405,38
DBSCAN(0,0008,12)	5,00	64,00	0,95	0,70	650729,85
DBSCAN(0,0008,13)	5,00	77,00	0,94	0,92	642818,87
DBSCAN(0,0009,11)	7,00	15,00	0,91	0,80	508302,01
DBSCAN(0,0009,12)	6,00	32,00	0,94	0,76	572250,54
DBSCAN(0,0009,13)	6,00	44,00	0,94	1,45	547223,63
DBSCAN(0,001,11)	6,00	12,00	0,91	0,85	542629,18
DBSCAN(0,001,12)	7,00	15,00	0,91	0,99	503599,34
DBSCAN(0,001,13)	6,00	32,00	0,93	0,63	594317,27

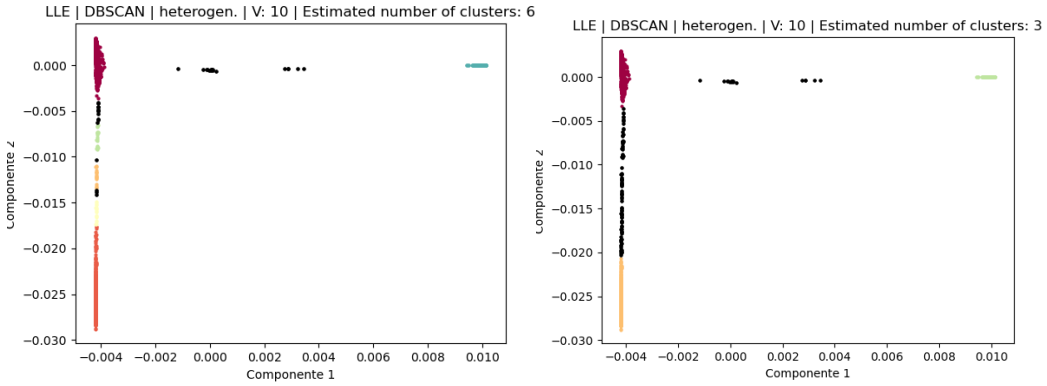


Figura 73 Resultados con el índice máximo de heterogeneidad condicionado con LLE y sin PCA.

Tabla 23 Resultados con el índice máximo de heterogeneidad condicionado con LLE y con PCA.

	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
DBSCAN(0,0002,6)	6,00	77,00	0,95	0,47	661754,42
DBSCAN(0,0002,7)	7,00	80,00	0,95	0,49	559088,68
DBSCAN(0,0002,8)	5,00	95,00	0,95	0,47	730410,64
DBSCAN(0,0003,6)	6,00	67,00	0,95	0,48	679778,74
DBSCAN(0,0003,7)	6,00	68,00	0,95	0,49	675943,73
DBSCAN(0,0003,8)	5,00	75,00	0,95	0,45	792632,18
DBSCAN(0,0004,6)	10,00	30,00	0,95	6,73	450236,39
DBSCAN(0,0004,7)	6,00	57,00	0,95	0,72	696898,48
DBSCAN(0,0004,8)	5,00	64,00	0,95	0,45	831756,79

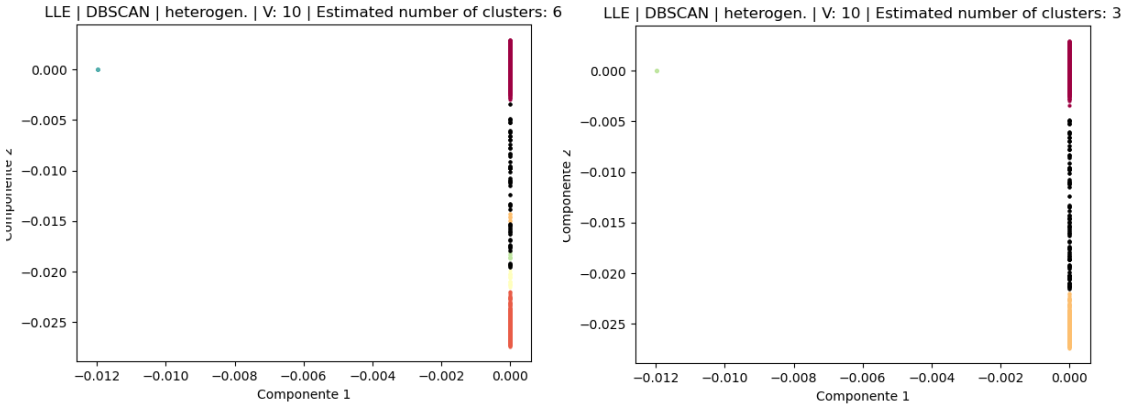


Figura 74 Resultados con el índice máximo de heterogeneidad condicionado con LLE y con PCA.

Tabla 24 Resultados con el índice máximo de heterogeneidad condicionado con ISMOAP y sin PCA.

Fallo 0	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
DBSCAN(0,059,34)	8,00	23274,00	0,16	0,60	111,43
DBSCAN(0,059,35)	4,00	23419,00	0,16	0,62	133,28
DBSCAN(0,059,36)	3,00	23478,00	0,15	0,62	132,87
DBSCAN(0,06,34)	8,00	23260,00	0,16	0,60	115,58
DBSCAN(0,06,35)	6,00	23344,00	0,16	0,61	117,82
DBSCAN(0,06,36)	5,00	23396,00	0,16	0,61	117,23
DBSCAN(0,061,34)	10,00	23180,00	0,17	0,60	111,43
DBSCAN(0,061,35)	6,00	23341,00	0,16	0,61	119,00
DBSCAN(0,061,36)	5,00	23391,00	0,16	0,61	119,53

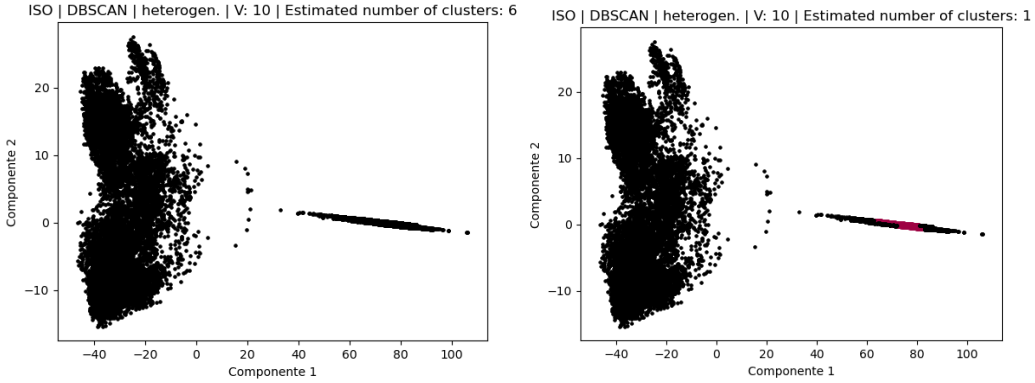


Figura 75 Resultados con el índice máximo de heterogeneidad condicionado con ISOMAP y sin PCA.

Tabla 25 Resultados con el índice máximo de heterogeneidad condicionado con ISOMAP y con PCA.

	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
DBSCAN(0,015,20)	8,00	18049,00	-0,12	4,74	57,73
DBSCAN(0,015,25)	12,00	18376,00	-0,19	3,66	37,08
DBSCAN(0,015,30)	15,00	18891,00	-0,21	3,75	26,90
DBSCAN(0,02,20)	6,00	17408,00	-0,12	5,19	80,75
DBSCAN(0,02,25)	6,00	17628,00	-0,21	4,30	79,68
DBSCAN(0,02,30)	2,00	17885,00	-0,13	4,26	231,01
DBSCAN(0,025,20)	6,00	17168,00	-0,11	6,08	82,00
DBSCAN(0,025,25)	5,00	17316,00	-0,11	5,24	97,56
DBSCAN(0,025,30)	4,00	17431,00	-0,21	4,54	121,26

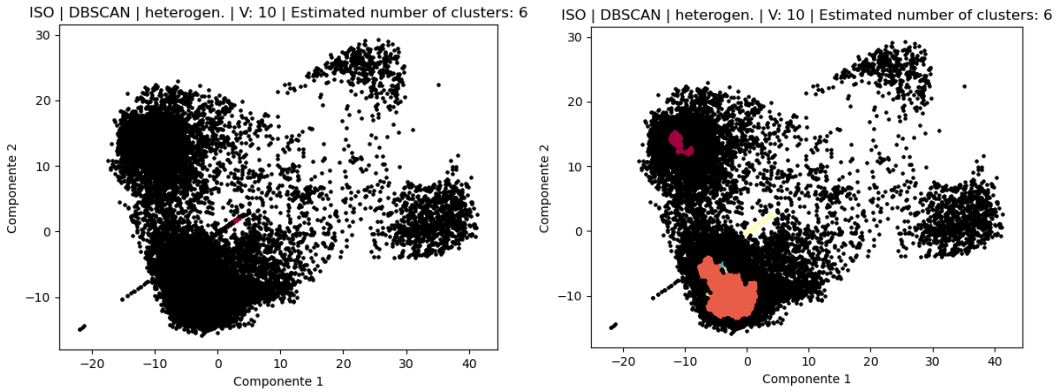


Figura 76 Resultados con el índice máximo de heterogeneidad condicionado con ISOMAP y con PCA.

A pesar de haber conseguido un número de grupos que nos permitan una clasificación correcta, no ha sido posible lograrla para todos los tipos de fallo. Este problema no es debido al mal funcionamiento del algoritmo el cual es capaz de realizar agrupaciones. Sino a los fallos y sus transformaciones *manifold* que no logran separarlos como sería óptimo para realizar una buena clasificación. La transformación LLE deja todos los puntos de los fallos y los clústeres muy juntos formando casi una línea y muy cercana a cero, lo que indica que este método no funciona adecuadamente cuando concatenamos todos los fallos en una sola matriz de datos, es un conjunto demasiado de datos para este tipo de transformación. Por el contrario, la transformación ISOMAP se expande por el espacio de los datos, pero no obtiene unos buenos agrupamientos, ya que la mayoría de los puntos se quedan sin clasificar.

Intentando mejorar los resultados, y siguiendo con el mismo tipo de análisis de los dos apartados anteriores, esta vez en lugar de utilizar todas las series de datos, se utilizará una serie de cada tipo de fallo y la serie sin fallo es decir las series *sin Fallo, Fallo 1, Fallo 6, Fallo 9, Fallo 12, Fallo 14 y Fallo 15.* Para intentar utilizar menos datos, y que ahora los algoritmos traten de obtener clústeres diferentes para cada fallo, y utilizar los fallos no entrenados como prueba, es decir pasar los datos nuevos y ver a que clúster de todos los calculados se parece más, y si es posible por tanto clasificar los fallos.

Los algoritmos de agrupamiento dependen de unos parámetros, como no se sabe cuál son los parámetros óptimos, se van a variar en un rango de valores definidos en la Tabla 26. Como en las pruebas anteriores y con el objetivo de evitar la repetición de resultados se mostrará únicamente los resultados de DBSCAN para LLE e ISOMAP con y sin PCA (Tablas 27 a 30).

Tabla 26 Parámetros para los resultados con el índice máximo de heterogeneidad condicionado por tipos de fallo.

	PCA	*manifold*	Eps	minPoints	ε
DBSCAN	No	LLE	0.00015,0.0002,0.00025	6,7,8	-
		ISOMAP	0.25,0.5,0.75	40,45,50	-
	Sí	LLE	0.00015,0.0002,0.00025	6,7,8	-
		ISOMAP	0.25,0.5,0.75	40,45,50	-
HDBSCAN	No	LLE	-	100,125,150	-
		ISOMAP	-	100,125,150	-
	Sí	LLE	-	178,180,182	-
		ISOMAP	-	125,150,175	-
OPTICS	No	LLE	50,60,70	0.1,0.11,0.12	0.0001,0.0004,0.0007
		ISOMAP	10,20,30	0.1,0.01,0.001	0.1,0.08,0.06
	Sí	LLE	50,60,70	0.1,0.11,0.12	0.0001,0.0004,0.0007
		ISOMAP	10,20,30	0.1,0.01,0.001	0.1,0.08,0.06

Tabla 27 Resultados con el índice máximo de heterogeneidad condicionado para tipo de fallo con LLE y sin PCA.

	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
DBSCAN(0,00015,6)	7,00	64,00	0,95	1,20	296334,40
DBSCAN(0,00015,7)	8,00	70,00	0,95	1,24	237506,76
DBSCAN(0,00015,8)	4,00	105,00	0,95	0,71	407387,25
DBSCAN(0,0002,6)	8,00	35,00	0,96	1,42	431422,00
DBSCAN(0,0002,7)	6,00	55,00	0,96	1,67	495882,06
DBSCAN(0,0002,8)	7,00	55,00	0,96	1,58	430750,76
DBSCAN(0,00025,6)	8,00	24,00	0,96	1,08	487546,48
DBSCAN(0,00025,7)	6,00	42,00	0,97	1,73	539313,23
DBSCAN(0,00025,8)	5,00	49,00	0,97	1,65	639427,94

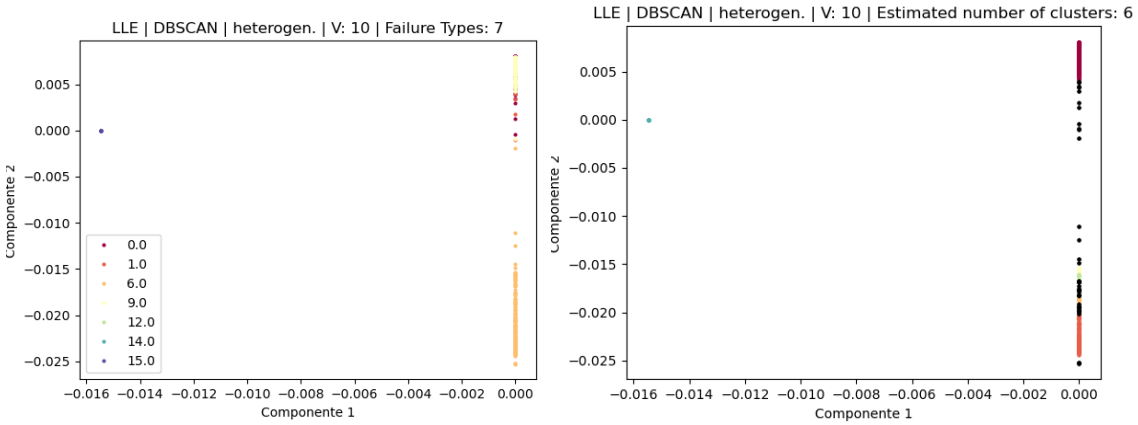


Figura 77 Resultados con el índice máximo de heterogeneidad condicionado para tipo de fallo con LLE y sin PCA.

Tabla 28 Resultados con el índice máximo de heterogeneidad condicionado para tipo de fallo con LLE y con PCA.

	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
DBSCAN(0,00015,6)	7,00	65,00	0,95	1,29	308196,58
DBSCAN(0,00015,7)	6,00	71,00	0,95	0,85	357105,36
DBSCAN(0,00015,8)	6,00	80,00	0,94	0,90	331698,22
DBSCAN(0,0002,6)	8,00	43,00	0,91	1,10	452253,22
DBSCAN(0,0002,7)	7,00	52,00	0,95	1,10	358177,99
DBSCAN(0,0002,8)	7,00	56,00	0,95	1,17	342845,82
DBSCAN(0,00025,6)	5,00	36,00	0,97	1,28	840663,42
DBSCAN(0,00025,7)	6,00	41,00	0,93	1,20	642542,52
DBSCAN(0,00025,8)	5,00	49,00	0,96	1,14	527320,88

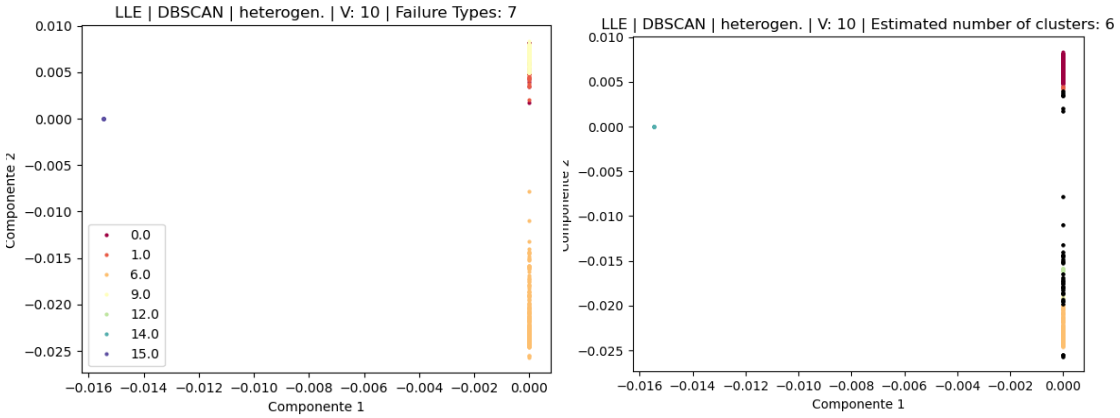


Figura 78 Resultados con el índice máximo de heterogeneidad condicionado para tipo de fallo con LLE y con PCA.

Tabla 29 Resultados con el índice máximo de heterogeneidad condicionado para tipo de fallo con ISOMAP y sin PCA.

	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz

DBSCAN(0,75,40)	7,00	425,00	0,40	1,57	2603,92
DBSCAN(0,75,45)	7,00	546,00	0,44	1,35	4242,10
DBSCAN(0,75,50)	8,00	732,00	0,50	1,35	7538,78
DBSCAN(0,75,55)	7,00	886,00	0,49	1,33	7872,72
DBSCAN(0,75,60)	7,00	1029,00	0,46	1,08	7394,55
DBSCAN(0,5,40)	8,00	1399,00	0,40	1,18	5621,23
DBSCAN(0,5,45)	6,00	1616,00	0,44	1,18	6389,81
DBSCAN(0,5,50)	5,00	1786,00	0,48	1,14	7189,77
DBSCAN(0,5,55)	5,00	1860,00	0,47	1,16	6900,01
DBSCAN(0,5,60)	6,00	1936,00	0,44	1,09	5587,70
DBSCAN(0,25,40)	15,00	3629,00	0,03	1,33	902,75
DBSCAN(0,25,45)	15,00	4221,00	-0,05	1,41	629,90
DBSCAN(0,25,50)	16,00	4723,00	-0,08	1,37	464,37
DBSCAN(0,25,55)	14,00	5158,00	-0,10	1,89	426,89
DBSCAN(0,25,60)	15,00	5620,00	-0,21	2,07	334,75

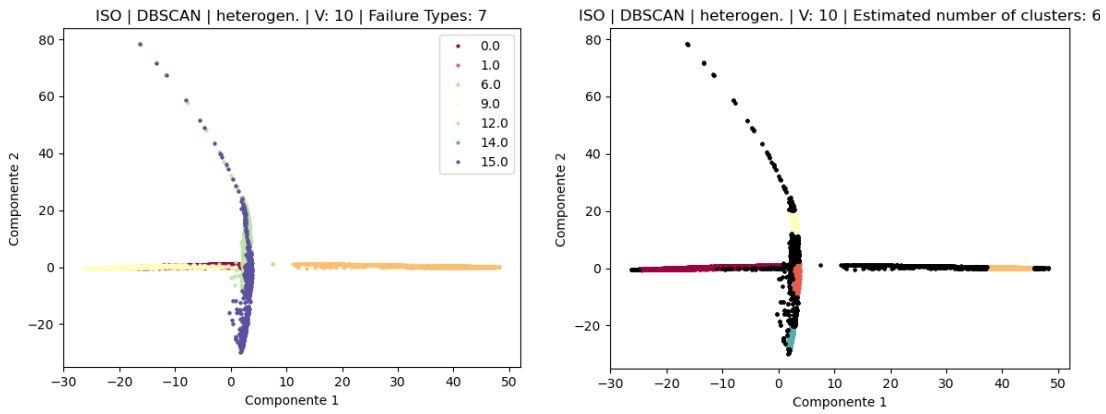


Figura 79 Resultados con el índice máximo de heterogeneidad condicionado para tipo de fallo con ISOMAP y sin PCA.

Tabla 30 Resultados con el índice máximo de heterogeneidad condicionado para tipo de fallo con ISOMAP y con PCA.

Fallo 0	Nº Grupos	Nº P. de ruido	Silhouette Coef.	Davies-Bouldin	Calinski-Harabasz
DBSCAN(0,75,40)	7,00	452,00	0,42	1,55	2709,42
DBSCAN(0,75,45)	5,00	665,00	0,40	2,45	3638,53
DBSCAN(0,75,50)	4,00	787,00	0,45	0,81	6914,53
DBSCAN(0,75,55)	4,00	822,00	0,45	0,83	6811,81
DBSCAN(0,75,60)	6,00	865,00	0,50	0,98	10108,60
DBSCAN(0,5,40)	7,00	1213,00	0,42	1,16	7209,45
DBSCAN(0,5,45)	5,00	1450,00	0,52	1,19	8978,74
DBSCAN(0,5,50)	5,00	1580,00	0,51	1,20	8513,92
DBSCAN(0,5,55)	6,00	1751,00	0,43	1,22	6409,81
DBSCAN(0,5,60)	8,00	1916,00	0,31	1,18	4613,13
DBSCAN(0,25,40)	11,00	3755,00	0,04	1,18	1077,21

DBSCAN(0,25,45)	11,00	4157,00	0,04	1,46	884,73
DBSCAN(0,25,50)	12,00	4497,00	-0,12	1,45	708,44
DBSCAN(0,25,55)	9,00	5052,00	-0,03	1,58	775,21
DBSCAN(0,25,60)	9,00	5372,00	-0,08	1,61	699,46

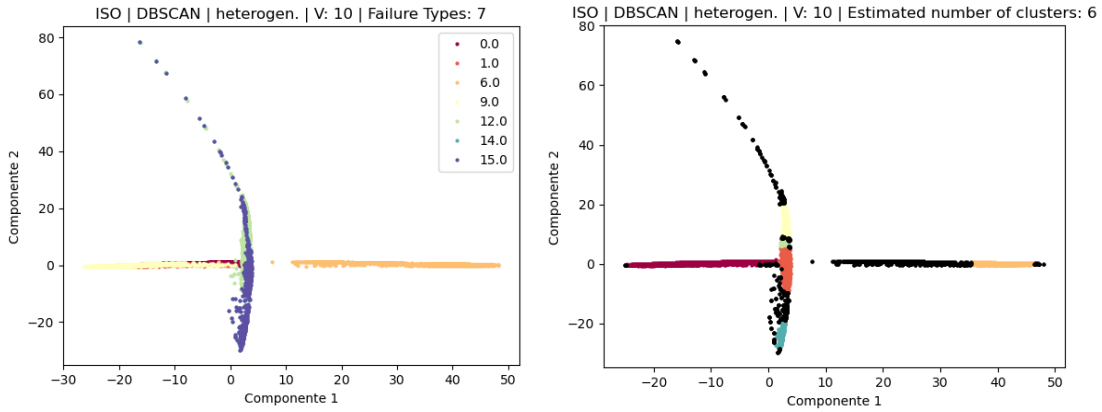


Figura 80 Resultados con el índice máximo de heterogeneidad condicionado para tipo de fallo con ISOMAP y con PCA.

Analizando las figuras anteriores, para LLE e ISOMAP respectivamente, se aprecia como varios tipos de fallo coinciden, lo que no nos permite una buena clasificación de los grupos. Se ve más claramente en la transformación ISOMAP de la Figura 80 donde existe una clara diferenciación entre el fallo 6 que forma su propio grupo, y los fallos 12, 14 y 15 que formarían otro conjunto, dejando para el último conjunto la serie sin fallo y los fallos 1 y 9. Como en la prueba anterior LLE deja todos los puntos de los fallos y los clústeres muy juntos formando casi una línea y muy cercana a cero, lo que indica que este método no funciona adecuadamente cuando concatenamos todos los fallos en una sola matriz de datos, es un conjunto demasiado de datos para este tipo de transformación.

Aplicando este método para los datos que se tratan se podría al menos realizar una clasificación del nuevo fallo en la que se descartarían la mitad de los posibles fallos. Se podría dividir la clasificación en fallos del 1 al 11 y fallos del 12 al 16. Aunque no se trata de una clasificación total y completa, permitiría cierta ventaja al tratar de resolver y analizarlos problemas que pudieran aparecer en la planta. Para comprobarlo se va a utilizar la transformación de la matriz de fallos para aplicársela a un nuevo fallo que se quiera identificar, el fallo 3 en este caso. Tras aplicarle la transformación ISOMAP se realizarán las agrupaciones y se mostrarán los resultados (Figura 81).

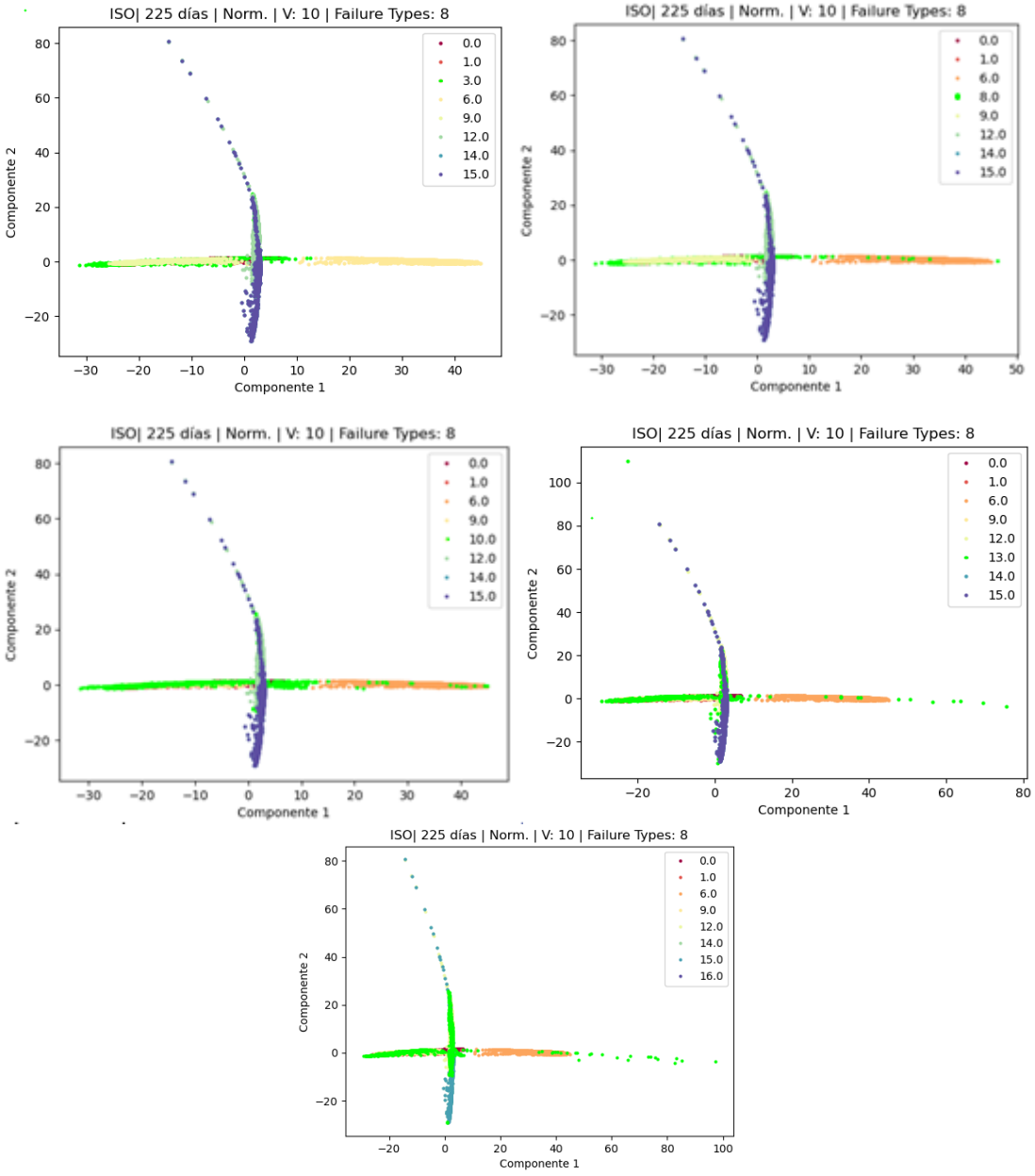


Figura 81 Resultado de la transformación con los fallos a identificar para 225 días de datos. Cada gráfica contiene un fallo distinto que se trata de clasificar en color verde fosforito.

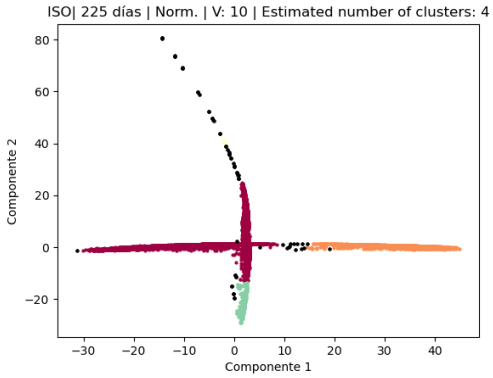


Figura 82 Resultados de agrupamiento para 225 días. Representa de forma general los agrupamientos de los datos de la Figura 82.

Como se puede observar en la Figura 81 los fallos del 1 al 11 tienen menos datos en la componente vertical que los del grupo 12 al 16. No obstante es complicado lograr alguna clasificación con estos datos, dado que se parecen demasiado entre ellos, además los algoritmos de agrupamiento no han conseguido ninguna clasificación positiva (Figura 82). A raíz de los resultados anteriores, en los cuales se ha estado cerca de conseguir al menos una clasificación parcial. Se ha decidido repetir esta última prueba, pero utilizando 75 días de datos en lugar de 225 días. Esta decisión se lleva a cabo teniendo en cuenta los resultados observados en el apartado 4.4.1.3 donde se ha llegado a la conclusión de que los fallos son lo más diferenciables entre sí cuando alcanzan los 75 días de datos.

Análisis adicional con 75 días de datos

Antes de realizar las pruebas hay que tener en cuenta los resultados obtenidos anteriormente por lo tanto únicamente se mostrarán los resultados para ISOMAP. Hay que recordar que el proceso es el mismo que el de la Figura 67 pero utilizando 75 días para los datos. Es decir, se crea la matriz de datos con los datos de fallo concatenados (incluido el fallo 7), y se realiza la trasformación ISOMAP sobre esa matriz (Figura 81). Después se verá cómo responde esta transformación, cuando pasamos unos datos nuevos (un nuevo fallo) para ver donde se colocan esos datos en las gráficas, y ver si se puede distinguir el fallo o no.

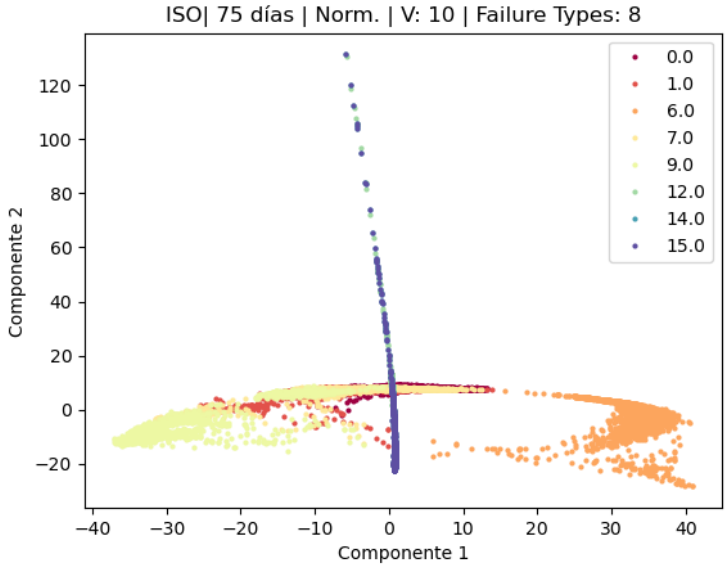
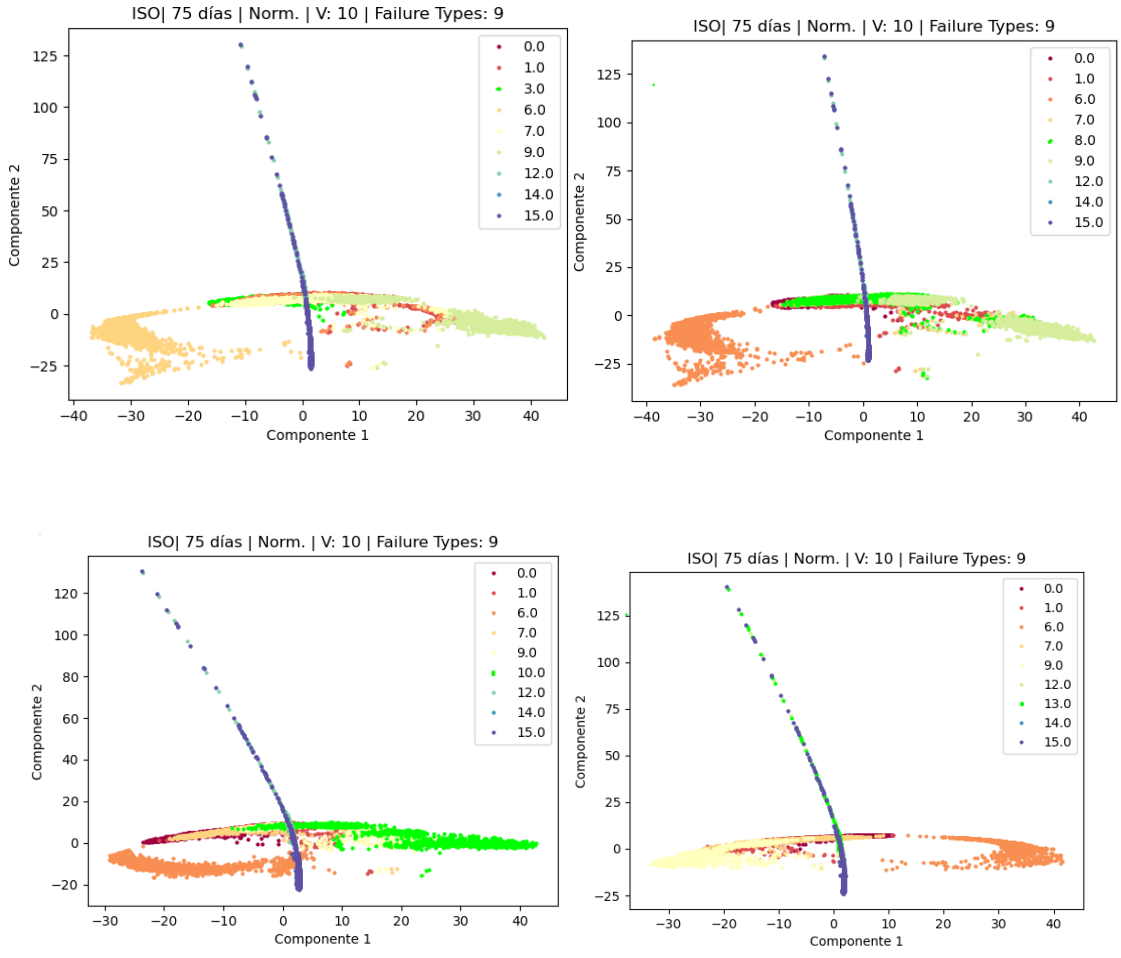


Figura 83 Resultado con sólo los tipos de fallos.

Se aprecia en la Figura 83 una silueta similar a las obtenidas en el apartado anterior. Los datos de fallo 6 crean su propio grupo, luego los fallos 12, 14 y 15 crean otro grupo y finalmente los datos de los fallos 0, 1 7 y 9 están solapados.

Ahora vamos a pasar datos de fallo diferentes de los usados para entrenamiento, para ver donde se colocan sus datos (Figura 84) donde se pasan datos de fallo 3, 8, 10, 13 y 15 respectivamente, y el resultado se muestra en la gráfica (en verde son los datos de los nuevos fallos).

Analizando la Figura 84 puede llegarse a la misma conclusión que en el apartado 4.4.1.3, Es posible que se pueda realizar una clasificación parcial de los fallos en dos grupos, por un lado los fallos del 1 al 11 y por otro lado los fallos del 12 al 16. De la misma manera no podemos utilizar el método para detectar el fallo, ya que pueden coincidir con la serie sin fallo. Si se analiza con más detalle puede observarse que los fallos del 1 al 11 únicamente aparecen en el eje horizontal, mientras que los fallos del 12 al 16 pueden aparecer en el eje vertical o en ambos.



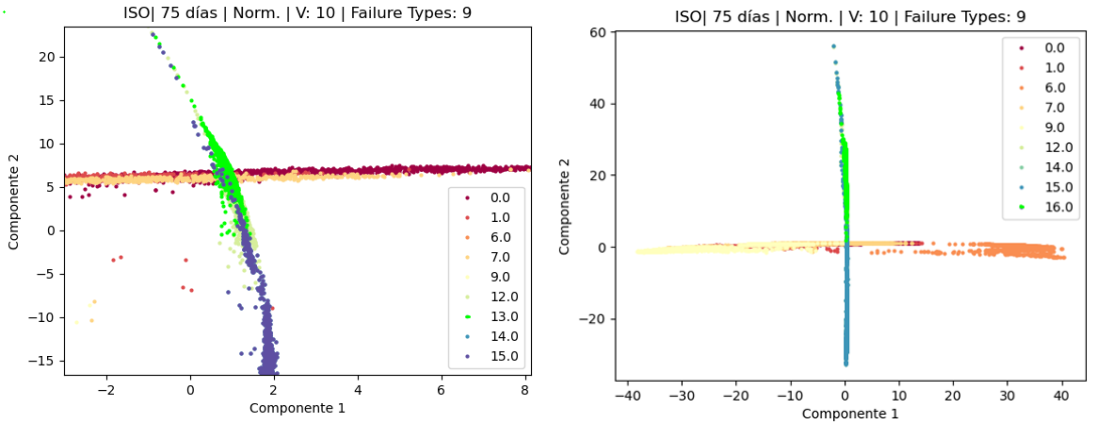


Figura 84 Resultado de la transformación con los fallos a identificar. Cada gráfica contiene un fallo distinto que se trata de clasificar en color verde fosforito.

Para concluir se ha realizado también el análisis con los algoritmos de agrupación aplicando los valores óptimos ya encontrados en la Tabla 26. Por último, se puede ver en la Figura 85 los algoritmos de agrupamiento no han sido capaces de lograr crear grupos que coincidan con los tipos de fallo.

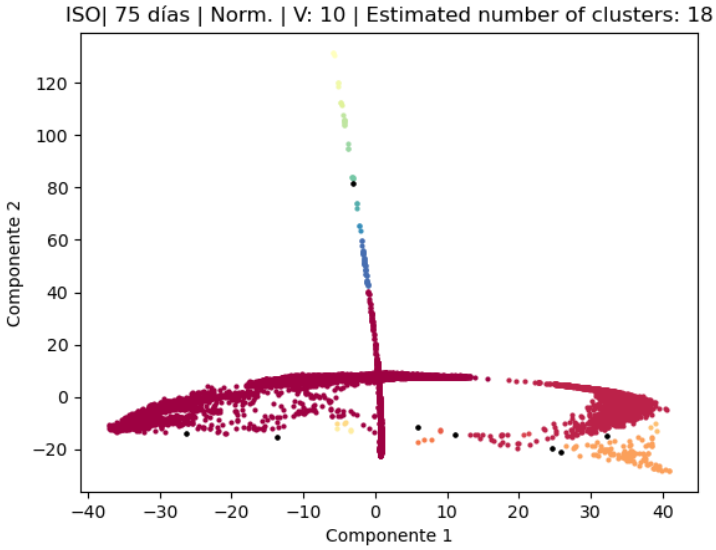


Figura 85 Agrupamientos para ISOMAP con DBSCAN y sin PCA con 75 días con cada tipo de fallo.

Capítulo 5.-Conclusiones

En este trabajo se ha analizado el comportamiento de los métodos *manifold Locally Linear Embedding* (LLE) e ISOMAP para la clasificación de fallos en la industria, concretamente en una planta depuradora de aguas residuales. Además, se han combinado con los algoritmos de agrupamiento por densidad DBSCAN, HDBSCAN y OPTICS con el mismo objetivo.

En primer lugar, se ha tratado de realizar diversas aproximaciones al problema y enfocarlo desde diferentes ángulos y el primero de ellos ha sido tratar de averiguar el número óptimo de vecinos con los que se debían aplicar los métodos *manifold*. El estudio realizado no ha arrojado datos contundentes sobre una solución clara, no obstante, con los resultados obtenidos y siguiendo la literatura previa que indica que el número óptimo suele encontrarse entre 8 y 12 vecinos, se ha optado por escoger 10 vecinos para realizar el resto de experimentos.

En cuanto a la evolución temporal se ha concluido que no siempre es mejor trabajar con el número máximo de datos (225 días). Dado que el fallo que ha ocurrido puede quedar enmascarado mientras más tiempo pasa, sobre todo si el sistema intenta compensarlo, ya que los sistemas industriales trabajan en lazo cerrado y las acciones de control del sistema pueden hacer que el fallo quede compensado. Por lo tanto, es necesario realizar un estudio temporal para conocer cuántas muestras son las óptimas del sistema. En el caso de este trabajo ha sido de 75 días.

Con respecto a la aplicación del método PCA para eliminar variables se puede concluir que es posible aplicar el método en combinación con los métodos *manifold*. No obstante, hay que tener en cuenta que la eliminación de variables lleva implícita la pérdida de datos y puede suponer variaciones en los resultados. La intensidad con la que afecten estas variaciones dependerá por supuesto del sistema, la calidad y la cantidad de los datos que se estén tratando.

En relación a los algoritmos de agrupación no han arrojado ningún resultado destacable. Debido a que ninguno de los algoritmos elegidos (DBSCAN, HDBSCAN y OPTICS) ha logrado crear agrupaciones que sean capaces de clasificar ningún fallo. Estos resultados no indican que los métodos no sean capaces de crear agrupaciones, sino que no son aplicables los datos con los que se han utilizado.

Cabe destacar que en dos casos se ha logrado una clasificación parcial de los fallos. La primera ha sido al comparar de manera individual los fallos aplicando LLE y utilizando los datos de 75 días, que es cuando más diferencia existía entre ellos. La segunda vez ha sido al tratar todos los tipos de fallo como una única matriz, esta vez aplicando ISOMAP. De esta manera aplicándole los métodos a la matriz entera se ha logrado la misma clasificación. Concretamente se ha logrado separar en dos grupos los

tipos de fallos, el primer grupo englobaría los fallos del 1 al 11, mientras que el segundo grupo estría formado por los fallos del 12 al 16. Hay que tener en cuenta que en ambos casos se clasificaban también los datos sin fallos en el primer grupo, lo que significa que el método no sería capaz de detectar un fallo, únicamente de clasificarlo. No obstante, sería posible utilizar las dos maneras con el fin de detectar a que grupo pertenece el fallo a analizar. Debe puntualizarse que la clasificación conseguida seguramente se deba a que los fallos pertenecientes a cada grupo se parecen entre sí y son notablemente distintos a los del grupo contrario.

De la clasificación lograda se infiere que cuando se trata una cantidad reducida de datos el método LLE funciona mejor que ISOMAP. Y al revés, cuando se trata de analizar de forma global los datos ISOMAP tiene ventaja. Esta conclusión coincide con la manera que tienen de trabajar ambos métodos, dado que como se ha explicado en la parte teórica LLE crea pequeños parches actuando de forma local. Mientras que ISOMAP preserva las propiedades globales de los datos. Este comportamiento ya fue estudiado por Silva y Tenenbaum en 2007 [36].

En definitiva, a pesar de no haber cumplido totalmente con el objetivo del trabajo, si se ha logrado una clasificación parcial de los fallos aplicando únicamente los métodos *manifold*. Al mismo tiempo no se ha logrado un uso efectivo de los algoritmos de agrupación basados en densidad. Además, se han extraído valiosas conclusiones sobre cómo deben aplicarse o que consideraciones se deben tener en cuenta.

Como trabajo futuro se puede considerar varias opciones:

- Trabajar con otros datos para ver si estos métodos LLE e ISOMAP son capaces de usarse para detección e identificación de fallos con otros datos de fallo que sean más diferentes entre sí.
- Trabajar con otro tipo de reducción de la dimensionalidad diferentes de los estudiados aquí.
- Trabajar con otro tipo de algoritmos de agrupamiento.
- Trabajar de forma distribuida, es decir, aplicar los métodos estudiados en vez de con datos de la planta global, con datos de unidades o partes diferentes de la planta, para ver si esto mejora los resultados.
- Aplicar otras técnicas de clasificación basada en datos distintas, como las redes neuronales *Deep learning*, los árboles de clasificación (RF), etc. técnicas basadas en Big data.

Capítulo 6.-Referencias

[1] K. Machandran, I. Jurčić, V. Corte, Ferdinand-James y D. Sharon, «Industry 4.0.: The New Industrial Revolution.,» de *Big Data Analytics for Smart and Connected Cities*, IGI Global, 2018.

[2] J. Lee y H.-A. S. Kao, «Service innovation and smart analytics for Industry 4.0 and big data environment,» vol. 2212, nº 8271, 2014.

[3] M. J. de la Fuente, «Fault Detection And Isolation: An Overview,» Valladolid.

[4] R. Iserman, Process Fault Detection Based on Modeling and Estimation Methods-A Survey, Vols. %1 de %220-4, Elsevier, 1984, pp. 387-404.

[5] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri y K. Yin, A review of process fault detection and diagnosis: Part III: Process history based methods, Vols. %1 de %227-3, Computers & Chemical Engineering, 2003, pp. 327-346.

[6] M. Shiker, «Multivariate Statistical Analysis,» vol. 6, nº 55-66, 2012.

[7] J. Alex, L. Benedetti, J. Copp, K. Gernaey, U. Jeppsson, I. Nopens, M. Pons, C. Rosen, J. Steyer y P. Vanrolleghem, «Benchmark simulation model no. 2,» 2008.

[8] E. W. Dijkstra, «A note on two problems in connexion with graphs,» vol. 1, nº 1, 1959.

[9] O. Samko, A. Marshall y P. Rosin, «Selection of the optimal parameter value for the Isomap algorithm,» *Patter Recognition Letters,* vol. 27, nº 968-979, 2006.

[10] R. Campello, P. Kröger, J. Sander y A. Zimek, «Density-based clustering,» vol. 10, nº e1343, 2020.

[11] Scikit-learn, [En línea]. Available: https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html#sphx-glr-auto-examples-cluster-plot-cluster-comparison-py. [Último acceso: 31 Agosto 2020].

[12] P. J. Rousseeuw, «Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,» vol. 20, nº 53-65, 1986.

[13] J.-A. Martínez-Comeche, «Determinación de grupos de usuarios de bibliotecas digitales mediante el análisis de ficheros log,» vol. 40(3): e181, 2017.

[14] M. Ester, H.-P. Kriegel, J. Sander y X. Xu, «A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,» vol. AAAI Press '96, nº 226-231, 1996.

[15] T. Oommen, D. Misra, N. Twarakavi y e. al., «An Objective Analysis of Support Vector Machine Based Classification for Remote Sensing,» vol. 40, nº 409-424, 2008.

[16] L. McInnes, J. Healy y S. Astels, «hdbscan: Hierarchical density based clustering,» vol. 2, nº 11, 2017.

[17] M. Ankerst, M. Breunig, H.-P. Kriegel y J. Sander, «OPTICS: Ordering Points To Identify the Clustering Structure,» vol. 28, nº 49-60, 1999.

[18] Python Software Foundation, Python Software Foundation, 2001. [En línea]. Available: https://www.python.org/. [Último acceso: 28 Agosto 2020].

[19] F. Pedregosa y e. al., «Scikit-learn: Machine Learning in Python,» vol. 12, nº 2825-2830.

[20] Scikit-learn, [En línea]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.manifold.Isomap.html. [Último acceso: 31 Agosto 2020].

[21] Z. Zhang y J. Wang, «MLLE: Modified Locally Linear Embedding Using Multiple Weights,» vol. 19, nº 1593-1600, 2006.

[22] Scikit-learn, [En línea]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.manifold.LocallyLinearEmbedding.html. [Último acceso: 31 Agosto 2020].

[23] Scikit-learn, [En línea]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html. [Último acceso: 31 Agosto 2020].

[24] Scikit-learn, [En línea]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.OPTICS.html. [Último acceso: 31 Agosto 2020].

[25] L. McInnes, J. Healy y S. Astels, «hdbscan: Hierarchical density based clustering,» vol. 2, nº 11, 2017.

[26] L. McInnes, J. Healy y S. Astels, 2016. [En línea]. Available: https://hdbscan.readthedocs.io/en/latest/index.html. [Último acceso: 31 Agosto 2020].

[27] The SciPy community, [En línea]. Available: https://numpy.org/doc/stable/user/whatisnumpy.html. [Último acceso: 1 Septiembre 2020].

[28] MathWorks, 2020. [En línea]. Available: https://es.mathworks.com/products/matlab.html. [Último acceso: 20 Septiembre 2020].

[29] NumFOCUS, [En línea]. Available: https://pandas.pydata.org/. [Último acceso: 01 Septiembre 2020].

[30] J. Hunter, D. Dale, E. Firing y M. Droettboom, 14 Agosto 2020. [En línea]. Available: https://matplotlib.org/tutorials/introductory/pyplot.html. [Último acceso: 01 Septiembre 2020].

[31] J. VanderPlas, Python Data Science Handbook, O'Reilly Media, Inc., 2016, pp. 445-462.

[32] I. Färber, S. Günnemann, H. Kriegel, P. Kröger, E. Müller, E. Schubert, T. Seidl y A. Zimek, «On Using Class-Labels in Evaluation of Clusterings,» Washington, 2010.

[33] R. A. Fisher, «The Use Of Multiple Measurements In Taxonomic Problems,» vol. 7, nº 2, 1936.

[34] Scikit-learn, [En línea]. Available: https://scikit-learn.org/stable/auto_examples/manifold/plot_lle_digits.html. [Último acceso: 2020 Agosto 28].

[35] T. Caliński y J. Harabasz, «A dendrite method for cluster analysis,» vol. 3:1, nº 1-27, 1974.

[36] V. de Silva y J. Tenenbaum, «Global versus local methods in nonlinear dimensionality reduction,» vol. NIPS'02, nº 721-728, 2002.

[37] L. K. Saul y S. T. Roweis, «An Introduction to Locally Linear Embedding,» 2020. [En línea]. Available: https://cs.nyu.edu/~roweis/lle/papers/lleintro.pdf.

[38] J. B. Tenenbaum, V. de Silva y J. Langford, «A Global Geometric Framework for Nonlinear Dimensionality Reduction,» vol. 290, nº 5500, 2319-2323, 2000.

[39] Scikit-learn, 31 Agosto 2020. [En línea]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestNeighbors.html.

[40] Scikit-learn, «Glossary,» [En línea]. Available: https://scikit-learn.org/stable/glossary.html#term-n-jobs. [Último acceso: 31 Agosto 2020].

[41] R. J. Patton, J. Chen y S. B. Nielsen, «Model-based methods for fault diagnosis.,» vol. 17(2), nº 73-83, 1995.

[42] M. Mishra, U. Sarkar, S. Taraphder, S. Datta, D. Swain, R. Saikhom, S. Panda y M. Laishram, «Principal Component Analysis,» vol. 1, nº 10, 2017.

[43] S. Mika, G. Rätsch, J. Weston, B. Scholkopf y K.-R. Müller, «Fisher Discriminant Analysis with Kernels,» vol. 9, nº 41-49, 1999.

[44] G. Mateos-Aparicio Morales, «Partial Least Squares (Pls) Methods: Origins, Evolution And Application To Social Sciences».