Trabajo Fin de Máster

Modelado del patrón de comportamiento de los equipos de climatización en el sector retail

Miguel Martínez Comesaña

Máster en Técnicas Estadísticas

Curso 2018-2019

Propuesta de Trabajo Fin de Máster

Título en galego: Modelado do patrón de comportamento dos equipos de climatización no sector retail

Título en español: Modelado del patrón de comportamiento de los equipos de climatización en el sector retail

English title: Modeling of the behavior pattern of climate control equipment in the retail sector

Modalidad: Modalidad B

Autor: Miguel Martínez Comesaña, Universidad de A Coruña

Director: Ricardo José Cao Abad, Universidad de A Coruña

Tutor: Miguel Díaz-Pache Gosende, EcoMT

Breve resumen del trabajo:

Este trabajo de fin de máster, de carácter práctico, se centra en el estudio de instalaciones de climatización; más concretamente, en el comportamiento de las bombas de agua que operan dentro de ellas. La novedad de este trabajo es que el análisis de los datos se hace desde un enfoque funcional; cada dato es una curva diaria con los valores de la variable estudiada. En este contexto, creamos tres algoritmos, cada uno con una función diferenciada, con el objetivo de obtener información significativa de las bombas de agua y, además, anticipar posibles malos comportamientos de una manera eficiente. El algoritmo *Clasificador*, analizando las temperaturas de impulsión y retorno, crea grupos de bombas en función de su patrón de comportamiento. El algoritmo *Clasificador.S* asigna a alguno de los grupos creados por *Clasificador* nuevas bombas que llegan sin clasificar. Y por último, el algoritmo *Detección* trata de anticipar incidencias futuras a través del estudio de curvas de temperaturas del pasado, con un comportamiento estándar para la bomba estudiada ,en comparación con los datos actuales.

Recomendaciones:

Otras observaciones:

Ricardo José Cao Abad, Catedrático de la Universidad de A Coruña, Miguel Díaz-Pache Gosende, Director de Soluciones Software de EcoMT, informan que el Trabajo Fin de Máster titulado

Modelado del patrón de comportamiento de los equipos de climatización en el sector retail

fue realizado bajo su dirección por Miguel Martínez Comesaña para el Máster en Técnicas Estadísticas. Estimando que el trabajo está terminado, dan su conformidad para su presentación y defensa ante un tribunal.

En A Coruña y Vigo, a 20 de Junio de 2019

El director:

Ricardo José Cao Abad

El tutor:

Miguel Díaz-Pache Gosende

El autor:

Miguel Martínez Comesaña

Agradecimientos

Agradezco a todo el equipo de EcoMT que ha participado y me ha ayudado en la elaboración de este TFM, a mi tutor Ricardo José Cao Abad por su guía en esta investigación y a las tres universidades (Universidad de Vigo, Universidad de Santiago de Compostela y Universidad de A Coruña) a cargo de este máster por hacer posible la realización de trabajos de fin de máster en colaboración con alguna empresa del sector privado.

Índice general

Resumen

Resumen en español

Este trabajo de fin de máster, de carácter práctico, se centra en el estudio de instalaciones de climatización; más concretamente, en el comportamiento de las bombas de agua que operan dentro de ellas. La novedad de este trabajo es que el análisis de los datos se hace desde un enfoque funcional; cada dato es una curva diaria con los valores de la variable estudiada. En este contexto, creamos tres algoritmos, cada uno con una función diferenciada, con el objetivo de obtener información significativa de las bombas de agua y, además, anticipar posibles malos comportamientos de una manera eficiente. El algoritmo *Clasificador*, analizando las temperaturas de impulsión y retorno, crea grupos de bombas en función de su patrón de comportamiento. El algoritmo *Clasificador.S* asigna a alguno de los grupos creados por *Clasificador* nuevas bombas que llegan sin clasificar. Y por último, el algoritmo *Detección* trata de anticipar incidencias futuras a través del estudio de curvas de temperaturas del pasado, con un comportamiento estándar para la bomba estudiada, en comparación con los datos actuales.

English abstract

This master's thesis, of practical character, is focused in the study of climate control equipment; in concrete, in the behaviour of the water pumps that work inside them. The novelty of this paper is that the data analisis is done from a funcional focus; each data is a daily curve with the values of the studied variable. In this context, we created three algorithms, each one with a differentiated function, with the aim of getting significant information of the water pumps and, moreover, anticipate possible bad behaviours in a efficient form. The algorithm *Clasificador* create pump´s groups based on their behaviour pattern analyzing the impulsion and return temperatures, . The algorithm *Clasificador.S* assign new unclassified pumps to some group created for the algorithm *Clasificador*. Last, the algorithm *Detección* try to anticipate future incidents through the study of past temperature curves, with a standar behaviour to the studied pump, in comparison with the current data.

Capítulo 1

Introducción

El trabajo de fin de máster aquí presentado se aleja bastante a la forma de los clásicos TFM teóricos. En este caso, se trata de un TFM dentro de la modalidad B; es decir, un trabajo realizado dentro de una empresa determinada. La que ha colaborado en la realización de este trabajo es Ecomanagement Technology (EcoMT), empresa dedicada al control y monitorización de equipos de clima por todo el mundo. De esta forma, este TFM se basa en la creación de tres algoritmos para poder estudiar de manera eficiente, y a través de un enfoque funcional, el patrón de funcionamiento de bombas de agua; las cuales operan en las instalaciones de climatización en una gran multitud de tiendas. Cada algoritmo tendrá una función específica. El algoritmo principal, conocido como *Clasificador*, se encarga de agrupar las bombas de agua según su patrón de funcionamiento, de forma que esa clasificación nos reporte información significativa sobre ellas y, además, sea más sofisticada que la división actual por nombres. Por otro lado, el algoritmo *Clasificador.S* lleva a cabo una clasificación supervisada de posibles nuevas bombas en base a los grupos creados por el algoritmo *Clasificador*. Por último, el algoritmo *Detección* va un paso más allá; su objetivo es anticipar posibles futuras incidencias para ahorrarles problemas a las tiendas en sus equipos de climatización.

A lo largo de este TFM explicaremos tanto el contexto como los resultados a los que se llega con estos algoritmos. En el segundo capítulo presentamos información relevante de la empresa colaboradora en este trabajo (EcoMT). En el tercero explicamos, de manera resumida, qué son y qué tiene de especial trabajar con datos funcionales. En el cuarto, en cambio, se presentan los datos reales con los que vamos a trabajar y poner a prueba los algoritmos. A continuación, en el capítulo 5, están explicados los tres algoritmos; todos los pasos que siguen para poder realizar, de manera automática, las funciones que tienen asignadas. En el capítulo 6 mostramos el análisis de datos llevado a cabo y los resultados obtenidos tras aplicar los tres algoritmos a dos muestras, bien diferenciadas, de bombas de agua. Para acabar, en el capítulo 7, mostramos las conclusiones a las que se llegaron tras observar los resultados que se obtienen; entre ellas, si verdaderamente los algoritmos tienen un rendimiento eficiente o no.

Capítulo 2

EcoMT

La empresa Ecomanagement Technology, con siglas EcoMT, fue la encargada de realizar la función de colaboradora externa en este trabajo de fin de máster. Es una empresa dedicada al control y monitorización de instalaciones; HVAC[1], iluminación, etc... de clientes *multisite*[2]. Su objetivo principal es conseguir un comportamiento eficiente en consumo de potencia, confort y O & M (Operations and Maintenance). Esta empresa trabaja con tiendas de más de 50 países, está gestionando más de 3000 instalaciones y controla más de 30 millones de maquinas. Además, monitoriza unas 600 mil variables y computa más de 3 billones de datos. Para resumir la evolución de la empresa en los últimos años mostramos, en el siguiente gráfico (figura 2.1), el crecimiento de las ventas junto con el consumo medio de cada instalación controlada.

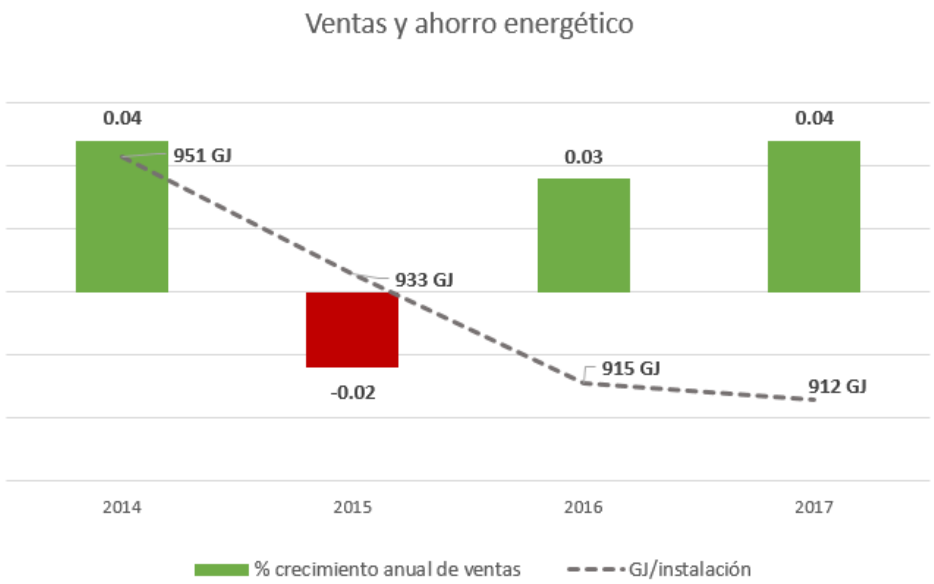


Figura 2.1: Evolución de las ventas y el consumo de energía desde 2014 hasta 2017. Las barras representan el crecimiento anual de las ventas y la línea discontinua los gigajulios (GJ) por instalación.

Exceptuando 2015, se observa una tendencia creciente en las ventas desde 2014. Además, el consumo medio por instalación[3] tiende a reducirse; sobre todo en los primeros años de control. Después, el objetivo es mantener consumos eficientes y buscar soluciones que optimicen estos consumos.

[1]Siglas con el significado siguiente: Heating Ventilating Air Conditioned.
[2]Se produce en más de un lugar o está relacionado con más de un lugar.
[3]Todas instalaciones similares para que tenga sentido el análisis.

2.1. OTEA

Para poder alcanzar sus objetivos, la empresa EcoMT cuenta con OTEA; una plataforma web que permite controlar, monitorizar y gestionar en remoto todo tipo de instalaciones. Es un sistema de telegestión centralizado para dispositivos que producen y consumen energía; dirigido a mejorar la sostenibilidad, centrándose en la productividad y en crear un entorno más confortable. En la figura 2.2 mostramos un esquema de la estructura de OTEA.

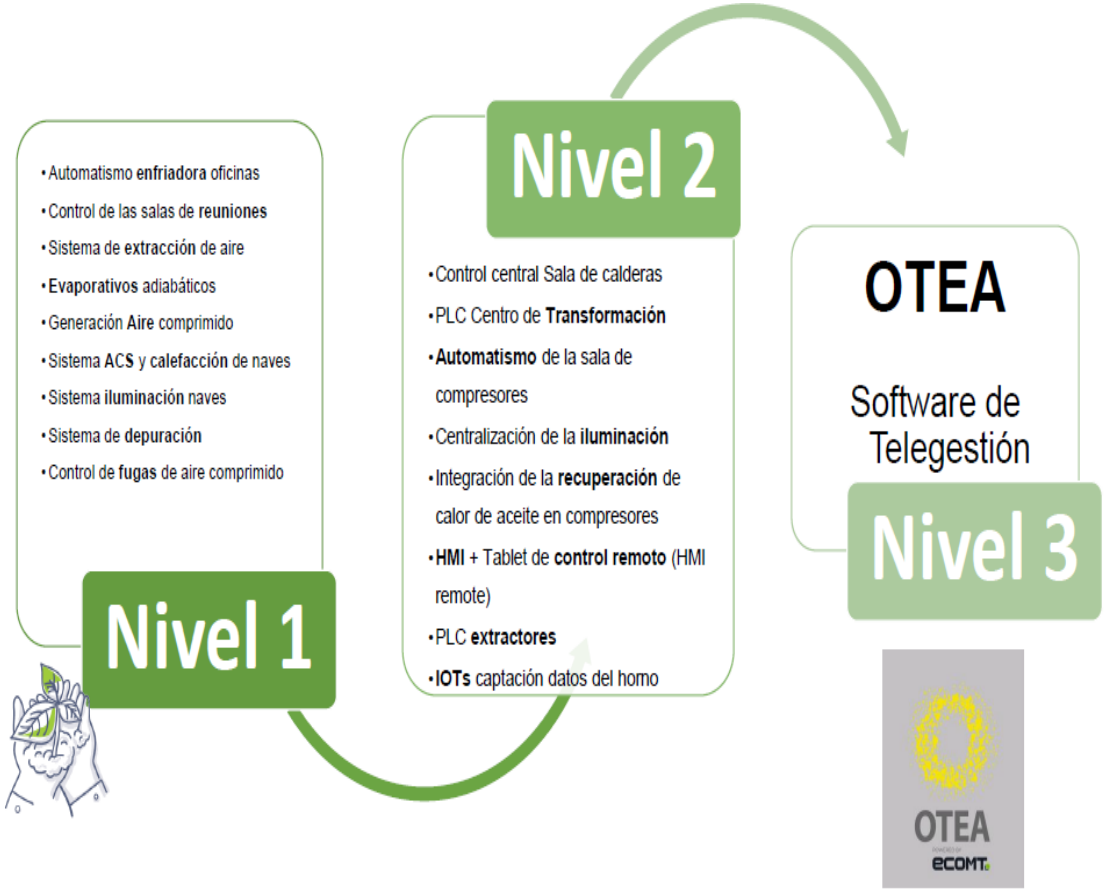


Figura 2.2: Estructura del sistema OTEA.

Características de OTEA:

- Solución muy experimentada, optimizada, segura y fiable.
- Controla equipos de climatización y iluminación en cualquier parte del mundo con un solo clic.
- Es un sistema abierto; compatible con cualquier hardware y maquina de mercado.
- Gestiona billones de datos para poder predecir futuras averías o disfunciones.
- Servicio *control - center* 24 horas del día, 365 días al año.
- Permite a las empresas tener un consumo energético eficiente y respetuoso con el medio ambiente.
- Facilita la operación y mantenimiento de instalaciones.
- Reducción de costes.
- Inteligencia artificial para gestionar las instalaciones.

2.2. Servicios avanzados de gestión externa para clientes *Multisite*

Las actuales funciones de la empresa Ecomanagement Technology se pueden dividir en: supervisión de instalaciones, informes de seguimiento, reportes de ingeniería, implantación de la normativa ISO 50001 y análisis avanzado de datos.

2.2.1. Supervisión de instalaciones: OTEA CENTER

La supervisión de un cliente *multisite* desde el centro de control de EcoMT permite mejorar las labores de O & M (Operations and Maintenance), obteniendo:

- Mejora en el servicio de mantenimiento de la instalación.
- Reducción de costes y tiempos de resolución de incidencias.
- Mejora de la productividad, aumentando el confort de clientes y personal.
- Eficiencia energética y un correcto, y continuo, funcionamiento de las instalaciones a través de la vigilancia de variables.

EcoMT desarrolla e implanta soluciones de servicios basadas en tecnologías de comunicación, automatización y gestión de datos para el diseño y/o O & M de instalaciones en clientes *multisite*; lo que permite a la dirección dedicar esfuerzos a otros procesos productivos. Las dos tareas principales para poder obtener todas estas mejoras son:

- Vigilancia de variables

  1. Uso de un algoritmo propio de filtrado y priorización en la vigilancia de variables.
  2. Autogestión de falsos positivos.
  3. Retroalimentación desde base de datos.
  4. Generador de reglas y propuestas de acciones (OTEA Expert).

- Gestión de Incidencias

  La gestión de incidencias se realiza a través de un módulo *Maker*; con un protocolo específico para cada cadena o cliente.

  1. Recepción.
  2. Registro.
  3. Filtrado de incidencias.
  4. Actuación según protocolo.
  5. Cierre e informe.

2.2.2. Informes de seguimiento

De manera periódica se generan informes de seguimiento energético. Se estudia la evolución del consumo de gas, de la energía activa, de la reactiva, del COP[4] de los equipos de producción...

2.2.3. Reportes de Ingeniería

El equipo de Ingeniería Energética de EcoMT puede suministrar información, como informes de comparación o estudios de eficiencia de líneas de equipos, útil para el control de las maquinas. Esto permite ver la evolución del confort y eficiencia de las instalaciones. Para ello controlan los parámetros de funcionamiento, proponen sugerencias para cambios en sistemas de automatización y/o diseñan mejoras.

[4]Coeficiente de rendimiento.

2.2.4. Soporte en la implantación de la normativa ISO 50001

Esta normativa internacional fue desarrollada por ISO (Organización Internacional de Normalización) y tiene como objetivo mantener y mejorar un sistema de gestión de energía en una organización; cuyo propósito es el de permitir una mejora continua de la eficiencia energética, la seguridad energética y la utilización de la energía y el consumo energético con un enfoque sistemático.

OTEA sirve de sistema de gestión para la planificación energética como punto referenciado de la norma. Existe la posibilidad de dar soporte y gestionar el seguimiento del desempeño energético de las mejoras anualmente planteadas y de todo lo relacionado con la parte energética de la norma. Algunas de sus ventajas las reflejamos en la figura 2.3.

Figura 2.3: Ventajas de la utilización de la plataforma OTEA.

2.2.5. Análisis avanzado de datos

- Modelado de parámetros.
- Detección inteligente de anomalías.
- Aplicación de algoritmos de Machine Learning.
- Integración con herramientas Big Data.

2.3. Últimos proyectos

A continuación mostramos los últimos proyectos donde EcoMT ha colaborado:

2.3.1. Fortissimo

Este proyecto está centrado, principalmente, en la mejora del sistema experto basado en el software OTEA. En el desarrollo de este proyecto ha participado personal de diferentes empresas e instituciones: EcoMT, ITMATI, UVigo, CESGA, Gompute, CINECA, Ato, y XLAB. El experimento OTEAres (Mejora

del sistema experto remoto basado en el software OTEA) se posiciona dentro de la tecnología de la información y la comunicación aplicada a la Industria 4.0.[5] OTEAres se centra en la predicción y correción temprana de incidencias, que está directamente relacionada con el uso óptimo de la energía y la reducción de los costes de mantenimiento. El monitoreo de equipos de energía como calefacción, ventilación y aire acondicionado (HVAC) implica el manejo de grandes cantidades de datos. Para cumplir con los objetivos es necesario aplicar Machine Learning (ML) junto con técnicas matemáticas y estadísticas, donde se requiere el uso de HPC[6].

El desarrollo y uso de OTEAres (esquematizado en la figura 2.4) beneficia a todos los miembros del consorcio. Por un lado, EcoMT ofrecerá un mejor servicio en el control de las instalaciones y un mayor beneficio para sus clientes con su versión mejorada de OTEA. Por otro lado, ITMATI[7] y la Uvigo aumentan su conocimiento de ML y los modelos predictivos en el campo de eficiencia energética. De este modo, el ITMATI podrá ofrecer estadísticas innovadoras y servicios Big Data dirigidos a la industria; abriendo posibilidades de participar en nuevos proyectos.

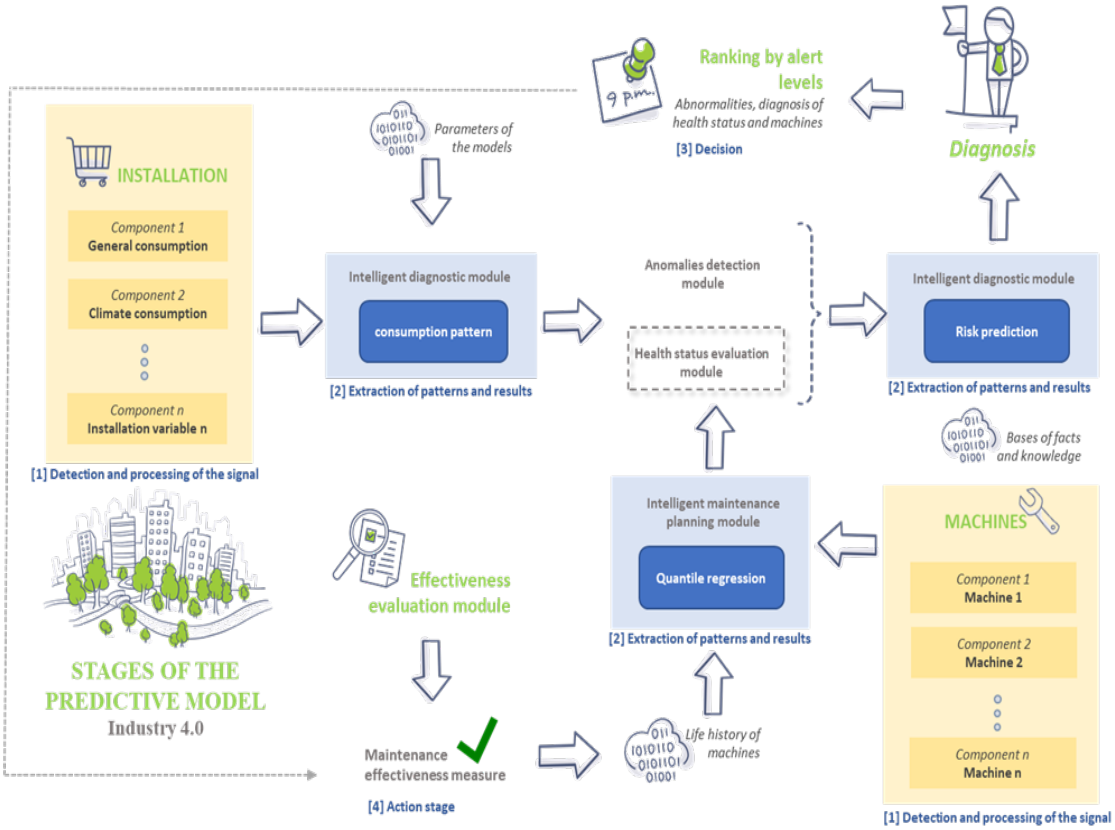


Figura 2.4: Diseño de OTEAres. Modelo predictivo para Industria 4.0.

2.3.2. MOIO

Este proyecto está orientado a la medida de volumen y temperatura en depósitos de cerveza de Bodega de Estrella Galicia. Se pretende conocer el volumen de cerveza que hay en cada depósito. El objetivo final es mostrar al usuario un porcentaje estimado del líquido que queda en el tanque, y así, optimizar la logística de reposición. Para poder alcanzar este objetivo se aplican técnicas de Machine Learning a los datos obtenidos desde OTEA.

[5]Industria consistente en la llamada *Cuarta Revolución Industrial*, enfatizando y acentuando la idea de una creciente y adecuada digitalización y coordinación cooperativa en todas las unidades productivas de la economía.
[6]Computación de alto rendimiento.
[7]Insituto Tecnológico de Matemática Industrial.

Figura 2.5: Pasos y estado actual del proyecto MOIO.

Estrategia

Midiendo cada segundo el volumen y la temperatura del aire que entra en el tanque, se pretende calcular el líquido extraído. Para que la aplicación muestre el porcentaje de forma correcta se debe conocer el volumen de líquido del que parte el tanque cada vez que se rellena. Los datos que se procesan son simulados ya que no se cuenta con datos pertenecientes a un entorno real. El modelo obtenido con datos reales tendrá que ser reajustado.

Para entrenar el modelo se aplican algoritmos matemáticos y físiscos a los datos simulados. Se consiguen resultados que consiguen simular el proceso aprendiendo de la repetición. Por otro lado, la validación del modelo se realiza con combinaciones de datos nuevos; se obtienen resultados con errores por debajo del 5 %.

2.3.3. BlockChain

Este proyecto es realizado en colaboración con la empresa ABanca. Determinadas sucursales de ABanca medirán consumos eléctricos y los almacenarán en BlockChain. EcoMT se encarga de conectar la sucursal a la plataforma OTEA para monitorizar consumos y telegestionar luces, alimentación y la energía global del establecimiento.

Por otro lado, se ha conectado la plataforma LoT OTEA a la plataforma Ithium de GodEnigma, que permite almacenar los consumos en una base de datos distribuida en BlockChain. De esta forma se puede garantizar que las lecturas no se pueden manipular ni cambiar y, por tanto, dar certificaciones de ahorros en consumo año a año. Con esto se podrá *tokenizar*[8] dichos consumos, que vendrían siendo como el valor de la *no emisión* de CO2 o el valor del ahorro energético de un periodo con respecto al pasado.

2.4. Colaboración

La colaboración *alumno - empresa* se tradujo en un contrato de prácticas, en el cual el alumno estuvo un total de 250 horas acudiendo a la sucursal de A Coruña, pudiendo obtener ayuda y orientación en el desarrollo de los algoritmos. Por otro lado, mediante el acceso a la plataforma *OTEA* (representada en la figura 2.6), que es una pagina web que da acceso a todas las tiendas y maquinas controladas por la empresa, se pudo buscar y obtener los datos que han sido utilizados en este TFM. La forma de obtener los datos de determinadas variables de una maquina era la descarga en csv de las variables en columnas. Esta descarga masiva de datos podía ser de varias variables de una instalación o de un grupo de ellas.

[8]Posibilidad de vincular, de forma rápida y sencilla, una tarjeta de pago con los servicios de pago digitales.

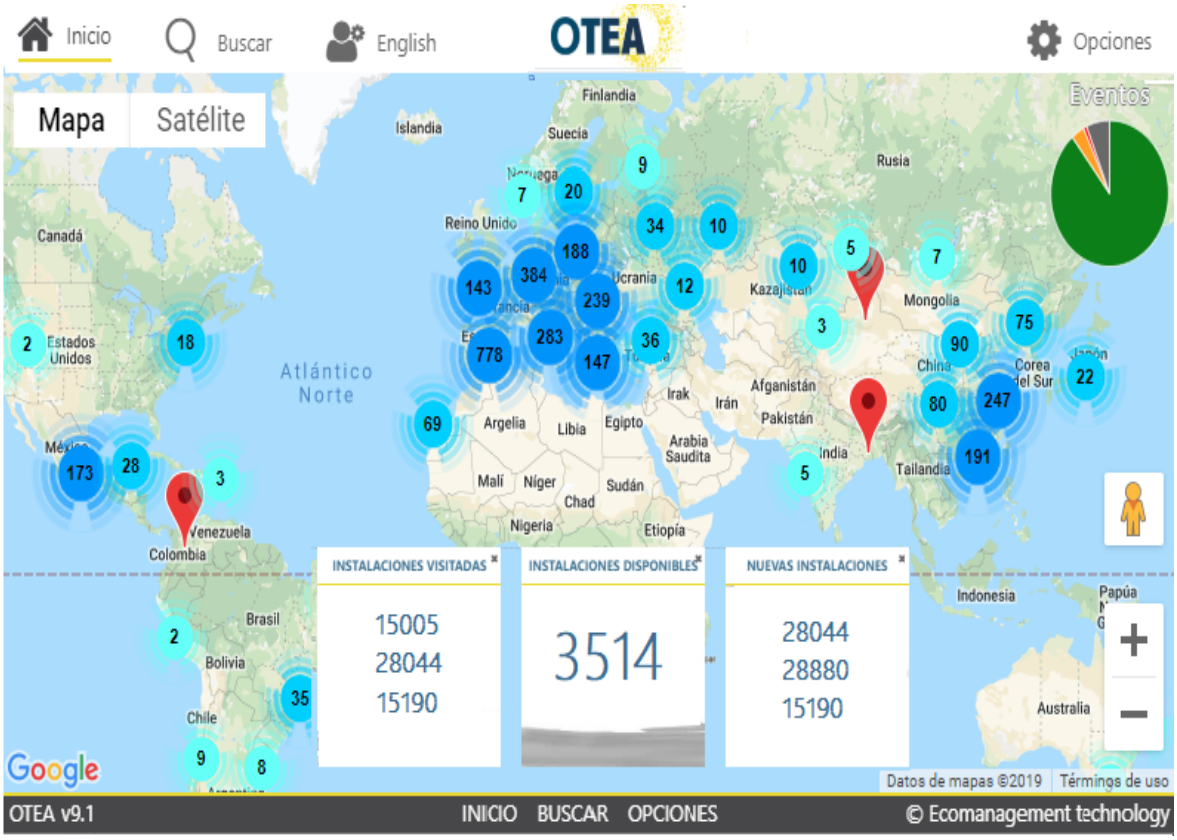


Figura 2.6: Apariencia de la pagina principal del sistema de telegestión OTEA.

2.5. Próximos pasos

En un primer momento EcoMT busca mejorar el Sistema Experto de Control remoto basado en OTEA; salir de los sistemas generales y pasar a lineas de proceso. Además, también se espera avanzar en los siguientes campos:

- Análisis de aprovechamiento energético de hornos.
- Nuevo proceso de depuración de aguas residuales.
- Producción de agua caliente primaria para equipos de lavado.
- Estudio de cambio de fuente de energía para la producción de calor desde electricidad a gas natural.
- Monitorización del consumo de gases (H_2, N_2...).
- Recuperación de energía calorífica vertida en los humos de escape.
- Estudio de la Huella de Carbono[9].

[9]Totalidad de gases efecto invernadero (GEI) emitidos directa o indirectamente por un individuo, organización, evento o producto.

Capítulo 3

Datos Funcionales

Antes de presentar el algoritmo o los datos con los que hemos trabajado, es necesario presentar qué son los datos funcionales y cuales son sus características más importantes. La forma más simple de explicarlos es la siguiente: en vez de trabajar con datos de la manera tradicional, variables que en determinados puntos de evaluación obtienen unos valores, trabajamos con datos que son funciones. Es decir, el análisis de datos funcionales (FDA, Functional Data Analysis) se centra en la descripción estadística y en la modelización de muestras de funciones aleatorias. Su objetivo es realizar inferencia con la curva entera, y no con números reales o vectores, como de costumbre.

Para mostrar el cambio que conlleva trabajar con este tipo de datos vamos a fijarnos en el espacio muestral $\mathcal{X}$[10] y el espacio paramétrico Θ[11]. Esto se debe a que el progreso en la estadística matemática, tal y como presenta Cuevas (2014), puede ser resumido en el estudio de estructuras más sofisticadas para $\mathcal{X}$ y Θ. En nuestro caso, el análisis de datos funcionales, es uno de los últimos avances en la estadística en el cual $\mathcal{X}$ (también Θ en otros muchos casos) es un espacio funcional de infinita dimensión. A continuación, en el Cuadro 3.1 presentamos las teorías estadísticas más conocidas de los últimos años, mostrando sus diferencias en términos del espacio muestral $\mathcal{X}$ y el espacio paramétrico Θ:

Teoría	Año	$\mathcal{X}$	Θ
Inferencia paramétrica clásica	1920	$\mathcal{R}$	$\Theta \subset \mathcal{R}$
Análisis multivariante	1940	$\mathcal{R}^d$ $(n >> d)$	$\Theta \subset \mathcal{R}^k$ $(n >> k)$
Inferencia no paramétrica	1960	$\mathcal{R}^d$ $(n >> d)$	Un espacio de funciones
Alta dimensión	2000	$\mathcal{R}^d$ $(n < d)$	$\Theta \subset \mathcal{R}^k$
Análisis datos funcionales	1990	Un espacio funcional	$\mathcal{R}^k$ o un espacio de funciones

Cuadro 3.1: Últimas teorías estadísticas y su interpretación del espacio muestral y el espacio paramétrico. Fuente Cuevas (2014).

Como ya mencionamos antes, el análisis de datos funcionales apareció para poder resolver problemas donde los datos disponibles son una muestra de n funciones $(x_1 = x_1(t), ..., x_n = x_n(t))$ definidas en un intervalo compacto de la recta real. Realmente, lo que hace de este análisis una nueva rama en la teoría estadística es la naturaleza infinita-dimensional de las muestras con las que se trabaja (puede consultarse en Cuevas (2014)). Algunos ejemplos de este tipo de datos serían los que presentamos a continuación (figura 3.1):

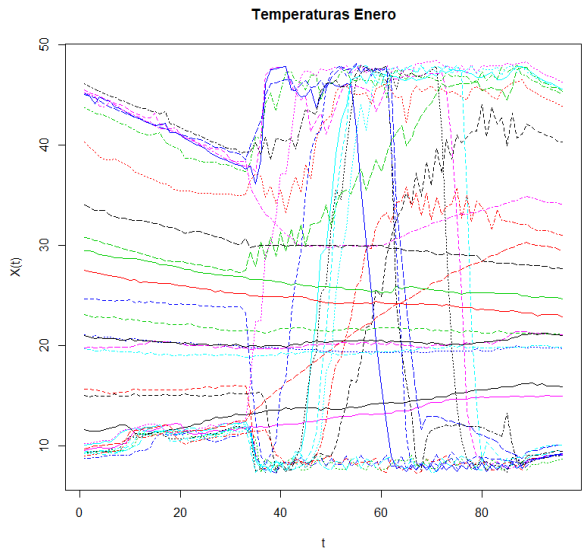
(a) Mediciones diarias de temperaturas siendo cada día una curva.

[10] Espacio donde se distribuyen los datos disponibles.
[11] Espacio donde debe caer el parámetro objetivo.

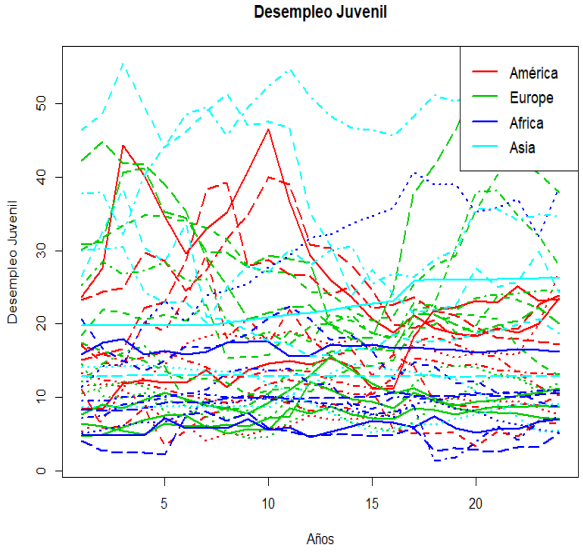
(b) Medidores económicos medidos a lo largo de un período de tiempo donde las curvas serían diferentes países.

(c) Niveles de nitrógeno de una ciudad a lo largo del día; cada día sería una curva.

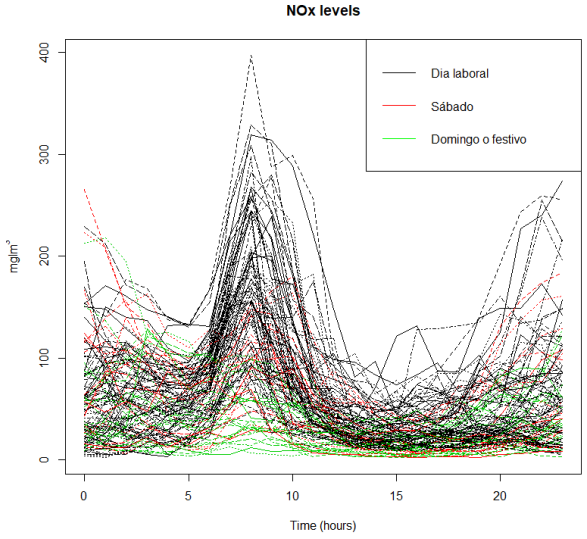
(d) Curvas de aprendizaje de diferentes fonemas. Cada curva sería un individuo.



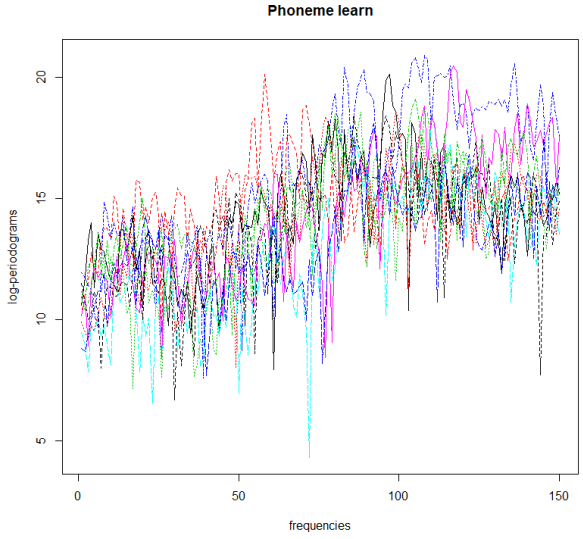
(a) BC01CiudadReal. Curvas de temperaturas de impulsión.



(b) Desempleo juvenil dividido por continentes.



(c) Niveles de Nitrógeno por día.



(d) Curvas de aprendizaje de la pronunciación de un determinado fonema de distintos individuos.

Figura 3.1: Ejemplos de diferentes conjuntos de datos funcionales.

Este trabajo ha sido elaborado en base a los datos funcionales y no a otro enfoque, como por ejemplo el multivariante, por diversas razones que pueden consultarse en Febrero et al (2008):

- La estructura de correlación temporal no se tiene en cuenta cuando utilizamos un análisis multivariante.
- El carácter de dimensión infinita de la variación funcional provoca, en algunos casos, que la rejilla de puntos donde se valoran las funciones sea mayor que el número de sujetos en el estudio. Algo que choca con el problema de la maldición de la dimensión (*curse of dimensionality*) que ocurre en la mayor parte de los métodos estadísticos multivariantes.

- Varias suposiciones sobre la distribución son impuestas en los conjuntos de datos funcionales. Sin embargo, los métodos multivariantes se restringen implícitamente a poblaciones Gaussianas o elípticas.
- En el contexto funcional, la estructura de covarianza y correlación son difíciles de interpretar, ya que no suelen presentar la verdadera variabilidad de los datos.

3.1. Teoremas limite funcionales

Existen versiones de la Ley Fuerte de los Grandes Números (SLL) y del Teorema Central del Limite (CLT) como herramientas en el análisis de datos funcionales. Los enunciados más básicos son análogas a los de una dimensión (puede consultarse en Cuevas (2014)):

- *Ley Fuerte Funcional de los Grandes Números*: Partimos de X_n como una secuencia de elementos aleatorios independientes e idénticamente distribuidos en un espacio de Banach $\mathcal{X}$ [12]. Si $\mathrm{E}||X_1|| < \infty$ tenemos que $\frac{\sum_{i=1}^{n} X_i}{n} \rightarrow E(X_1)$ *casi seguro*, siendo $E(X_1)$ la esperanza de X_1.
- *Teorema Central del Limite Funcional*: Partimos de X_n como una secuencia de elementos aleatorios independientes e idénticamente distribuidos en un espacio separable de Hilbert $\mathcal{X}$ [13]. Si $E||X_1||^2 < \infty$ luego $\sqrt{n}\left(\frac{\sum_{i=1}^{n} X_i}{n} - E(X_1)\right) \rightarrow \mathcal{G}(0, \Gamma x_1)$ *en distribución*, siendo $\mathcal{G}(0, \Gamma x_1)$ la distribución de probabilidad Gaussiana en $\mathcal{X}$ con esperanza 0 y covarianza $\Gamma x_1(x^*, y^*) = Cov(x^*(X_1)), y^*(X_1))$ para $x^*, y^* \in \mathcal{X}^*$.

3.2. Media y mediana funcional

La media en el contexto funcional es similar al de números reales. Si partimos de un $\mathcal{X}$ que toma valores en un espacio de Hilbert y $E||X||^2 < \infty$, la media de $\mathcal{X}$; $m = E(\mathcal{X})$, cumple $E||X - m||^2 = min_{a \in \mathcal{X}} E||X - a||^2$. Por otro lado, la mediana *funcional* $M = M(X)$ minimiza $E(||X - a|| - ||X||)$; expresión que existe incluso cuando $E||X|| = \infty$. En el caso de $E||X|| < \infty$ se cumple que $E||X - M||^2 = min_{a \in \mathcal{X}} E||X - a||^2$. Además, para garantizar que solo tendremos una única *curva* mediana, X tiene que ser un espacio estrictamente convexo ($(||x + y|| < ||x|| + ||y||)$). Y en el caso de simetría, asumiendo que las distribuciones de $X - m$ y $m - X$ son iguales, tenemos que la media es la misma que la mediana (explicado en Cuevas (2014)).

En el caso empírico nos tenemos que basar en la muestra $X_1, ...X_n$ de la que dispongamos y substituir las esperanzas por los valores muestrales. De esta forma las fórmulas nos quedarían de la siguiente manera:

$$\hat{m} = \frac{1}{n} \sum_{i=1}^{n} X_i \qquad ; \qquad \hat{M} = argmin_a \sum_{i=1}^{n} ||X_i - a||$$

Nosotros en el análisis de los datos de curvas de temperaturas utilizamos la curva mediana como forma de caracterizar cada bomba individualmente. Hemos tomado esta decisión debido a que existe una pequeña posibilidad de que en el análisis final de los datos pueda llegar alguna observación, que aunque no sea considerada outlier, no está dentro del patrón de funcionamiento que intentamos extraer de los datos. La mayor robustez de la mediana la hace, en este caso, más eficiente.

3.3. Profundidad (basada en mediana)

Este concepto, en el contexto de datos funcionales, tiene una gran importancia. Una función de profundidad, relativa a una medida de probabilidad P en el espacio muestral $\mathcal{X}$, tal como explica Cuevas (2014) es una función no negativa definida en $\mathcal{X} \rightarrow D(P, x)$ que nos proporciona *cómo de profundo* es el dato x respecto a la distribución P. Actualmente este tipo de funciones están adquiriendo popularidad porque nos reporta una forma de ordenar los datos, en base a las determinadas profundidades (por ejemplo, la mediana sería el dato más profundo). Pongámonos en el contexto más simple donde $\mathcal{X} = R$; las funciones de profundidad tendrán

[12] Espacios de funciones de dimensión infinita.

[13] Espacio de producto interior. Un producto interior en un espacio vectorial H da lugar a una norma definida como sigue:

$$||x|| = \sqrt{\langle x, x \rangle}$$

H será un espacio de Hilbert si es completo con respecto a esa norma.

la siguiente forma[14]:

$$D_0(P,x) = P(-\infty,x]P[x,\infty) \qquad ; \qquad D_1(P,x) = \text{mín}(P[x,\infty), P(-\infty,x])$$

Si cogemos la definición D_1 de manera más general obtenemos la profundidad de *Tukey*, $D_T(P,x)$, definida como la profundidad que es mayor o igual a todas la demás profundidades; medidas respecto a proyecciones de x de una dimensión. También podemos obtener una versión aleatorizada; en la cual trabajamos con k direcciones de las proyecciones independientes e idénticamente distribuidas $v_1, ...v_k$, y tiene la forma siguiente:

$$D_{RT}(P,x) = \text{ínf}_{1\leq i\leq k}\, D_1(P_{v_i}, \langle v_i, x\rangle)$$

siendo P_{v_i} la distribución de $\langle v_i, X\rangle$; proyecciones undimensionales de X en la dirección v_i.

Existen muchas otras formas de obtener medidas de profundidad, pero nosotros hemos optado por utilizar la profundidad *Fraiman-Muniz*, mejor explicada en Febrero et al (2008). Su fórmula es la siguiente:

$$FMD_n(x_i) = \int_a^b 1 - \|\tfrac{1}{2} - F_{n,t}(x_i(t))\|dt$$

dónde $F_{n,t}$ es la función de distribución acumulada empírica de los puntos de las curvas en un momento $t \in [a,b]$ y tiene la forma:

$$F_{n,t}(x_i(t)) = \tfrac{1}{n}\sum_{k=1}^{n} I(x_k(t) \leq x_i(t))$$

siendo I() la función indicadora.

Principalmente utilizamos este tipo de profundidad en la detección de outliers. En el cuadro 3.2 mostramos el resultado de un pequeño estudio de simulación[15], donde se compara el rendimiento de tres tipos de profundidades en la detección de outliers. Como se puede observar, las profundidades Fraiman-Muniz y Proyecciones Aleatorias siempre muestran un tiempo de computación menor. Entre ellas, se diferencian en que la primera tiene el menor tiempo de computación en media. Respecto al número outliers detectados, el resultado del estudio de simulación difiere entre trabajar con una muestra recortada o con pesos basados en las profundidades. Como no se observa ninguna regla que asegure una detección de más outliers de manera generalizada, nos quedamos con el método más *rápido*: Fraiman-Muniz.

	Muestra Recortada		Muestra Ponderada	
Tipo de profundidad	T. Comp.	n	T. Comp	n
Fraiman - Muniz	6.37	3.51	7.35	1.99
Modal	16.05	5.47	17.78	0.38
Random Projections	6.86	5.17	7.95	1.03

Cuadro 3.2: Comparación entre profundidades a través del tiempo de computación y del número de outliers que detectan. Simulación en base a remuestras bootstrap (n= 75) de las 365 curvas de una bomba de agua aleatoria y 100 repeticiones de Monte Carlo.

[14]Este tipo de funciones son mucho más útiles en un contexto multivariante $\mathcal{X} = R^d$.
[15]Con datos de temperaturas de impulsión de bombas de agua.

3.4. Outliers

La detección de outliers es algo esencial, y más en este trabajo, porque eliminamos de nuestro conjunto de datos observaciones que distorsionarían su posterior análisis debido a su estructura alejada del comportamiento normal del resto de los datos. Normalmente estos datos *extraños* son causados por diferente tipo de contingencias que hacen que estos datos no sean aptos para el estudio de patrones de funcionamiento. Esta búsqueda puede realizarse desde diferentes puntos de vista. Un enfoque univariante no sería eficiente porque puede existir una curva que sea outlier pero sus valores no lo sean de manera univariante. Una detección multivariante tampoco sería válida, por las razones que mencionamos arriba relacionadas con la maldición de la dimensión o su restrictividad a muestras normales o elípticas. Por lo tanto, lo mejor vuelve a ser una búsqueda de datos outliers desde la perspectiva funcional; en concreto, basándonos en las profundidades de las curvas, tal y como se plasma en Febrero et al (2008). Como ya adelantábamos anteriormente, la profundidad elegida para el análisis de outliers es la profundidad de Fraiman-Muniz. De esta forma un dato outlier será aquel dato con un valor de profundidad muy pequeño; es decir, estará lejos del centro de la muestra (curva central o curva mediana).

3.4.1. Outliers en un contexto funcional

Si estamos trabajando con datos funcionales, como es el caso, un dato (una curva) podrá ser outlier por las siguientes razones:

1. Una curva que contenga errores de medición.
2. Una curva real (sin errores de medición) pero de la que tenemos sospechas que no ha seguido el mismo patrón que las demás curvas.

Aunque en un entorno de datos funcionales dar una definición de dato outlier no es fácil, nosotros seguimos la definición que da Febrero et al (2008). Un dato outlier será aquel que fue generado por un proceso estocástico con diferente distribución que el resto de las curvas; que son consideradas distribuidas idénticamente. Una curva, por lo tanto, puede ser un outlier si está lejos de la función esperada del proceso estocástico o tiene una forma diferente que las demás curvas. De este modo, estamos considerando como datos *extraños* aquellas curvas que difieren de la mayoría únicamente en algunos subintervalos del periodo completo considerado. Como ya mencionamos antes, para poder llegar a medir la distancia de estos posibles outliers, utilizamos las profundidades. Los pasos a seguir para que finalmente un dato sea etiquetado como outlier son los siguientes:

1. Obtener las profundidades funcionales $D_n(x_1), ..., D_n(x_n)$ siguiendo alguno de los muchos métodos para obtener medidas de profundidad que existen. En nuestro caso, utilizamos la de Fraiman - Muniz.
2. Seleccionar k curvas $x_{i1}, ..., x_{ik}$ tales que $D_n(x_{ik}) \leq C$ para un C dado. Estas curvas serán consideradas outliers y serán eliminadas de la muestra.
3. Volvemos al paso 1 con la nueva muestra, tras haber eliminado los outliers, y repetimos hasta no encontrar ningún nuevo outlier.

Este proceso es importante para evitar posibles efectos de *enmascaramiento*. Este fenómeno se produce cuando algunos outliers reales *enmascaran* la presencia de otros. Si esto ocurre en la primera iteración, Febrero et al (2008) prueba que en alguna iteración posterior conseguiremos detectar los outliers *enmascarados*. Por otro lado, para definir la constante C tendremos en cuenta que los datos outliers deberán tener las menores profundidades. Además será conveniente que elijamos un C que nos proporcione un nivel de errores de tipo I [16] controlado. De esta forma se elegirá un C tal que, en ausencia de outliers, la probabilidad de etiquetar mal a un dato correcto como outlier sea aproximadamente un 1 %:

$$Pr(D_n(x_i) \leq C) = 0.01 \; i = 1, ...n$$

El C elegido sería el percentil 1 % de la distribución de la profundidad funcional elegida. Debido a que esta distribución es desconocida tendremos que estimar este percentil en base a la muestra que tenemos. Para esto existen varias técnicas bootstrap: bootstrap basado en una muestra recortada y bootstrap basado

[16] Rechazar la hipótesis nula cuando es verdadera; es decir, definir como dato normal un dato verdaderamente outlier.

en probabilidades en función de las profundidades (ponderado). En este trabajo hemos seguido el bootstrap basado en una muestra recortada, ya que en cuadro 3.2 se observa que, en media, es el método que detecta más outliers. El proceso de suavización bootstrap basado en recortar la muestra sigue los siguientes pasos:

1. Obtener las profundidades funcionales $D_n(x_1), ..., D_n(x_n)$.
2. Obtener B remuestras bootstrap estándar de tamaño n del conjunto de datos antes de eliminar el $\delta\,\%$ de las curvas menos profundas.

$$x_i^b \qquad \text{para } i = 1, ...n,\ b = 1, ...B$$

3. Obtener las remuestras bootstrap suavizadas $y_i^b = x_i^b + z_i^b$, siendo $(z_i^b(t_1), ...z_i^b(t_m))$ normal con media 0 y matriz de covarianzas $\gamma\Sigma_x$, donde Σ_x es la matriz de covarianzas de $x(t_1), ..., x(t_m)$ y γ el parámetro de suavización bootstrap (en nuestro caso el 5 %).
4. Para cada remuestra bootstrap $b = 1, ..., B$ calcular el percentil 1 % empírico C^b de las profundidades $D(y_i^b)$, $i = 1, ..., n$.
5. Definir C como la media de los B valores C^b.

En este tipo de bootstrap, la forma de recortar la muestra es esencial. Por este motivo, el $\delta\,\%$ tomado para recortar la muestra original, es lógico que sea algo cercano a la proporción de outliers que sea espera. La eficiencia de cada uno de los métodos bootstrapo mencionados también está plasmado en Febrero et al (2008). El método de una muestra recortada detecta mejor outliers, pero también detecta más falsos outliers. Como en nuestro caso penalizamos más detectar de menos que de más, será más apropiada la técnica bootstrap con la muestra recortada. En relación al efecto de *enmascaramiento*, nuestros datos no nos permiten iterar en la búsqueda de outliers porque, por término general, la naturaleza de los datos hace que no pare hasta que queden pocos o ningún dato para analizar. La solución a este problema la explicaremos en la sección 5.2.1.

3.5. Clasificación

En este trabajo se abordan dos tipos de clasificación: supervisada y no supervisada. En la clasificación supervisada tratamos de asignar un dato a una categoría; ya existe una muestra de *entrenamiento* de elementos bien clasificados. En cambio, en la clasificación no supervisada el objetivo es diferenciar los elementos en grupos o clases homogéneas según la información que tenemos. Paralelamente, hay que tener en cuenta que en un contexto de datos funcionales, las clasificaciones no son las mismas que en un contexto multivariante. Para ilustrarlo utilizaremos la motivación dada por Cuevas (2014).

Partimos de un elemento aleatorio X, el cual solo toma valores en el espacio $\mathcal{X}$ y que puede ser observado en dos poblaciones G_1 y G_2. Además definimos μ_i como las distribuciones de $X|Y = j$, $j = 0, 1$, donde Y es una variable dicotómica que nos dice la población a la que pertenece el dato. Los datos de los que disponemos es una muestra de observaciones independientes $\{(X_i, Y_i), 1 \leq i \leq n\}$. El problema a resolver sería clasificar una nueva observación en una población o en otra, en función de la información que nos reporta una muestra de *entrenamiento*. Aquí es donde está la principal diferencia entre clasificaciones en un contexto funcional y un contexto multivariante. El espacio muestral $\mathcal{X}$, en el caso multivariante, es un espacio Euclideo $\mathcal{X} = R^d$; sin embargo, trabajando con datos funcionales $\mathcal{X}$ es un espacio funcional (dimensión infinita). De este modo, el objetivo es encontrar una regla de clasificación que minimice [17] el error de clasificación $P(g(X) \neq Y)$. La regla de clasificación óptima es la regla conocida como *Regla de Bayes*:

$$g^*(x) = 1_{\{\gamma(x) > 1/2\}} \qquad \text{siendo } \gamma(x) = E(Y|X = x)$$

De hecho, el error de clasificación mínimo $L^* = P(g^*(X) \neq Y)$ es conocido como error de Bayes. Aunque por término general se desconoce la forma de $\gamma(x)$, se puede aproximar a través de la muestra de entrenamiento.

[17] al menos asintóticamente.

3.5.1. Clasificación supervisada

Acabamos de ver en la anterior expresión que los problemas de clasificación tienen una estrecha relación con los problemas de regresión. Cualquiera estimador de $\hat{\gamma}(x)$ nos proporcionaría un clasificador a través de un método *plug-in*. O por otro lado, obtendríamos clasificadores del estilo kernel o k vecinos más próximos si introducimos en la formula un estimador no paramétrico de la manera siguiente (puede consultarse en Cuevas (2014)):

- Kernel (Nadaraya -Watson): $H_{ni}(x) = \frac{K\left(\frac{D(x,X_i)}{h}\right)}{\sum_{j=1}^{n} K\left(\frac{D(x,X_j)}{h}\right)}$ siendo D distancias, h un parámetro de suavizado y K una función núcleo.
- K-vecinos proximos: $H_{ni}(x) = \frac{1_{B(x,C_k(x))(X_i)}}{k}$ donde $C_k(x)$ es la distancia de x a la k observación más cercana entre $X_1, ...X_n$. En otras palabras, para estimar $\gamma(x)$ estamos haciendo un promedio de las respuestas Y_i que corresponden a X_i cercanas. En este caso k ejerce de parámetro de suavizado.

Estos dos tipos de clasificación no paramétrica, en el contexto de regresión con dimensión finita y bajo ciertas condiciones, gozan de la propiedad de *consistencia universal débil*. Esto significa que el estimador tipo núcleo cumple $E||\hat{\gamma(X)} - \gamma(X)||^2 \to 0$ y el estimador de vecinos más próximos asegura, con una probabilidad 0, la existencia de dependencia entre los datos. En nuestro caso, que trabajamos con dimensión infinita, estos estimadores pierden esta propiedad. Sin embargo, existen otras condiciones de consistencia específicas para los datos funcionales. Para poder afirmar que un estimador es consistente universal y débilmente, el error de clasificación $L_n = P(g_n(X) \neq Y|(X_1, Y_1), ...(X_n, Y_n))$ tiene que cumplir:

$$L_n \quad \to \quad L^* \text{ (en probabilidad)} \qquad \text{ó} \qquad E(L_n) \to L^*$$

Tal y como remarca Baíllo et al (2008), con el estimador de vecinos más próximos, si tenemos un espacio métrico separable $\mathcal{X}$, $k \to \infty$ y $k/n \to 0$, se consigue que $E(L_n) \to E(L^*)$. Lo que a su vez hace que se cumple la siguiente condición de *Besicovich* [18]:

$$\frac{1}{P_x(B(X,\delta))} \int_{B(X,\delta)} |\gamma - \gamma(X)| dP_X \quad \overrightarrow{p} \quad 0$$

cumpliéndose también $\delta \to 0$ y siendo P_X la distribución de X.

Al margen de las formas de obtener reglas de clasificación expuestas antes, también existen métodos que se apoyan en las profundidades. El proceso de clasificación basado en profundidades consiste en calcular la profundidad de la nueva observación (x_{new}) en las submuestras que tengamos $P_0, ..., P_W$ y, posteriormente, asignarlo a la submuestra donde obtiene mayor profundidad. Normalmente los métodos que se basan en las profundidades fallan en casos dónde las poblaciones están anidadas [19] o son muy heterogéneas. Se puede ver que Li et al (2012) propone una solución a este problema basándose en las profundidades a través de los DD-plots. El funcionamiento de este clasificador es el siguiente.

Imaginémonos que estamos en un caso donde solo hay dos clases de datos; clase R y clase P. Partimos de dos muestras aleatorias $\{X_1, ...X_m\}$ y $\{Y_1, ...Y_n\}$ de las clases R y P. Si las clases son similares el DD-plot debería estar concentrado sobre la linea de 45 grados. Sin embargo, si las clases son diferentes, el DD-plot mostrará un alejamiento de la linea de 45 grados. Los gráficos o DD-plots, mostrados en las figuras 3.2 y 3.3, muestran, en este caso, las profundidades $(D_1(x), D_2(x))$ siendo $D_i(x)$ la profundidad del punto x respecto a los datos del grupo i. El DD-plot es un gráfico que relaciona el espacio $\mathcal{X}$, donde se definen los datos, y R^2:

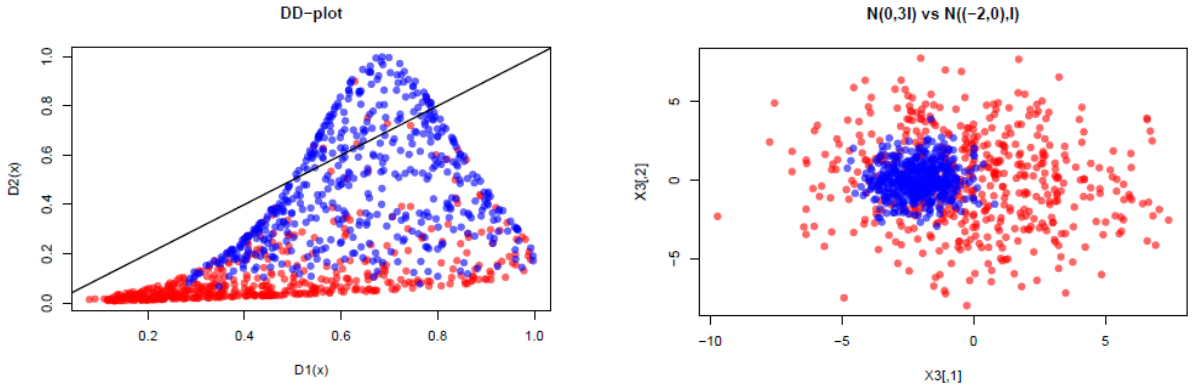
$$\begin{gathered} \mathcal{X} \to R^2 \\ x \to (D_1(x), D_2(x)) \end{gathered}$$

[18] $\overrightarrow{p}$ significa convergencia en probabilidad.
[19] Poblaciones formadas por conglomerados dentro de otros conglomerados.

De esta forma, aunque trabajemos en un entorno de R^4 o R^5 siempre podremos graficar en R^2. La regla más básica en este contexto es asignar una nueva observación al grupo que obtenga la mayor profundidad para esa observación (Maximun depth , MD), pero tiene muchas limitaciones. Para solucionar esto se propone:

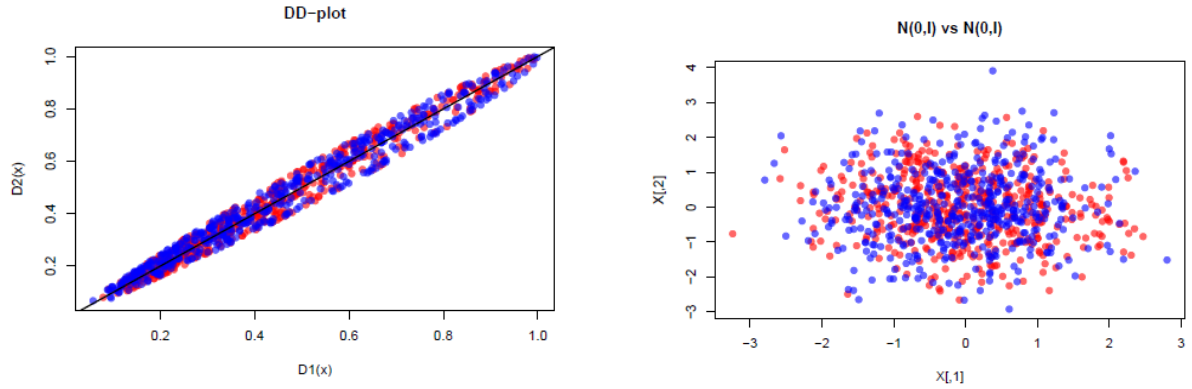
$$\mathcal{X} \to R^g$$
$$x \to \mathbf{d} = (D_1(x), ..., D_g(x))$$

siendo $D_k(x)$ la profundidad de x respecto al grupo $k = 1, ...g$. El clasificador DD^G comprime la información de y_i, x_i en un espacio real de dimensión $(g+1)$ con la forma $y_i, D_1(x_i), ..., D_g(x_i)$.



(a) DD-plot. (b) Gráfico de dispersión.

Figura 3.2: Ejemplo de un DD-plot y un gráfico de dispersión con clases diferenciadas.



(a) DD-plot. (b) Gráfico de dispersión.

Figura 3.3: Ejemplo de un DD-plot y un gráfico de dispersión con clases no diferenciadas.

El clasificador basado en DD-plots destaca por diversas razones:

- Existen muchos métodos de clasificación disponibles.
- El gráfico proporciona información útil sobre lo que está ocurriendo; como por ejemplo, que profundidades son más influyentes.
- Posible reducción de las dimensión del problema; esencial en el contexto de datos funcionales.
- No es importante cómo de complejo es el espacio analizado, solo que una función de profundidades pueda ser definida; por ejemplo, datos funcionales mutivariantes (MFD: $(\mathcal{X} = \mathcal{X}_1 \times ... \times \mathcal{X}_p)$).
- Posibilidad de trabajar con más de una variable funcional.

Por estos motivos, en nuestro análisis, concretamente para la clasificación supervisada, utilizaremos el clasificador en base a los DD-plots; utilizando la profundidad de *Random Projection* (Proyecciones Aleatorias) y el clasificador DD^2. Li et al (2012) plasma el mejor rendimiento de este clasificador en relación a los demás mencionados. Concretamente, calcula el grado G de DD^G por medio de validación cruzada, pero nosotros aquí nos quedamos con $G = 2$ por simplificación.

3.5.2. Clasificación no supervisada (clustering)

Tanto el análisis cluster como la clasificación no supervisada, en general, se basa en ordenar los datos en grupos en función de la similitud de los datos. Esta tarea es esencial, ya que puede proporcionarnos más información que la propia diferenciación en grupos; podemos detectar outliers, relaciones previamente pasadas por alto o conseguir información acerca la dimensión de los datos. En un análisis cluster no existe ninguna especificación sobre el tamaño o el numero de grupos que se crearan en base a la muestra. La formación de grupos solo se basa en la similitud de los datos, y el número óptimo de grupos se determina normalmente al margen. Este hecho es el que hace del análisis cluster algo complejo.

La mayoría de los algoritmos de cluster se basan en las diferencias a pares de los datos que queremos agrupar. Existen muchas formas de calcular estas diferencias entre los datos, y la forma óptima dependerá del tipo de datos con los que estamos trabajando (discretos, continuos, factor,etc). Por otro lado, estas diferencias suelen medirse a través de algún tipo de distancia. Si estamos trabajando con datos funcionales la manera más eficiente de medir las disparidades entre dos funciones $y_i(t)$ y $y_j(t)$, medidas en un soporte como $[0, T]$, es la distancia al cuadrado $L2$ entre las dos curvas (puede consultarse en Ferreira et al (2009)):

$$d(i, j) = \int_0^T [y_i(t) - y_j(t)]^2 dt$$

En este apartado vamos a hablar del análisis cluster jerárquico[20]. Dentro de este tipo de métodos de agrupar datos tenemos dos variedades: cluster aglomerativo y cluster divisivo. Nosotros nos centraremos en el cluster jerárquico aglomerativo ya que es el que llevamos a cabo con nuestros datos. Este tipo de técnica parte de la premisa de que cada dato es un cluster, y luego, en sucesivas uniones va juntando los datos más similares hasta que todo el conjunto es un grupo. Aún así, para averiguar que grupos o datos deberían ir juntándose, Ferreira et al (2009) nos muestra varias formas de proceder (teniendo en cuenta que $d(i, j)$ es la distancia entre el i-ésimo objeto y el j-ésimo th objeto):

- Single linkage: junta grupos en base a la mínima distancia entre dos objetos de dos grupos; esa distancia se calcula de la siguiente forma:

$$d_s(R, Q) = \text{mín}_{i \in R, j \in Q}\, d(i, j)$$

- Complete linkage: une grupos en base a la máxima distancia entre dos objetos de dos grupos; las distancias se definen por:

$$d_C(R, Q) = \text{máx}_{i \in R, j \in Q}\, d(i, j)$$

- Average linkage: crear los grupos en función de la distancia promedio de todos los objetos en un grupo con los objetos del otro grupo; la distancia es definida como:

$$d_A(R, Q) = \frac{1}{|R||Q|} \sum_{i \in R, j \in Q} d(i, j),$$

  dónde $|R|$ es el número de objetos en el cluster R.

- Método de Ward: es similar a los métodos linkage pero no utiliza distancias para agrupar los objetos. Se calcula la suma total de errores cuadrados (SSE), dentro del grupo, para decidir que dos grupos se unirán en cada paso del algoritmo. El SSE se calcula de la siguiente manera:

[20] Existe un número creciente de clases anidadas; clases dentro de otras clases más grandes.

$$SSE = \sum_{i=1}^{K} \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_{i.})'(y_{ij} - \bar{y}_{i.}),$$

siendo y_{ij} el objeto j en el cluster i y n_i el número de objetos en el cluster i.

Se han realizado muchos estudios para intentar averiguar si existe un método que presente un rendimiento por encima de los demás de manera uniforme. La conclusión es que no existe una regla de clasificación que sea mejor que las demás con independencia de los datos. En Ferreira et al (2009) se ha llevado a cabo un estudio de simulación en diferentes escenarios en base al indice Rand (presentado a continuación) para decidir que método reporta mejores resultados.

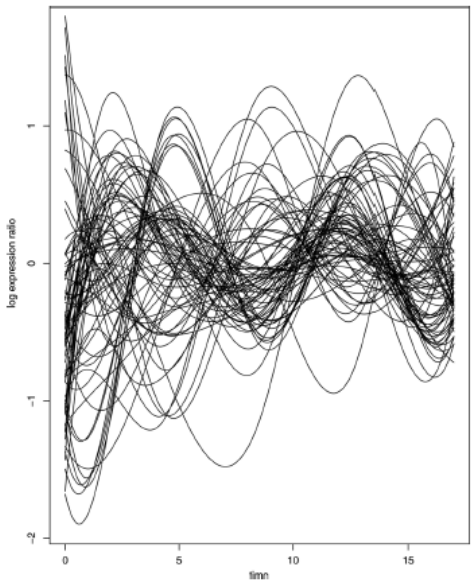
$$Rand = \frac{N_{00}+N_{11}}{N_{00}+N_{01}+N_{10}+N_{11}}$$

dónde N_{00} representa el número de pares de objetos ambos agrupados correctamente en diferentes grupos, N_{01} presenta el número de pares de objetos que no fueron agrupados en el mismo grupo pero deberían, N_{10} representa el número de pares de objetos que deberían estar en distintos grupos pero están en el mismo y N_{11} representa el número de pares de objetos que fueron situados correctamente en el mismo grupo.

Este índice nos reporta una medida de eficiencia entre la verdadera estructura de grupos y el output producido por un determinado algoritmo de cluster; apropiado para comparar diferentes métodos. Tal y como prueba Ferreira et al (2009), en general, el método de Ward siempre da mejores resultados, obteniendo la media más alta del índice Rand en la mayoría de los casos, excepto cuando existen uno o dos grupos grandes y unos pocos pequeños donde el método average linkage lo hace mejor. También se observa que el método complete linkage obtiene buenos resultados pero nunca los mejores. El método single linkage, por su parte, siempre reporta los peores resultados excepto en determinados casos. Por estos motivos, en un contexto de datos funcionales no sería apropiado el uso de los métodos single o complete linkage. Si no se sospecha de que estamos en la situación concreta donde el método average linkage funciona mejor, los resultados nos llevan a usar el método de Ward.

En las clasificaciones utilizando técnicas de cluster jerárquicas es normal cometer algún error en la elección del número de grupos en el proceso; ya que en este tipo de técnicas este número debe ser proporcionado por el investigador. Ferreira et al (2009) también lleva a cabo un estudio donde se compara las distintas formas de llevar a cabo el cluster en situaciones de subestimación y sobreestimación del número óptimo de grupos. Los resultados son similares a los presentados en el caso general. El método de Ward es el que obtiene mejores resultados; mayor precisión que los demás métodos en la mayoría de los casos. Aunque cabría destacar que en situaciones donde existen grupos de diferente tamaño y hay posibilidad de sobrestimar el verdadero número de grupos, el método average linkage es el que reporta mejores resultados.

El estudio llevado a cabo en Ferreira et al (2009) está basado en 78 observaciones de genes de levadura medidas a lo largo del tiempo con una transformación logarítmica (figura 3.4). Partiendo de la situación dónde se conoce la verdadera estructura (por grupos) de los datos, se efectúa una comparación del rendimiento de las distintas formas de llevar a cabo del cluster jerárquico. Los resultados obtenidos los mostramos a continuación (figura 3.5):



Suspected clustering structure for yeast gene data

Cluster name	Cluster size	Observations
G1	13	1–13
S	39	14–52
S/G2	8	53–60
G2/M	7	61–67
M/G1	11	68–78

S = Synthesis, M = Mitosis, G = Gap.

(a) Datos de genes de levadura. 78 curvas suavizadas y centradas. (b) Cuadro con la verdadera estructura de grupos de los genes.

Figura 3.4: Resultados de la simulación para realizar una diagnosis de los diferentes métodos cluster.

Clustering results for each hierarchical method (yeast gene data)

Method	Clustering structure	Rand index
Assumed structure	{1–13}, {14–52}, {53–60}, {61–67}, {68–78}	N/A
Ward's method	{1–2, 4, 6–11, 13, 70, 74–75}, {5}, {3, 12, 14–52, 61–67, 69, 78}, {54–60}, {68, 71–73, 76–77}	0.7949
Single linkage	{1–4, 6, 8–64, 66–75}, {5}, {7}, {64}, {76–77}	0.4006
Complete linkage	{1–4, 6–11, 13, 19, 22, 27, 30, 32, 41–42, 44, 47, 61–62, 65–66, 69–70, 74–75, 78}, {5}, {12, 14–15, 18, 21, 24–26, 28–29, 31, 33–35, 38–40, 43, 45–46, 48, 50, 52}, {16–17, 20, 23, 36–37, 50, 52, 53–60, 63–64, 67}, {68, 71–73, 76–77}	0.6857
Average linkage	{1–4, 6–16, 18–19, 21–53, 61–63, 65–67, 69–70, 74–75, 78}, {5}, {17, 20, 54–60}, {64}, {68, 71–73, 76–77}	0.5951

Note: Clustering partitions defined by the observation numbers in braces in the table.

Figura 3.5: Resultados de cada método de cluster y cálculo de su rendimiento (índice de Rand).

Los resultados corroboran lo expuesto antes; el método de Ward es el más conveniente si nos encontramos en un contexto de datos funcionales, seguido por el complete linkage y average linkage. Esto lo observamos en los índices de Rand obtenidos por cada método en la tabla de la figura 3.5. En esta tabla también se plasma el pobre rendimiento del método single linkage, que no llega a un índice de Rand de 0.5. Por todos los motivos expuestos, entre todos los métodos que tenemos a nuestra disposición para realizar el análisis clúster, nosotros nos hemos decantado por el método de Ward.

Capítulo 4

Datos Analizados

Para el desarrollo de este TFM, de un enfoque práctico, contamos con datos sobre distintas variables medidas en diversas modalidades de bombas de agua y que fueron facilitados por la empresa EcoMT; concretamente utilizando el sistema de telegestión OTEA. A través de esta página web podemos acceder a todas las tiendas controladas por EcoMT y, del mismo modo, a una gran variedad de datos sobre cada una de ellas. Desde información sobre el consumo de la tienda o de la maquina hasta las diferentes temperaturas de consigna de las tiendas. Nosotros, para desarrollar nuestro algoritmo y lograr los objetivos del TFM, nos centramos en las variables que mostramos a continuación. Pero antes, destacar que estas variables, como ya dijimos, serán analizadas en formato funcional. Esto quiere decir que cada día del año equivaldrá a una curva de 96 valores (un valor cada 15 minutos) que nos resumirá el comportamiento de la maquina durante el día. En definitiva, trabajaremos con los datos transformados de tal forma que tendremos, al comienzo del análisis, 365 curvas diarias de cada una de las siguientes variables:

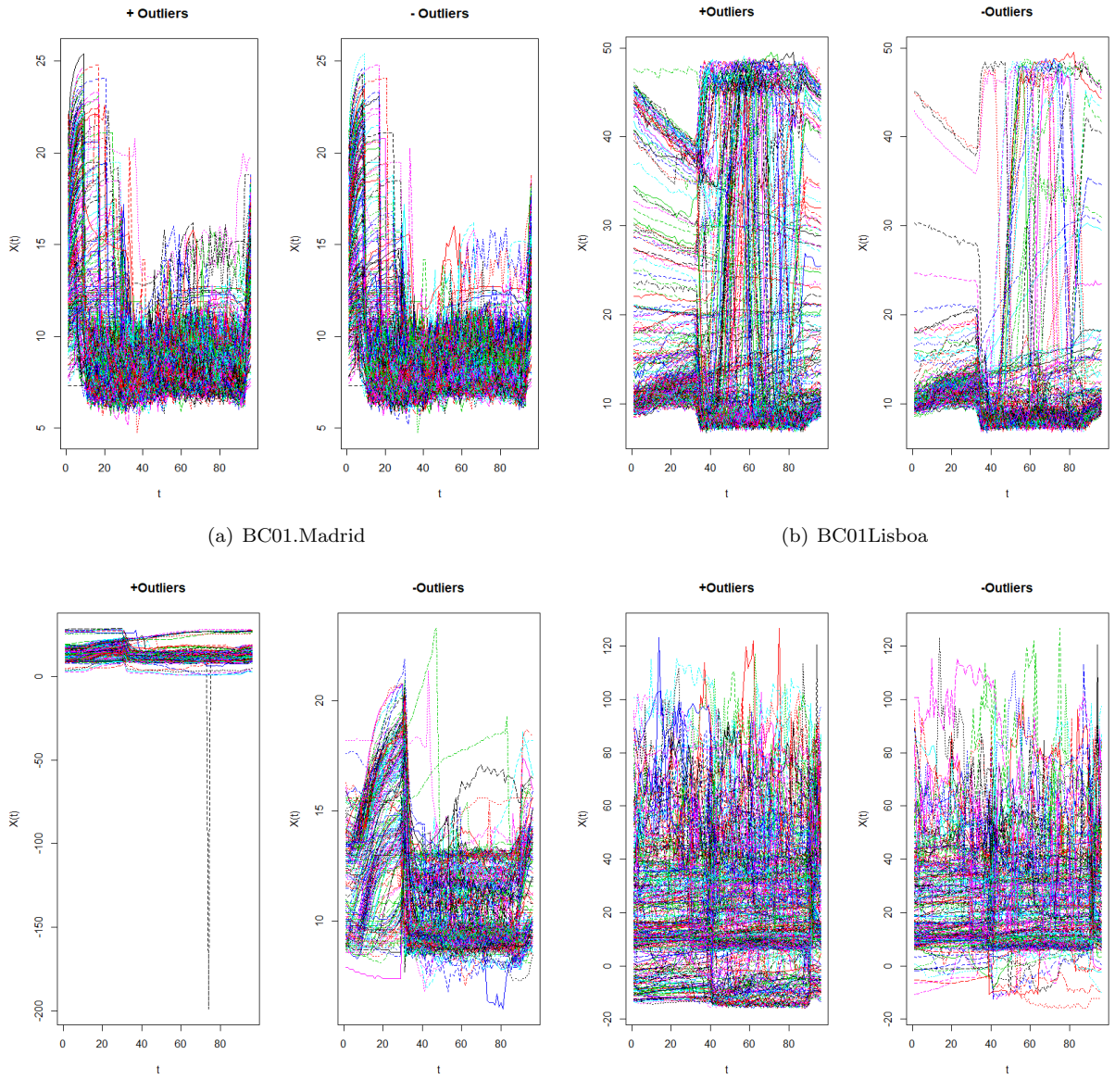
- *Temperatura de impulsión*: variable que toma valores cada 15 minutos de la temperatura a la que la bomba impulsa el agua.
- *Temperatura de retorno*: variable que toma valores cada 15 minutos de la temperatura a la que retorna el agua a la bomba.
- *Código de encargado/a*: variable binaria que nos reporta 0 cuando la tienda esta cerrada, y nos reporta 1 desde el momento en que el/la encargado/a llega a la tienda hasta el momento en el que el/la encargado/a deja la tienda.
- *Incidencias*: variable creada en base a datos en los que están anotadas las incidencias reales que sucedieron en las diferentes tiendas. Estas incidencias están anotadas cronológicamente, por este motivo la variable *Incidencias* tendrá unicamente un 1 en los momentos quinceminutales donde la incidencia ocurrió o acaba de ocurrir y fue anotada. En los demás momentos habrá un 0.

Tanto la temperatura de impulsión como la temperatura de retorno son dos variables clave en nuestros algoritmos. En base a estas variables, en formato funcional, creamos los grupos de bombas según su patrón de funcionamiento, realizamos la posterior clasificación supervisada e, incluso, estudiamos la existencia de una relación entre la evolución de estas variables y la detección de incidencias. A pesar de ser tan importantes, realizar una diferenciación de bombas con ellas es una tarea difícil. Las temperaturas de impulsión y de retorno de una misma maquina, entre sí, son similares; pero la heterogeneidad entre las bombas a analizar es enorme. Mas adelante presentamos los datos de varias maquinas de distintas tiendas elegidas al azar para mostrar esta gran heterogeneidad, que dificulta la obtención de datos que de verdad aporten información significativa y, paralelamente, entorpece la labor de clasificación. Hay que tener en cuenta que, para poner a prueba el algoritmo, hemos trabajado con dos muestras: una centrada en tiendas de la península ibérica (básicamente España), y otra centrada en tiendas del centro y norte de Europa.

- Muestra de la Península Ibérica: En esta muestra tenemos 26 bombas de agua de diferentes provincias de España y alguna de Portugal. Los tipos de bombas que podemos encontrarnos en esta muestra son: BC, B, UE, UI, UC, IE, EN, EF CAL, C o GF.

- Muestra de Europa: En esta muestra contamos, en cambio, con 28 bombas de agua pertenecientes a tiendas del centro y del norte de Europa; bombas de países como Países Bajos, República Checa o Francia. Entre las bombas que entraron en nuestra muestra existen varios tipos de bombas: BC, B, Boiler, UE, UEC, UEG, UH, UC, UP, EN, EF, C, CAL, CU, GF, GE, GD, NRL 500 HE.

Las diferentes bombas no tienen una traducción directa, ya que en cada tienda reciben etiquetas diferentes. A pesar de los nombres o etiquetas que tienen, principalmente, son bombas que tienen como función principal *calentar* o *enfriar*. En algún caso, una bomba puede variar su función a lo largo del año; ciertos días calienta y ciertos días enfría[21]. A continuación, en las figuras 4.1 y 4.2, presentamos 8 bombas de agua a través de sus curvas diarias de temperaturas de impulsión; antes y después de eliminar outliers:



(a) BC01.Madrid

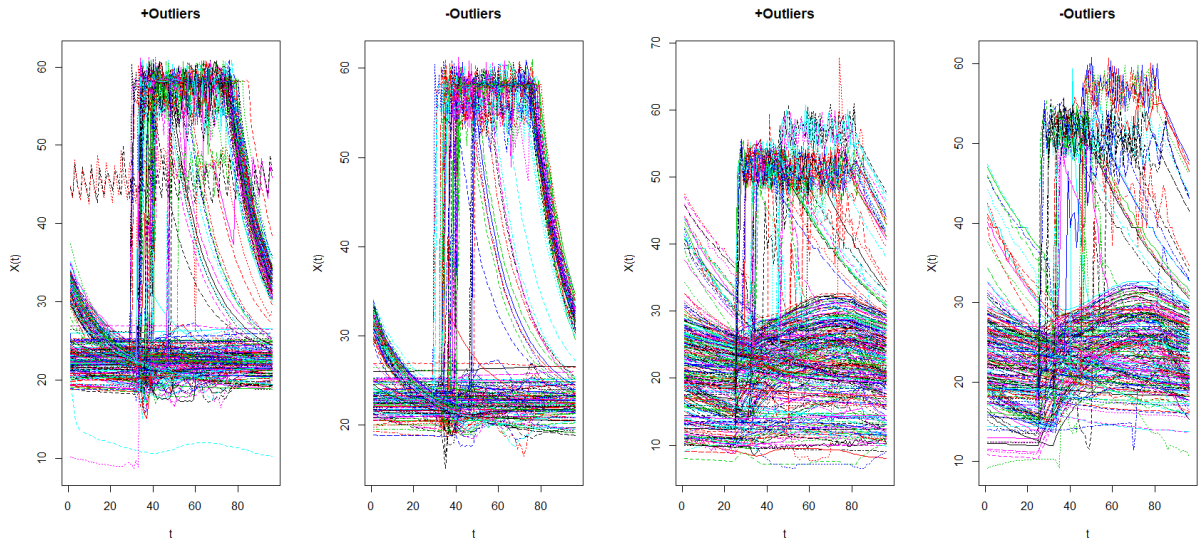
(b) BC01Lisboa

(c) IE02Barcelona1

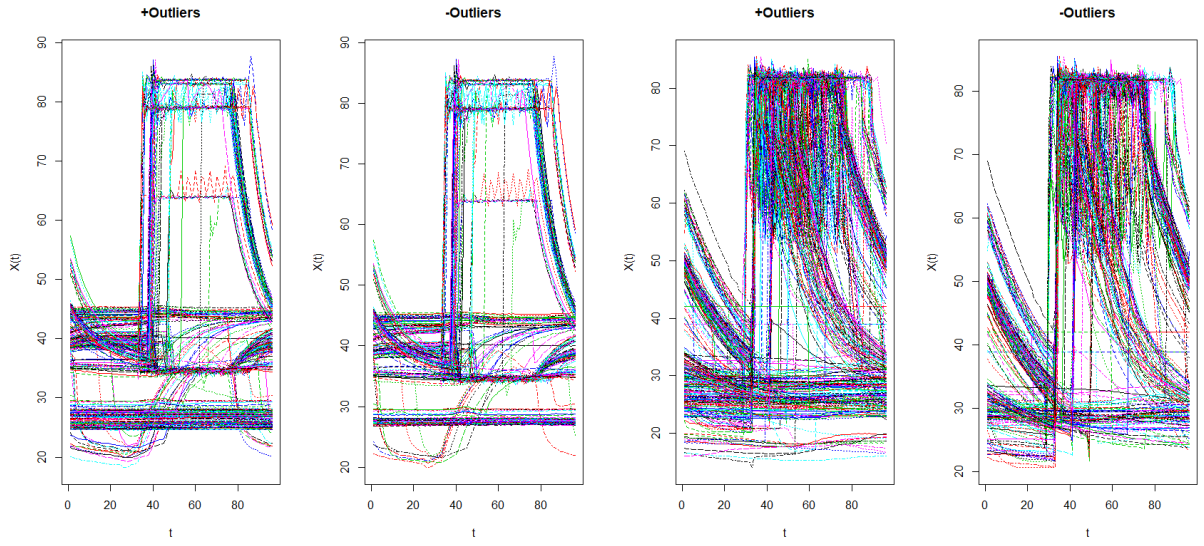
(d) UE01Valencia

Figura 4.1: Temperaturas de impulsión antes y después de haber eliminado Outliers de 4 bombas de la muestra 1.

[21] Bombas multifunción.



(a) BO01Gante (b) C01Viena



(c) C01Amsterdam (d) C01Amersfoot

Figura 4.2: Temperaturas de impulsión antes y después de haber eliminado Outliers de 4 bombas de la muestra 2.

En estos gráficos mostramos las temperaturas de impulsión en formato funcional; una curva por día. Solo mostramos las temperaturas de impulsión ya que las temperaturas de retorno se comportan de una forma muy similar, y, en este caso, nos aportarían la misma información que las temperaturas de impulsión. En cada bomba vemos diferentes tipos de curvas: unas mas suaves, otras haciendo ondas, otras con grandes saltos, etc.. Aquí mostramos todos los días del año, pero claramente no todos estos días son dignos de analizar; ya que hubo varios festivos a lo largo del año o días que simplemente la tienda no abrió. Encontrar y eliminar estos días no hábiles, junto los outliers que sean necesarios, es una tarea muy costosa debido a que en cada tienda abren días distintos, las maquinas se comportan de forma diferente y la curvas de temperaturas también son diferentes. Además, como se observa en los gráficos, con una búsqueda clásica de outliers[22] conseguimos eliminar varios días *raros*; pero no todos los que deberíamos. A pesar de esto, con técnicas cluster y eliminando grupos de días en los cuales las curvas funcionales tuvieron muy poca variabilidad[23] en relación a otros grupos, conseguimos aislar los días útiles y analizarlos posteriormente. Si nos fijamos en las dos muestras, vemos que en las bombas de la muestra de Europa se aprecian muy bien curvas

[22] Utilizando alguno de los métodos explicado en la sección *Outliers* en una única iteración.

[23] Los días en los que la tienda no abrió. normalmente, tienen un comportamiento muy suave ya que las maquinas no se pusieron en marcha.

muy planas que se corresponderían con días donde las tiendas no abrieron. En España, por ejemplo, en la maquina de la tienda de Madrid no se consigue distinguir ningún día *plano*; esto es porque esta tienda no cierra prácticamente ningún día al año. A grandes rasgos, vemos un comportamiento totalmente heterogéneo entre las curvas de cada bomba. A causa de todo esto, extraer la información que nos interesa de los días donde realmente existió actividad en la tienda es una tarea muy laboriosa, que explicaremos en el apartado 5.2.

Por otro lado, nos quedan las variables código de encargada/o e incidencias; ambas también muy importantes para los objetivos fijados en este TFM. El código de encargada/o lo utilizamos para conseguir una *rejilla* de puntos que plasme el horario de cada bomba, y a la vez sea un horario estándar que permita analizar todas las bombas sin perderme momentos de funcionamiento. En caso contrario, podría ocurrir que el valor del primer momento analizado de una determinada bomba no sea cuando comienza a funcionar, sino que, por diferentes razones, ya llevaba funcionando horas. Esto es esencial cuando queremos analizar bombas con entornos diferentes, ya que cada una seguirá un horario variable y totalmente diferente. De esta forma, a raíz de esta variable, podemos obtener una *rejilla* de 60 valores para cada bomba donde el primer valor siempre es el momento donde la bomba comienza a funcionar con más frecuencia, y el último, el momento cuando deja de funcionar más habitualmente.

Por último, la variable incidencias es la que informa de en qué momentos ocurrieron incidencias reales, alguien se percató y fueron anotadas. Con incidencias reales nos referimos a que una enfriadora esté impulsando a una temperatura muy alta cuando el entorno requiere que enfríe, o que un equipo de clima deje de funcionar por alguna avería. Por lo tanto, con esta información podemos obtener las curvas funcionales de las temperaturas de impulsión y de retorno previas a tales hechos y analizar si realmente las curvas previas a esas incidencias son distintas de las curvas donde no ocurrió nada.

Capítulo 5

Algoritmos

A continuación vamos a explicar los tres algoritmos, creados en este trabajo de fin de máster, con el objetivo de facilitar y adelantar información sobre el comportamiento de bombas de agua. El algoritmo principal (*Clasificador*) permite realizar una clasificación de una determinada muestra de bombas de agua según su patrón de funcionamiento (clasificación no supervisada). A raíz de esto, también se ha conseguido desarrollar el algoritmo *Clasificador.S*. Con cierta información proporcionada por *Clasificador*, este nuevo algoritmo puede llevar a cabo una clasificación supervisada (en función de los grupos creados con el algoritmo principal) de nuevas bombas de agua. Por último, para poder anticipar de manera efectiva posibles incidencias en las máquinas, se ha creado el algoritmo *Detección*. Este algoritmo, a través del análisis de las curvas de temperaturas del agua actuales y viendo si el comportamiento de la bomba de agua se está alejando de lo esperable, predice posibles incidencias en un futuro inmediato.

Antes de pasar a explicar con calma cada uno de los algoritmos, vamos a presentar las nuevas funciones artificiales que se crearon para utilizar en los algoritmos y que llevan a cabo tareas imprescindibles para su buen funcionamiento.

5.1. Funciones

- *Inflect*: busca los óptimos locales, si los tiene, de alguna función. En concreto, nosotros la utilizamos para encontrar mínimos locales en la función de densidad de las distancias entre las curvas diarias de temperaturas a analizar de cada bomba.
- *Cuantiles*: calcula el tercer cuartil (percentil 75 %) del vector que se le introduzca.
- *Cuantiles2*: calcula el primer cuartil (percentil 25 %) del vector que se le introduzca.
- *Cortes*: convierte un vector de valores quinceminutales en una matriz donde las filas representan los momentos del día (96 concretamente, cada 15 minutos) y las columnas los días.
- *Cortes2*: Realiza una tarea similar a la de *Cortes*; lo que cambia en este caso, es que trabajamos con 60 momentos diarios en vez de 96. El día estará dividido esta vez en 60 instantes. Este cambio es a causa de la estandarización de los horarios, que explicaremos más adelante.
- *Arreglo*: analiza los datos para que no existan valores extremos (valores por encima o por debajo de unos límites lógicos). Analiza día a día, tanto para temperaturas de impulsión como de retorno, la existencia de mediciones por encima de los 65 grados centígrados y por debajo de los 0 grados centígrados. Se procederá a eliminar los días que tengan más de 20 valores extremos o más de 10 consecutivos; en caso de que existan únicamente un par de ellos serán substituido por NAs.
- *Comprobación*: De manera parecida a la función *Arreglo*, esta función estudia la existencia de valores faltantes en los datos. Si a lo largo de un día existen demasiados datos faltantes (> 15) o muchos consecutivos (> 5) distorsionarán nuestro análisis; por lo tanto, esos días serán eliminados.
- *Aproximación*: interpola los valores faltantes que existen día a día. Por ejemplo, un valor NA será substituido por la media entre el valor siguiente y el anterior. Téngase en cuenta que diariamente solo podrán existir unos cuantos valores faltantes, ya que los excesivos fueron suprimidos por la función comprobación.

- *Getmode*: obtiene la moda de un conjunto de datos.
- *Estandar*: necesita diversos argumentos para crear los horarios estandarizados. Primero necesita una *rejilla* de 60 valores que tenga en cuenta el momento quinceminutal más habitual de apertura y de cierre de cada tienda. Además, también necesita pesos que serán utilizados con las variables del estudio y que se calcularon en base a la proximidad de los valores de la *rejilla* a los valores enteros (inferior o superior). Introduciendo estos argumentos esta función reporta un horario estándar para cada bomba. Estos horarios permitirán analizar tiendas de cualquier entorno ya que siempre tendremos 60 valores de *rejilla* que contendrán los momentos de apertura y de cierre de cada tienda.

5.2. *Clasificador*: clasificación no supervisada

Dentro de este algoritmo, que se centra en crear grupos de bombas en función de su comportamiento, se pueden distinguir varias fases:

1. Primera fase: Agrupamiento de bombas en calentadoras, enfriadoras y multifunción[24] según los datos de la diferencia entre las temperaturas de impulsión y las de retorno.
2. Segunda fase: En base a técnicas cluster, se realiza una búsqueda de días que, efectivamente, la tienda estuvo abierta y la maquina funcionó con normalidad. Esto equivale a decir que se eliminan de nuestro análisis días donde la información que tenemos no es buena.
3. Tercera fase: Utilizando las curvas medianas funcionales, relativas a las temperaturas de impulsión y a las de retorno de cada bomba de agua, y a través de técnicas cluster, agrupamos las bombas según su patrón de funcionamiento.

Además, los argumentos que debemos pasarle para que comience a trabajar son los siguientes: una lista con las diferentes bombas de agua en matrices y con las variables en columnas (*Bombas*), una lista con los nombres de todas las bombas (*names*), el número de repeticiones bootstrap para la búsqueda de outliers (*nb*), una *rejilla* de valores con la longitud deseada para calcular la consiguiente *rejilla* estadarizada con los horarios de cada bomba (*regg*) y un valor para el cuantil, que calcularemos sobre las distancias entre bombas, y que servirá para hacer automática la creación de grupos con el cluster (*corte*).

5.2.1. Primera fase

Como ya se mencionó más arriba, partimos de archivos con los datos de las diferentes variables distribuidas en columnas. Las columnas se distribuyen de la siguiente manera: fecha, temperatura de impulsión, temperatura de retorno, código de encargada/o e incidencias. Lo primero que hace el algoritmo es crear una nueva lista (*Bombas2*) con todas las bombas analizadas pero solo con las variables numéricas de las temperaturas.

Antes de llevar a cabo la primera clasificación de las bombas es necesario estandarizar los horarios de cada bomba; cada una está ubicada en una tienda y cada tienda abre y cierra en un horario diferente. Para ello se utiliza la variable *código de encargada/o*; después de haber estudiado y solucionado la existencia de valores NAs. Esta tarea se basa en averiguar los momentos quiceminutales donde es más probable que abra y que cierre la tienda; esto, el algoritmo, lo consigue buscando el momento de apertura que más se repite (y lo mismo con el momento de cierre). De esta forma obtiene valores de inicio y final para la *rejilla* de valores que se estudiará en cada bomba. El siguiente paso es calcular los valores intermedios, que vayan desde el punto inicial al final y, además, sumen 60 valores. En este momento surge la necesidad de calcular unos pesos debido a que muchos valores de la *rejilla* no son enteros[25]. Con diversas operaciones obtiene una rejilla y unos pesos para cada bomba; datos necesarios para llevar a cabo la estandarización de horarios con la función *estándar*.

Tras obtener los intervalos de tiempo donde mejor se aprecia el patrón de funcionamiento de cada bomba, el algoritmo pasa a trabajar con la lista *Bombas2*. Independientemente de la existencia de valores faltantes a lo largo del periodo de tiempo analizado, también existen mediciones de temperatura erróneas; como por ejemplo, temperaturas de impulsión de más de 3000 grados o en niveles negativos. Para solucionar esto creamos la función *arreglo*. Tras el uso de la función *arreglo* es necesario volver a realizar un análisis de valores

[24] Bombas que a lo largo del periodo analizado presentan días donde realizan la función de calentadoras y, otros, función de enfriadoras.

[25] Pesos para calcular el valor de las variables (impulsión y retorno) en función del valor de la *rejilla*. Si el valor es decimal y más cercano al entero superior, se le dará un peso mayor para el valor de la variable del momento quinceminutal siguiente.

faltantes con la función *comprobación* sobre las variables temperatura de impulsión y temperatura de retorno [26]. Por último, aplicando la función *aproximación*, el algoritmo hace que desaparezcan los últimos valores faltantes que podía haber en la muestra.

Tras un primer análisis de los datos, eliminando posibles datos faltantes y/o datos erróneos, el siguiente paso es realizar una primera clasificación de las bombas de agua en: bombas calentadoras, bombas enfriadoras y bombas multifunción. Los pasos a seguir por el algoritmo para realizar esta división son los siguientes:

1. Aplicación de la función *cortes* para crear matrices con los datos de las diferentes variables; las columnas pasan a ser días y las filas momentos del día.

2. Aplicación de la función *estándar* para calcular los horarios estandarizados de cada bomba de la muestra.

3. Cálculo de la variable diferencia entre la temperatura de impulsión y la temperatura de retorno. Esta variable es esencial para ver si una bomba está calentando o enfriando; por ejemplo, si está calentando la temperatura de impulsión tiene que ser mayor que la temperatura de retorno.

4. Conversión de las matrices a formato funcional para llevar a cabo un análisis de outliers. Se buscarán, y posteriormente se eliminarán, días que sean outliers en las temperaturas de impulsión o en las temperaturas de retorno. Se queda con los días que no sobresalen del comportamiento normal de la bomba.

5. Aplica las funciones *cuantiles* y *cuantiles2* día por día a la variable diferencia. Con esto, busca los días que cada bomba está enfriando o está calentando. Para que el algoritmo considere que un día la bomba esté enfriando tendrá que obtener un cuantil del 75 % de la variable diferencia menor de 0. En cambio, para ser candidato a día con función de calentar tiene que obtener un cuantil del 25 % mayor que 0. De esta forma, los días seleccionados serán días que, al menos el 75 % de su duración, está realizando la función propia del grupo al que se asigna.

6. Pasando unos determinados filtros, según la cantidad de días que enfría o calienta, la bomba analizada se introduce en alguno de los 3 grupos básicos presentados arriba. Los filtros son los siguientes[27]:

   - Enfriadora
     a) El numero de días enfriando debe ser mayor o igual al 75 % de los días analizados o, el número de días que está calentando ser mínimo (menos de 3 días).
     b) La mediana de la temperatura de impulsión en los días que presumiblemente está enfriando debe ser menor a 20 grados.
   - Calentadora
     a) El numero de días calentando deben ser más o igual al 75 % de los días analizados o, el número de días candidatos a enfriar ser mínimo (menos de 3 días).
     b) La mediana de la temperatura de impulsión en los días que presumiblemente está calentando debe ser mayor o igual a 20 grados.
   - Multifunción
     Las bombas irán a este grupo si no cumplen los requisitos de las bombas enfriadoras o calentadoras.

Una vez tenemos esta primera clasificación, el algoritmo considera que las bombas que pasaron los filtros para ser *Calentadora* y *Enfriadora* están bien clasificadas. En cambio, el hecho de que el grupo de *Multifunción* este lleno de bombas que simplemente no han cumplido los requisitos previos, hace necesario analizarlas con más calma. Para ello, los candidatos previos a ser días que enfrían y días que calientan, deberán pasar nuevos filtros. La forma de proceder en este caso será la siguiente:

1. Creación de *díasfrio*; lista donde se guardan los días que eran candidatos a enfriar y, además, cumplen que:

   - Impulsan el agua a 18 grados, o menos, durante el 75 % del día o más.
   - Retornan el agua a 22 grados, o menos, durante el 75 % del día o más.

[26] Tal y como se explica en el apartado *Funciones* la función *arreglo también convierte valores extremos en valores NAs.*
[27] Necesario cumplir ambos requisitos.

2. Creación de *díascalor*; lista donde se guardan los días que eran candidatos a calentar y, además, cumplen que:

   - Impulsan el agua a 22 grados, o más, durante el 75 % del día o más.
   - Retornan el agua a 18 grados, o más, durante el 75 % del día o más.

3. Eliminación de la muestra las bombas que se queden sin días para analizar porque no pasaron los filtros o, la suma de los días (calentando y enfriando) que sí los pasaron, es menor de 10.

4. Traspaso de bombas al grupo de calentadoras si los *diasfrio* son menos de 3 y los *diascalor* son más de 10.

5. Traspaso de bombas al grupo de enfriadoras si los *diascalor* son menos de 3 y los *diasfrio* son más de 10.

Tras todos estos pasos el algoritmo finaliza esta primera etapa (esquematizada en la figura 5.1) con tres grupos de bombas de agua bien diferenciados en base a su comportamiento general: bombas enfriadoras, bombas calentadoras y bombas multifunción. A continuación, en las siguientes fases, nos adentraremos en el estudio de estos grupos y profundizaremos en los verdaderos patrones de funcionamiento de cada bomba.

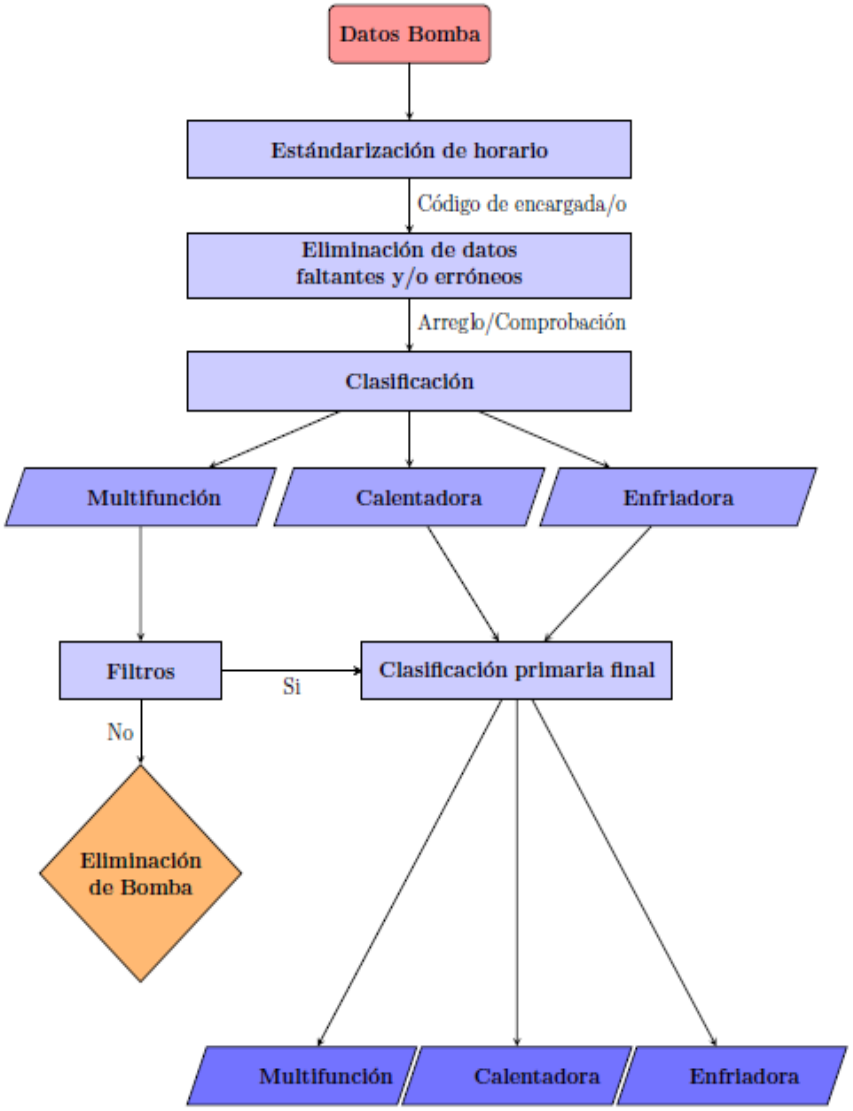


Figura 5.1: Diagrama de flujo de la primera fase del algoritmo.

5.2.2. Segunda fase

En esta segunda fase (resumida en la figura 5.5), el principal objetivo del algoritmo es eliminar de la muestra días, que por diversas razones (temperaturas fuera de rango, temperatura de impulsión demasiado altas para estar enfriando o viceversa, etc...), no son adecuados para nuestro análisis. Dicho de otra manera, el algoritmo seleccionará los días donde podemos extraer información relevante sobre su patrón de comportamiento.

General

Lo primero que hace el algoritmo aquí es observar, día a día, las temperaturas de impulsión y de retorno, diferenciando entre enfriadoras y calentadoras. Tal y como ya se hizo en la primera fase, bomba a bomba se seleccionarán los días que presenten un comportamiento acorde al grupo primario asignado a la bomba:

- Enfriadoras: selección de días que satisfagan las siguientes condiciones:

  1. Cuantil del 75 % de la temperatura de impulsión menor o igual que 18 grados. Días donde, el 75 % de su duración o más, está impulsando a 18 grados o menos.
  2. Cuantil del 75 % de la temperatura de retorno menor o igual que 22 grados. Días donde, el 75 % de su duración o más, está retornando a 22 grados o menos.
  3. Cuantil del 75 % de la variable diferencia menor o igual a 0. Días donde, el 75 % de su duración o más, la bomba estuvo enfriando.

  La diferencia entre la restricción de grados de la temperatura de impulsión y la de retorno es debido al margen que dejamos debido a la suposición de que la temperatura de impulsión debe ser menor que la de retorno; por eso está en el grupo de enfriadoras.

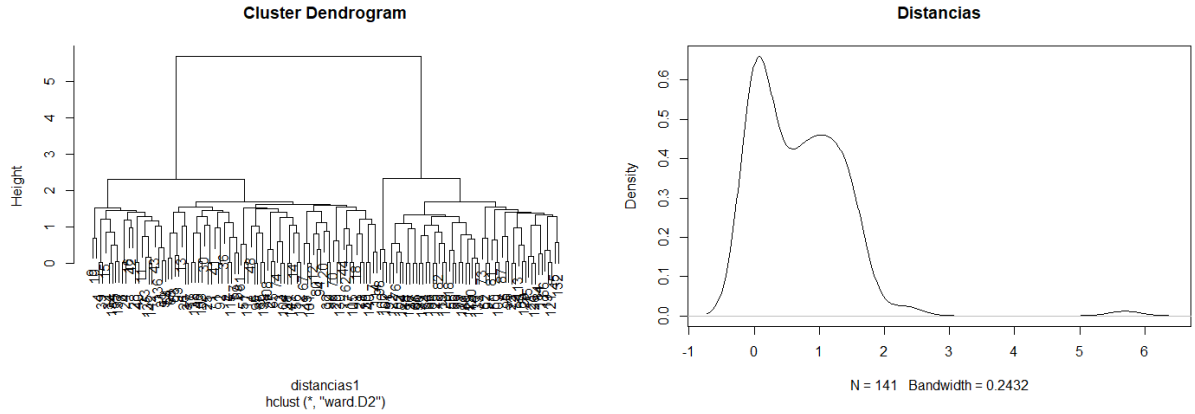
- Calentadoras: selección de días que satisfagan las siguientes condiciones:

  1. Cuantil del 25 % de la temperatura de impulsión mayor o igual que 22 grados. Días donde, el 75 % de su duración o más, está impulsando a 22 grados o más.
  2. Cuantil del 25 % de la temperatura de retorno mayor o igual que 18 grados. Días donde, el 75 % de su duración o más, está retornando a 18 grados o más.
  3. Cuantil del 25 % de la variable diferencia mayor o igual a 0. Días donde, el 75 % de su duración o más, la bomba estuvo calentando.

  En este caso permitimos mayores temperaturas de impulsión ya que estamos con el grupo de calentadoras; impulsión a temperaturas más altas que las de retorno.

El siguiente paso es transformar los datos a un formato funcional; lo cual permite al algoritmo calcular distancias entre las curvas (en nuestro caso días). Esta es la pieza clave de esta fase. Con las distancias entre los días, el *Clasificador*, efectúa un cluster a través de la técnica Ward. La forma automática, de la que dotamos al algoritmo, para primero, crear los grupos de días, y segundo, detectar grupos de días no aptos para nuestro análisis, es la siguiente:

1. Estima la función densidad de las distancias obtenidas en el cluster (figura 5.2).



(a) Dendograma de los días analizados de una bomba aleatoria.

(b) Densidad estimada de las distancias del cluster de la izquierda.

Figura 5.2: Ejemplo de un dendograma de días y una densidad estimada de distancias.

2. Calcula el cuantil 0.90 de las distancias; queremos *cortar* el cluster dejando por debajo, al menos, el 90 % de las observaciones.

3. Busca un mínimo local de la densidad estimada de las distancias, mayor o igual al cuantil 90 % de las distancias. Esto es así porque las modas de esta densidad estimada son los rangos de distancias donde se fusionan, en un mismo grupo, observaciones que antes estaban separadas. Por lo tanto, es lógico coger como umbral automático para la creación de grupos, un mínimo local entre dos máximos locales o modas de las distancias.

4. Corta los grupos del cluster por la distancia calculada en el paso anterior.

5. Elimina los días que estén en un grupo cuyo tamaño es menor a 5 días. Si el cluster separa grupos pequeños de días es porque están alejados del comportamiento *normal* de la bomba. Son considerados una especie de outlier.

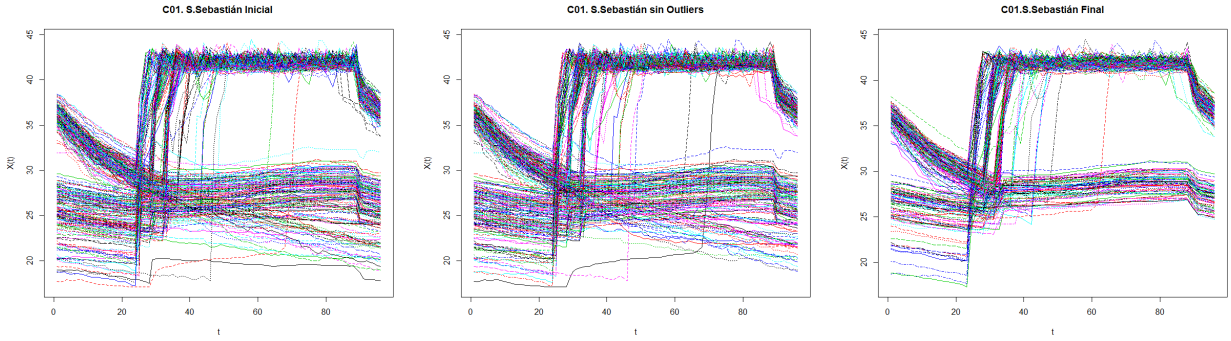
6. Elimina grupos de días cuya variabilidad mediana es muy grande (posibles mediciones extremas). Por medio de un análisis empírico se ha puesto de limite de variación superior 120.

7. Elimina grupos de días que presenten una variabilidad mediana muy pequeña en comparación con el grupo que presenta mayor variabilidad mediana. En este caso, buscando días donde la tienda no abrió[28], el algoritmo eliminará los días cuya variabilidad mediana del grupo sea menor a un 20 % de la variabilidad del grupo que presente la mayor.

8. Se queda con los grupos de días que a estas alturas no han sido eliminados.

Para poder plasmar de forma rápida y sencilla el rendimiento del algoritmo, mostramos la selección de días con los que se queda el algoritmo, en el horario estandarizado, en comparación con una búsqueda de outliers ordinaria (figuras 5.3, 5.4 y 5.6).

[28]Si la tienda no abre las curvas de temperatura serán mucho más suaves; la tienda no esta funcionando.

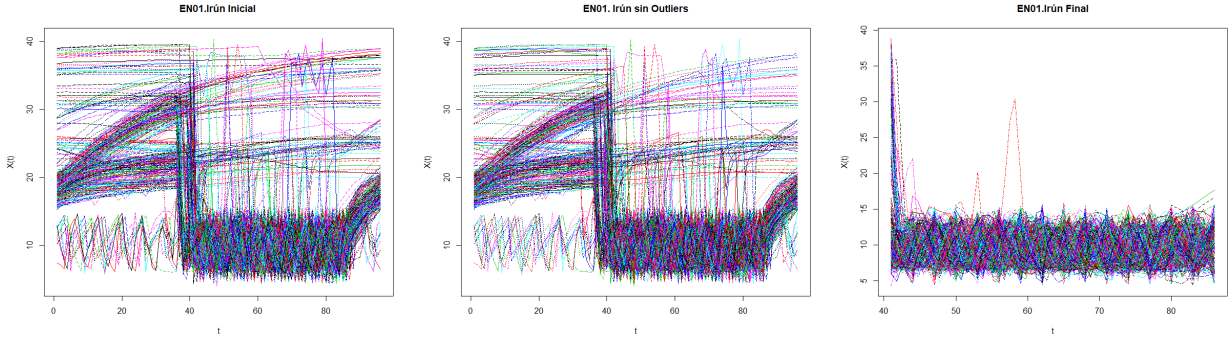


(a) Gráfico de las temperaturas de impulsión diarias tras eliminar unicamente valores fuera de rango (346 días)).

(b) Gráfico de las temperaturas de impulsión diarias tras eliminar outliers de una forma clásica (344 días).

(c) Gráfico de las temperaturas de impulsión diarias con las que trabajará el algoritmo (231 dias).

Figura 5.3: Ejemplo de un análisis de los días significativos para una bomba calentadora.



(a) Gráfico de las temperaturas de impulsión diarias tras eliminar unicamente valores fuera de rango (349 días))

(b) Gráfico de las temperaturas de impulsión diarias tras eliminar outliers de una forma clásica (336 días).

(c) Gráfico de las temperaturas de impulsión diarias con las que trabajará el algoritmo (249 días).

Figura 5.4: Ejemplo de un análisis de los días significativos para una bomba enfriadora.

A lo largo de estos pasos si una determinada bomba se queda sin días para analizar, debido a que se han eliminado demasiados, será eliminada de la muestra. El limite está en 5 días.

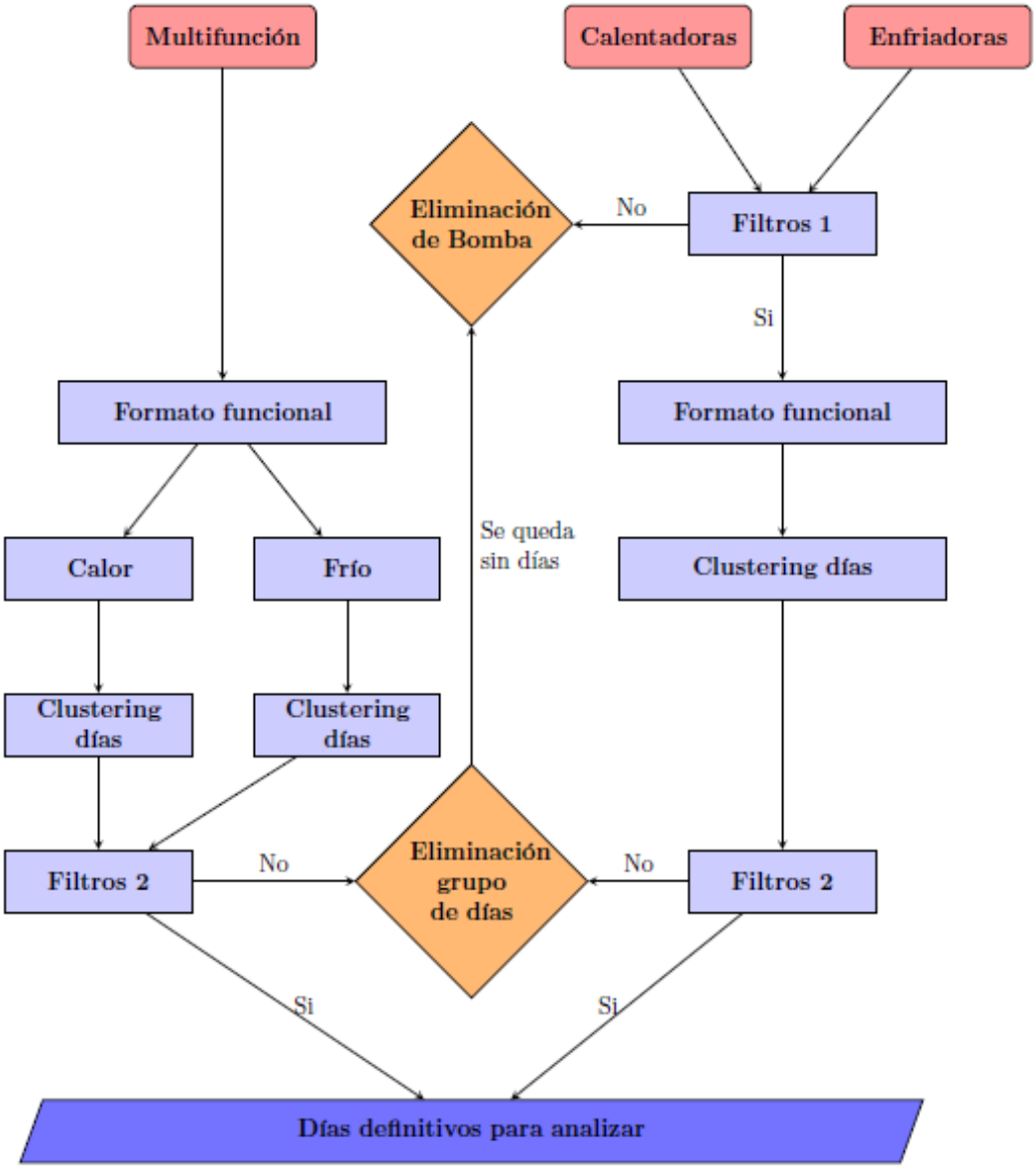


Figura 5.5: Diagrama de flujo de la segunda fase del algoritmo.

Bombas Multifunción

Esta segunda fase es diferente en el caso de bombas que están dentro del grupo multifunción. A continuación presentamos las peculiaridades del proceso dentro de este grupo:

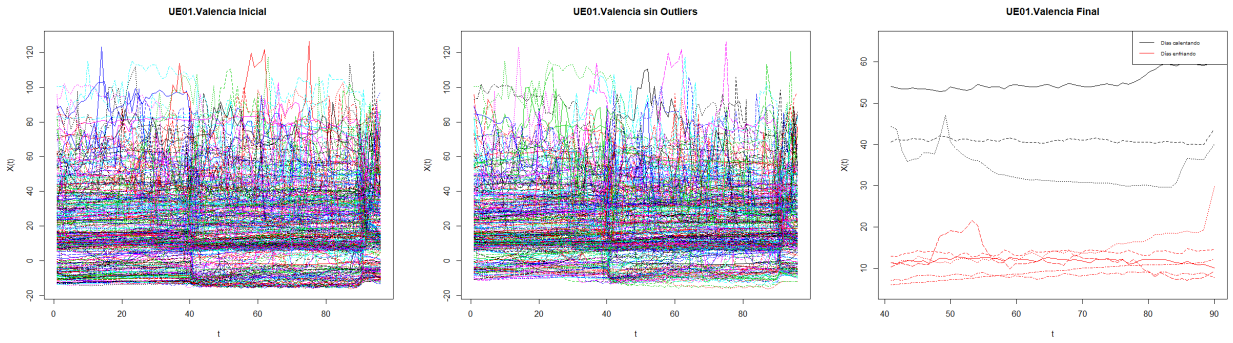
1. Separación de días donde la bomba está calentando de días donde está enfriando. Para esto utilizamos los *diascalor* y *diasfrio* obtenidos en la primera fase.

2. El límite para eliminar grupos de días con pocos datos pasa a ser 2; menos restrictivo.

3. Búsqueda automática de días con un comportamiento *correcto*, dentro de los días donde la bomba está calentando.

4. Búsqueda automática de días con un comportamiento *correcto* dentro de los días donde la bomba está enfriando.

El resto del proceso, se realiza de manera similar que con bombas enfriadoras o calentadoras. Por ejemplo, la selección de días, previa al análisis cluster, no se realiza ya que, para este grupo de bombas ya se realizó en la primera etapa. Además, como acabamos de presentar, la búsqueda automática del algoritmo, para buscar los días donde mejor se observa los patrones de comportamiento se lleva a cabo por dos ramas: una para los días donde está enfriando y otra para los días donde está calentando. Cada grupo se analiza por separado; como si fueran dos bombas distintas. Respecto a la parte del análisis de las distancias, es idéntica a la de las otras bombas.



(a) Gráfico de las temperaturas de impulsión diarias tras eliminar unicamente valores fuera de rango (354 días))

(b) Gráfico de las temperaturas de impulsión diarias tras eliminar outliers de una forma clásica (323 días).

(c) Gráfico de las temperaturas de impulsión diarias con las que trabajará el algoritmo (9 días).

Figura 5.6: Ejemplo de un análisis de los días significativos para una bomba multifunción.

5.2.3. Tercera fase

Nuestro algoritmo, en la última etapa, se centra en realizar un análisis cluster de las bombas dentro de cada grupo primario (calentadoras, enfriadoras o multifunción). La creación de grupos se llevará a cabo teniendo en cuenta las distancias $L2$ [29] entre las curvas medianas funcionales de las bombas. Además, el algoritmo en esta etapa, también se encargará de calcular y proporcionar información acerca de esta clasificación no supervisada, necesaria para la posterior clasificación supervisada que se quiera hacer.

Lo primero a destacar de esta fase del algoritmo (resumida en la figura 5.10) es que para llevar a cabo la agrupación de bombas de agua se tendrán en cuenta las temperaturas de impulsión y las temperaturas de retorno. Es decir, el cluster se efectuará en base a las distancias entre las curvas medianas funcionales de las temperaturas de impulsión, y también, de las de retorno. Para poder llegar a conseguir eso, y además hacerlo automático, el algoritmo sigue los siguientes pasos:

1. Cálculo de las curvas medianas funcionales de las temperaturas de impulsión sobre los días que pasaron los filtros para ser del grupo primario asignado (figuras 5.7 y 5.8).

2. Cálculo de las distancias entre las curvas medianas funcionales.

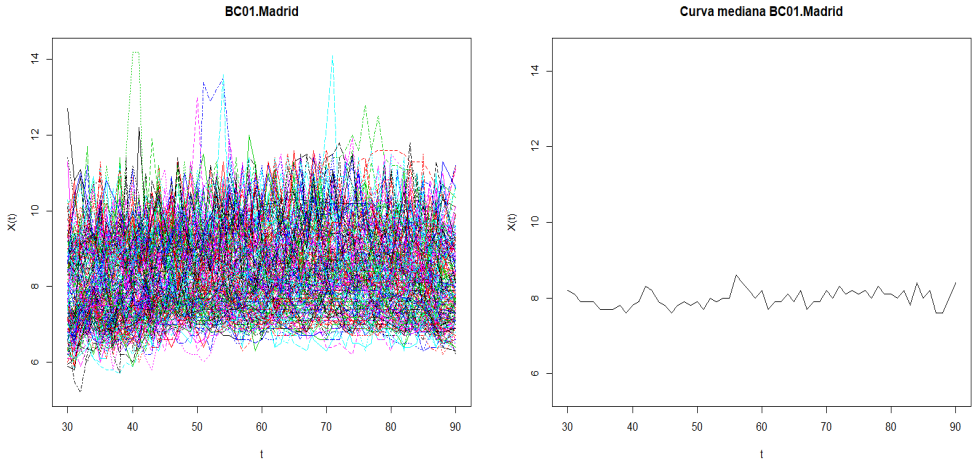
3. Estandarización de las distancias para poder operar con las distancias entre curvas de temperatura de impulsión y de retorno.

$$\text{Estandarización distancias (D): } \frac{D-\bar{D}}{sd(D)}$$

4. Repetir los pasos anteriores para los tres grupos primarios.

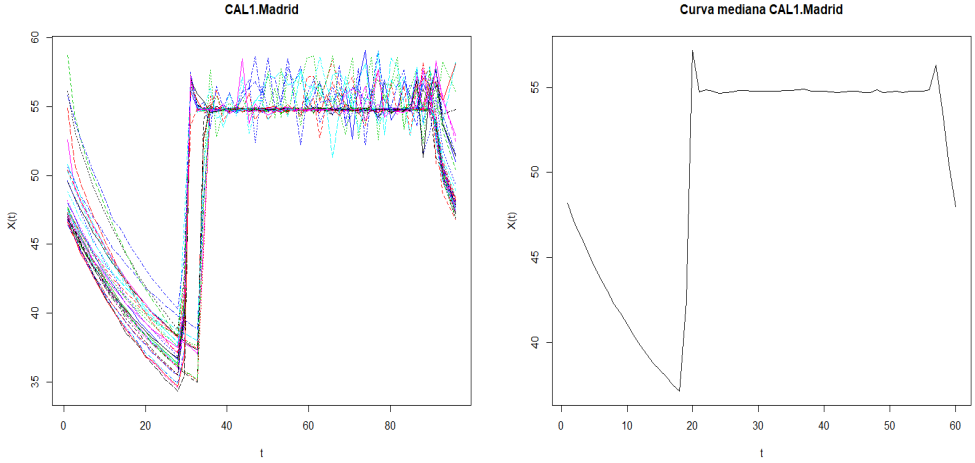
5. Repetir los pasos anteriores para las temperaturas de retorno.

[29] Distancias medidas trabajando en un espacio de Hilbert; explicado en el apartado 3.5.2.



(a) Temperaturas de impulsión diarias. (b) Curva mediana funcional.

Figura 5.7: Ejemplo del cálculo de la curva mediana funcional de una enfriadora.



(a) Temperaturas de impulsión diarias. (b) Curva mediana funcional.

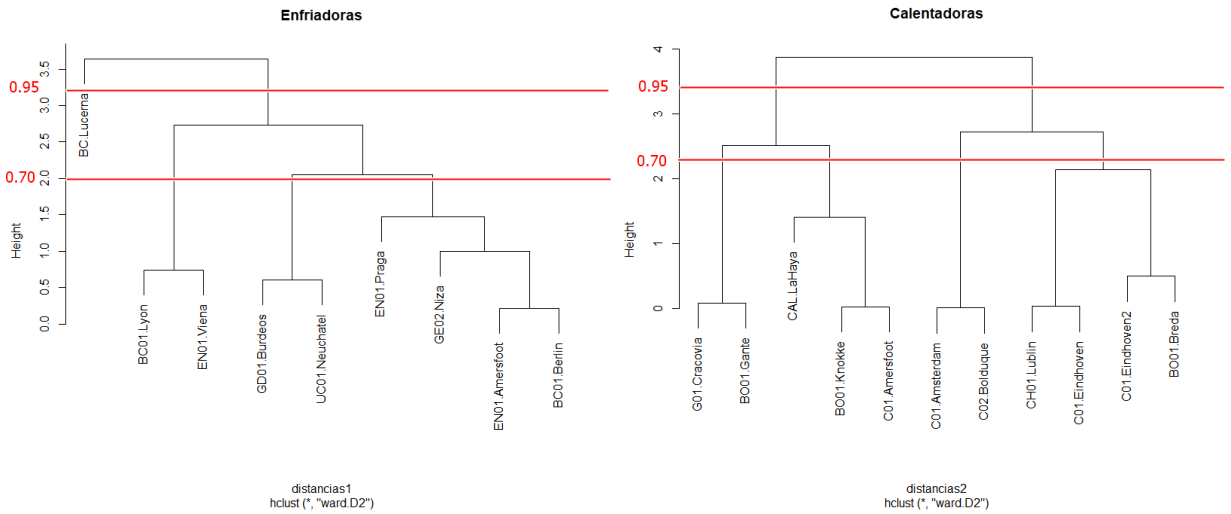
Figura 5.8: Ejemplo del cálculo de la curva mediana funcional de una calentadora.

Una vez realizadas las acciones anteriores, a el algoritmo solo le queda efectuar el análisis cluster que se llevará a cabo en base a las distancias estandarizadas entre las curvas medianas funcionales de ambas variables. Al estar estandarizadas podremos introducir, en el cluster, las distancias de la siguiente forma:

$$D = DI + DR$$

siendo DI las distancias relativas a las temperaturas de impulsión y DR a las temperaturas de retorno.

El siguiente problema al que se enfrenta el algoritmo es el de crear los grupos en base a la información proporcionada por el análisis cluster. En esta parte, hay que utilizar uno de los argumentos necesarios en el algoritmo. El *corte* definirá como queremos cortar el dendograma de bombas resultante del cluster. Por lo tanto, una vez realizado el cluster utilizando las distancias estandarizadas, el algoritmo corta el dendograma por el valor de distancias que dicte el argumento *corte*. De esta forma el algoritmo nos reporta una división en grupos por cada conglomerado primario y, además, dentro del grupo multifunción, se crean los grupos, por un lado, con las curvas medianas funcionales de las bombas cuando calientan, y por otro, cuando están enfriando. De forma simplificada, este proceso lo mostramos en la figura 5.9, donde se presentan dos dendogramas y los diferentes grupos resultantes del análisis, dependiendo del valor de *corte*.



(a) Dendograma de las enfriadoras de la muestra 2 cortadas por diferentes valores de *corte*.

(b) Dendograma de las calentadoras de la muestra 2 cortadas por diferentes valores de *corte*.

Figura 5.9: Ejemplo de una posible forma de creación de grupos.

En la figura 5.9 se plasma perfectamente lo que ocurre con el argumento *corte*. Como ya se expresó, puede variar entre 0 y 1, y es el valor del cuantil de distancias que calcularemos para obtener la distancia elegida. Dicho de otro modo, será el porcentaje de distancias que queremos que queden por debajo del valor elegido. Cuanto mayor sea este valor, menor será el número de grupos de bombas creados por el algoritmo. Como se puede observar en la figura 5.9 este argumento es importante, ya que al pasar el *corte* de un valor de 0.70 a un valor de 0.95, los grupos pasan de ser 4 a 2 en los dos conglomerados primarios mostrados. Define la exigencia para crear los grupos que le traspasamos al algoritmo.

Por otro lado, como ya mencionamos al principio de este capítulo, el algoritmo se encarga de proporcionar ciertas variables necesarias para realizar clasificaciones supervisadas en base a la información obtenida en este análisis. Estas variables son las siguientes:

- *groups*: data frame en el cual se encuentran, por columnas, los grupos a los que pertenecen las bombas de agua de la muestra. En la primera columna está la división en los grupos primarios y en la segunda columna la división en función del patrón de comportamiento de las bombas.

- *Bombas*: lista con las bombas que tienen información necesaria para realizar el estudio de su patrón de comportamiento. Por el camino alguna bomba puede haber sido eliminada.

- *tdatos*: matriz con la cantidad de días que cada bomba llega al clúster final. Días que pasaron todos los filtros para considerarse dignos de estudio.

- *rejilla*: lista con los horarios estandarizados de las bombas que han llegado al análisis final.

- *pesos*: matrices con los pesos necesarios para calcular el valor de las variables analizadas dentro de los horarios estandarizados.

- Las curvas medianas funcionales, tanto para las temperaturas de impulsión como para las de retorno, separadas por grupos primarios.

- Nombre de las bombas por grupo, tanto primarios como en función del patrón de funcionamiento.

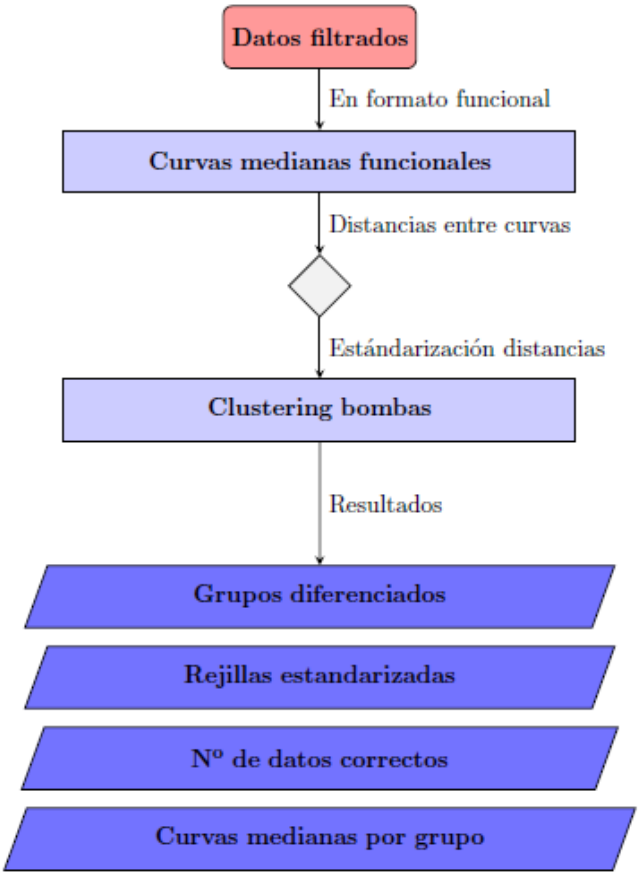


Figura 5.10: Diagrama de flujo de la tercera fase del algoritmo.

5.3. *Clasificador.S*: clasificación supervisada

En este apartado vamos a explicar el segundo algoritmo creado. El *Clasificador.S* se encarga de asignar, a uno de los grupos creados a través de técnicas cluster con el *Clasificador*, una o varias bombas de agua nuevas. Es decir, mediante este algoritmo conseguimos relacionar una nueva muestra de bombas con los grupos creados en el pasado, y ver en cuál encaja mejor. A través de unos sencillos pasos, muy similares a los llevado a cabo en la clasificación no supervisada, se consigue averiguar qué grupo encaja mejor con el patrón de comportamiento de estas nuevas bombas de agua. Los argumentos que son necesarios introducir en el algoritmo son los siguientes:

- *new*: lista con las nuevas bombas en formato matriz con las variables en columnas.
- *newnames*: lista con los nombres de las nuevas bombas.
- *nb*: número de repeticiones bootstrap en la búsqueda de outliers.
- *regg*: una *rejilla* de valores con la longitud deseada para calcular la consiguiente *rejilla* estandarizada con los horarios de cada bomba.
- *groups*: data frame en el cual se encuentran, por columnas, los grupos a los que pertenecen las bombas de agua de la muestra original.
- *memb[...]*: grupos de bombas diferenciados para cada una de las agrupaciones primarias (enfriadoras, calentadoras y multifunción).
- *medianas.[...]*: curvas medianas funcionales de las temperaturas de impulsión y retorno de los diferentes grupos primarios.
- *Bombas*: lista de las bombas que han sido agrupadas a través del algoritmo *Clasificador*.
- *names*: nombres de las bombas de la muestra original.
- *rejilla*: lista con las *rejillas* de puntos estandarizadas para cada bomba de la muestra original.
- *pesos11*: matriz con los pesos inferiores necesarios para ajustar los valores de las variables analizadas a la *rejilla* estandarizada. Los pesos de cada bomba se distribuyen en columnas.
- *pesos22*: matriz con los pesos superiores necesarios para ajustar los valores de las variables analizadas a la *rejilla* estandarizada. Los pesos de cada bomba se distribuyen en columnas.

Aunque parezca una cantidad relativamente alta de argumentos, la mayoría de ellos son proporcionados ya por el algoritmo *Clasificador*. La nueva muestra deberá ser de la misma estructura que la muestra original. Las variables temperatura de impulsión, temperatura de retorno, código de encargada/o e indicencias dispuestas en columnas. Debido a la posibilidad de la existencia de valores faltantes o valores fuera de rango el algoritmo hace una *limpieza* de datos. Además, como no se tiene ninguna información sobre estas bombas, también realiza una clasificación primaria para saber si son enfriadoras, calentadoras o multifunción. Esto, con alguna variación para adecuar el código, será igual que en la fase 1 de la clasificación no supervisada. A continuación, tal y como se desarrolla en la fase 2 de la clasificación no supervisada, hace una búsqueda, con técnicas cluster, de los días donde de verdad se puede extraer información significativa.

Una vez que tenemos los datos adecuados para trabajar, el algoritmo calcula las curvas medianas funcionales de las bombas de agua de la nueva muestra. Con estas curvas, en comparación con las curvas medianas de la muestra original, el algoritmo averiguará a qué grupo de bombas, de los ya creados, asignará cada bomba nueva. Para poder conseguir esto el algoritmo lleva a cabo los siguientes pasos:

1. Obtiene la estructura de los grupos ya creados, en base a que la nueva bomba sea una enfriadora, una calentadora o una bomba multifunción.
2. Con las curvas medianas originales y la información sobre a qué grupo pertenece cada bomba (es decir, cada curva mediana), realiza una clasificación supervisada en base al clasificador DD (explicado en el apartado 3.5.1). Hemos escogido este clasificador porque nos permite introducir en el análisis curvas de temperatura de impulsión y de retorno; así la clasificación se lleva cabo con los mismos datos que el cluster. Además, la profundidad elegida es la de proyecciones aleatorias y el orden DD^2, ya que, como

mostramos en el cuadro 5.1, tienen un menor porcentaje, en media, de malas clasificaciones que las demás opciones.

	Tipo de profundidad	Calentadoras	Enfriadoras
Clasificador DD^2	*Modal*	0.410	0.450
Clasificador DD^2	*Fraiman-Muniz*	0.046	0.020
Clasificador DD^2	*Proyecciones aleatorias*	**0.002**	**0.000**
Clasificador DD^3	*Modal*	0.430	0.404
Clasificador DD^3	*Fraiman-Muniz*	0.052	0.032
Clasificador DD^3	*Proyecciones aleatorias*	0.004	0.000

Cuadro 5.1: Comparación entre distintas profundidas para realizar una clasificación supervisada en base los DD-plots. Media de la proporción de mal clasificados para cada profundidad a través de remuestras bootstrap sobre la muestra de bombas original (100 repeticiones de Monte Carlo).

3. Con la información obtenida en el anterior paso, el algoritmo, a través de probabilidades y en función de las curvas medianas funcionales de las temperaturas, calcula a que grupo sería más probable que perteneciera la nueva bomba.

Los anteriores pasos son el procedimiento normal a seguir por el algoritmo; pero si el número de bombas de la muestra original, por ejemplo calentadoras, son pocas esto cambia. Al haber muy pocas bombas clasificadas en grupos diferenciados es muy díficil estimar el grupo al que pertenecería una nueva bomba. El resultado no sería fiable. Por este motivo, si las bombas dentro de un grupo primario son menos de 8, el procedimiento será el siguiente:

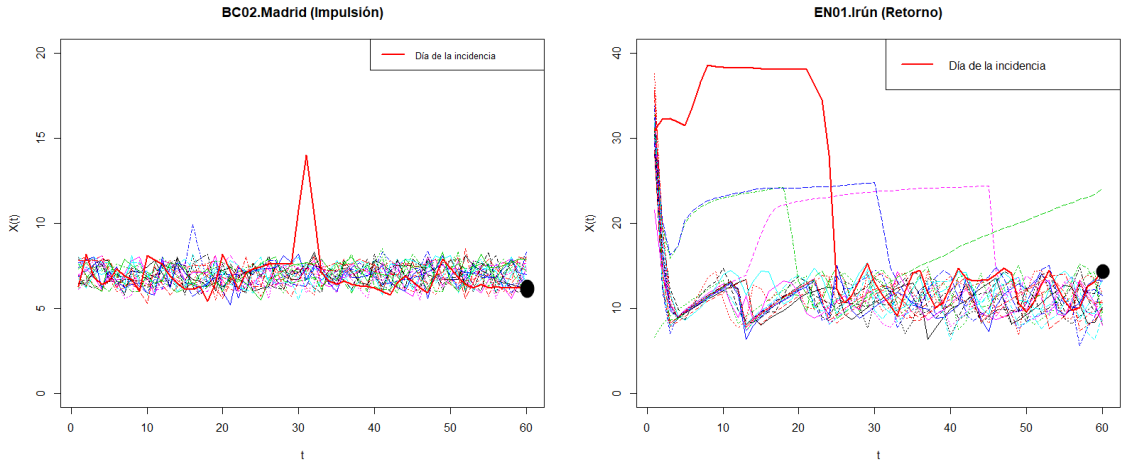
1. Cálculo de la distancia $L2$ de la curva mediana funcional de la nueva bomba respecto a las curvas medianas de la muestra original, tanto para la temperatura de impulsión como para la de retorno.
2. Estandarización de las distancias para poder trabajar con las distancias entre las curvas de temperatura de impulsión y entre las curvas de retorno.
3. Cálculo de la distancia, en media, de la nueva curva mediana funcional a cada uno de los grupos de bombas ya creados.
4. Selección del grupo, que tiene una menor distancia, en media, a la nueva curva mediana.

De esta forma sencilla y rápida, el algoritmo nos reporta el grupo primario al que pertenecen las nuevas bombas, y también el grupo, que según sus patrones de comportamiento, se asemejan más. Sabremos si estas nuevas bombas son enfriadoras, calentadoras o bombas multifunción, y también con qué bombas tendremos que relacionarlas en el futuro. Por otro lado, como en el *Clasificador*, este algoritmo nos muestra cuántos datos fueron dignos de analizar en cada bomba; es decir, en cuántos días no hubo errores de medición durante el período de tiempo analizado. Por último, el *Clasificador.S* incorpora los datos de estas nuevas bombas a las listas con los datos de las bombas originales (*Bombas*, *names*, *pesos*...) para tener toda la información junta, organizada y lista, y poder empezar a aplicar el algoritmo de detección de incidencias en estas nuevas bombas cuanto antes.

5.4. *Detección*: detección de incidencias

Un objetivo importante de este TFM era poder anticipar malos comportamientos en las instalaciones de climatización. Este último algoritmo estudia la evolución de las curvas funcionales de la temperatura de impulsión y de retorno, en días donde todo marcha correctamente, para poder detectar las futuras incidencias antes de lo que lo están haciendo actualmente. Este algoritmo, a pesar de la gran utilidad que puede reportar, no necesita gran cantidad de argumentos para su funcionamiento. Los argumentos son: una lista con las bombas a las que se le quiere aplicar el estudio de detección de incidencias; con datos del pasado para analizar (*Bombas*), una lista con los nombres de todas las bombas (*names*), una lista con las *rejillas* estandarizadas para todas las bombas (*rejilla*), el número de momentos quinceminutales hacia atrás deseados para analizar las posibles futuras incidencias (*long*), el número de repeticiones bootstrap para la búsqueda de outliers (*nb*) y la división en grupos primarios de todas las bombas a analizar (*grupos*).

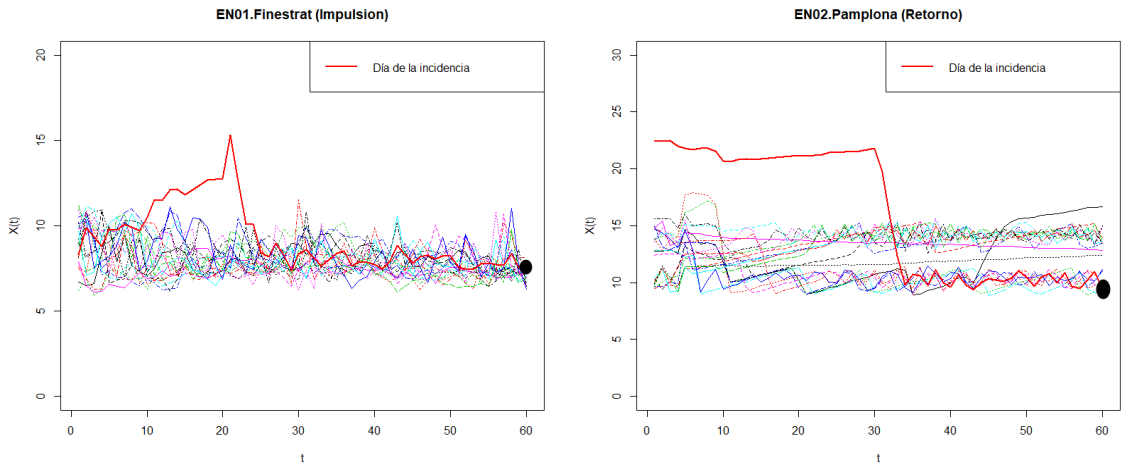
En las gráficos siguientes (figura 5.11 y 5.12) mostramos algunos ejemplos de curvas que tienen anotadas incidencias, y el porqué no debemos tenerlas en cuenta para comprobar el buen funcionamiento actual de las bombas además de detectarlas lo antes posible.



(a) BC02.Madrid 21 dias. Día de la incidencia: *Martes*.

(b) Irun 20 dias. Día de la incidencia: *Viernes*.

Figura 5.11: Ejemplos de dos bombas de agua, diferenciando curvas con incidencias; desde que se produjo la anotación de la incidencia (punto negro) hasta 24 horas atrás. Tope 25 curvas.



(a) Finestrat 19 dias. Día de la incidencia: *Sábado*.

(b) Pamplona 20 dias. Día de la incidencia: *Jueves*.

Figura 5.12: Ejemplos de dos bombas de agua, diferenciando curvas con incidencias; desde que se produjo la anotación de la incidencia (punto negro) hasta 24 horas atrás. Tope 25 curvas.

El primer paso de este algoritmo es transformar el formato de fecha que viene por defecto en los archivos descargados desde la plataforma *OTEA*. Los pasa a formato día, es decir, lo único que nos indicará esta variable es el día que estamos analizando. El algoritmo realiza esta tarea para distinguir los lunes y domingos del resto de días de la semana. Posiblemente los domingos muchas tiendas cierran, y por lo tanto, su comportamiento, junto al de los lunes (día posterior a un día de cierre), sea diferente al resto de días. Por este motivo, si en la actualidad estamos en un lunes o en un domingo, el proceso para detectar incidencias será algo diferente.

A continuación, el algoritmo estudia la existencia de valores faltantes en el inicio y final del período de tiempo analizado por lo problemas que pueden generar al aproximar los demás valores faltantes. Una vez arreglado esto, se procede a separar el día actual, seleccionando el intervalo de tiempo más reciente con su longitud determinada por el argumento *long*. Los pasos a seguir por el algoritmo a partir de aquí son los siguientes:

1. Comprobación de valores faltantes en las curvas actuales (temperatura de retorno y de impulsión); si la cantidad de NAs es mayor al 30 % del intervalo de tiempo analizado saltará un aviso informando de la existencia de demasiados valores faltantes.

2. Aproximación de los posibles valores faltantes en la curvas actuales.

3. Transformación de esas curvas a formato funcional.

4. Comprobación de la existencia de valores fuera de rango en las curvas actuales. De ser así, salta un aviso informando que alguna curva actual está midiendo fuera de rango.

5. Análisis del día o días en que cae el intervalo de tiempo analizado. Según el día en que estemos en la actualidad el algoritmo procede de manera diferente:

   - Domingos o lunes: Búsqueda de domingos o lunes pasados[30] en el mismo intervalo de tiempo estudiado en la curva actual. Días donde no se produjeron incidencias [31] ni mediciones fuera de rango. En este caso, únicamente buscamos domingos o lunes porque, en el resto de días de la semana, el comportamiento de la bomba, en media, es diferente.
   - Días laborales: Búsqueda de cualquier día *laboral*[32] pasado analizando el mismo intervalo de tiempo que las curvas actuales. Como en el anterior caso, los días que aceptará el algoritmo serán días en que no se produjeron incidencias ni mediciones *erróneas*. En cambio, en este caso, podemos trabajar con varios días de la semana ya que su comportamiento es similar de un día para otro. Si la tienda está abierta, las bombas tendrán un patrón de funcionamiento cercano.

   La cantidad de curvas que el algoritmo seleccione para la comparación con la curvas actuales se puede variar. Actualmente le pasamos al algoritmo un tope de 50 curvas; es decir, buscará 50 curvas en el pasado que cumplan las condiciones mencionadas, pero sin irse demasiado lejos en el tiempo (las condiciones climatológicas pueden variar fuertemente) [33]. De este modo, puede presentarse el caso en el que tenemos información de menos de 50 curvas.

6. Si el número de curvas pasadas seleccionadas para comparar con las actuales es muy pequeño saltará un aviso; las mediciones tomadas en el pasado de esta bomba no son buenas. En cambio, si el número de curvas es aceptable [34], las pasamos a formato funcional.

7. Búsqueda de outliers en las curvas pasadas, para prevenir la posible existencia de curvas con un comportamiento alejado al estándar de la bomba y que distorsionara el posterior análisis.

8. Cálculo de las distancias entre todas las curvas pasadas; tanto para la temperatura de impulsión como de retorno, (D).

9. Cálculo de las distancias de las curvas actuales con todas las curvas del pasado, ($D2$).

[30] Si el intervalo actual cae en domingo bucaremos domingos pasados, y si el intervalo cae en lunes buscaremos lunes.
[31] Esto lo comprobamos a través de la variable incidencias.
[32] De martes a sábado.
[33] Alrededor de un año como máximo para retroceder en el tiempo.
[34] 10 o más curvas.

10. Cálculo de la derivada de las curvas actuales. En base a ellas el algoritmo estudiará la existencia de mediciones constantes que sería interesante detectar. En este caso, si la longitud de las mediciones constantes (una recta) de las temperaturas supera el 75 % de la longitud del intervalo a analizar, saltará un aviso advirtiendo de que las mediciones actuales muestran un posible riesgo de incidencia.

11. Estudio, a través del test *Kolmogorov - Smirnov* [35], de la igualdad de distribuciones de las distancias D y $D2$. Más concretamente, el algoritmo estudiará si la distribución de distancias de las curvas actuales contra las curvas pasadas ($D2$) está por encima de la distribución de las distancias de las curvas pasadas entre sí (D). Esto significaría que la curva actual está alejada de las curvas del pasado con buen comportamiento. El test *Kolmogorov - Smirnov* estudia las funciones de distribución de cada muestra, y en este caso, parte de la premisa de que la distribución de $D2$ es la misma que la de D; por este motivo, el aviso saltará cuando rechacemos la hipótesis nula acabada de mencionar [36]. Programando el algoritmo con estas reglas conseguimos que tenga que observar evidencias de un comportamiento extraño para rechazar la hipótesis nula y concluir que pueden existir problemas.

12. Análisis de los valores de las temperaturas en las curvas actuales. Si, por ejemplo, estamos ante una calentadora y las curvas actuales están por debajo de unos limites más del 25 % de la duración del intervalo, saltará un aviso señalando que la bomba esta impulsando o retornando fuera de rango. Los limites, como en el algoritmo *Clasificador*, son:

    - *Calentadoras*: la temperatura de impulsión debe ser de 22 grados o más, y la temperatura de retorno de 18 grados o más.
    - *Enfriadoras*: la temperatura de impulsión debe ser de 18 grados o menos, y la temperatura de retorno de 22 grados o menos.
    - *Multifunción*: no se exigen limites de temperatura porque el día actual puede ser de calentar o enfriar. Pero lo que si estudia el algoritmo, son los valores de las temperaturas conjuntamente con la variable diferencia entre ellas. De esta forma el algoritmo puede detectar si la bomba está calentando a temperaturas muy bajas o viceversa. Dicho de otro modo, una bomba de agua no podría estar enfriando si, en media, impulsa por encima de los 22 grados. El aviso nos mostrará exactamente lo que esta ocurriendo.

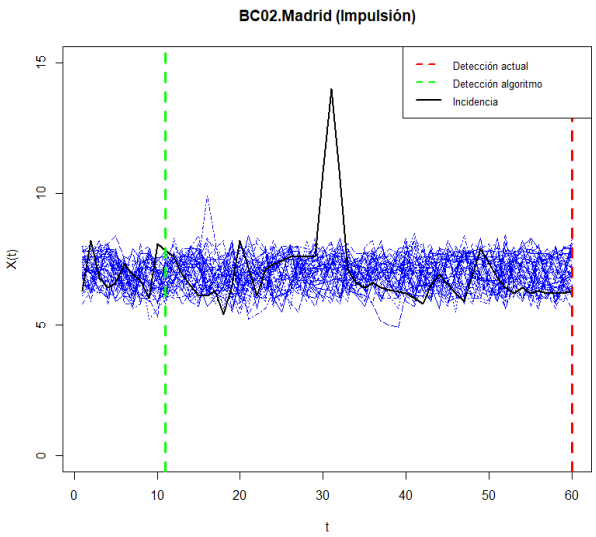
13. Comprobación del correcto funcionamiento de la bomba de agua en las curvas actuales. En el caso de las bombas calentadoras y enfriadoras, el algoritmo es capaz de detectar si la relación entre la temperatura de impulsión y de retorno es la adecuada. Es decir, el algoritmo mostrará un aviso si, por ejemplo, una calentadora tiene las temperaturas de impulsión por debajo de las de retorno más de un 50 % de la duración del intervalo estudiado (lo mismo para el caso de enfriadoras). En el supuesto de estar analizando una bomba multifunción, el proceso es el mismo diferenciando si está calentando o enfriando.

Este algoritmo, tal y como acabamos de explicar, permitirá a la empresa detectar riesgos en el funcionamiento de las bombas antes que si lo hacen mediante comprobaciones físicas en las tiendas. En el caso de EcoMT, mostramos la mejora que les reportaría el algoritmo en las figuras 5.13 y 5.14 [37].

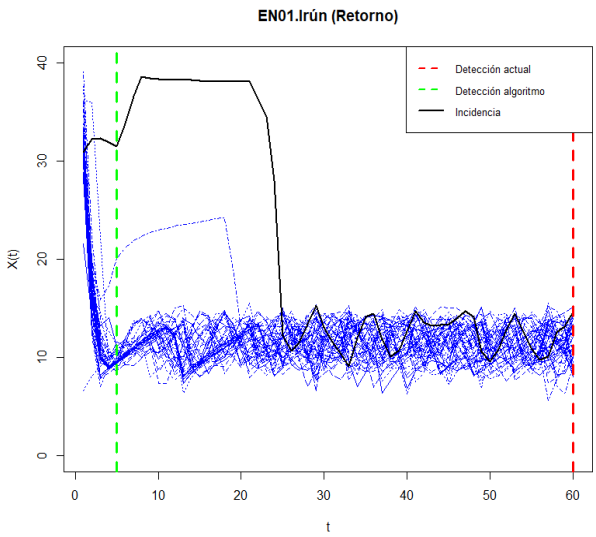
[35] Con las funciones de distribución de las dos muestras se calculará el estadístico: $D_{D2,D} = \sup_x(F_{D2}(x) - F_D(x))$ y se rechaza la hipótesis nula si el valor del estadístico es muy grande.

[36] Este aviso informará de que alguna de las curvas actuales está fuera de rango.

[37] Diferentes escalas para resaltar el comportamiento de la curva con incidencia.

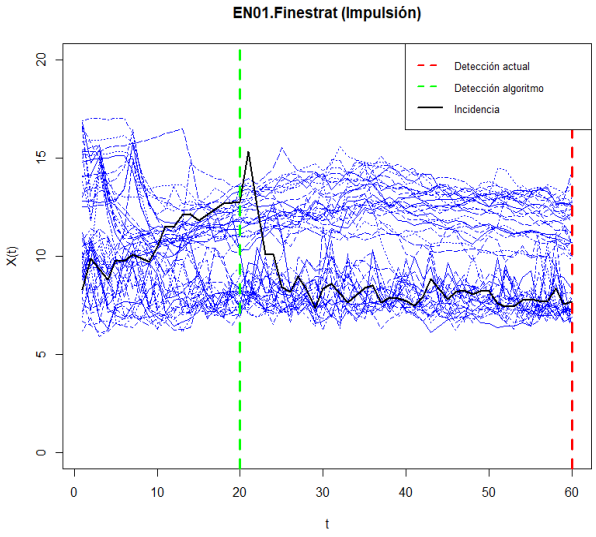


(a) Detección de incidencia (*Martes*).

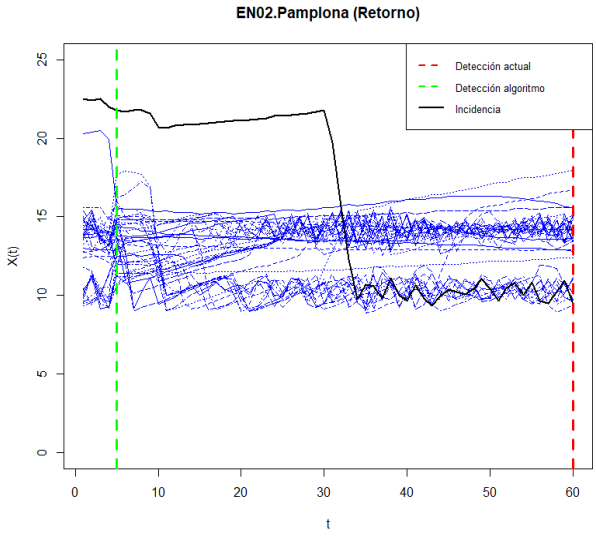


(b) Detección de incidencia (*Viernes*).

Figura 5.13: Comparación del momento de detección de una incidencia: de la forma actual y con el algoritmo (tope 50 curvas).



(a) Detección de incidencia (*Sábado*).



(b) Detección de incidencia (*Jueves*).

Figura 5.14: Comparación del momento de detección de una incidencia: de la forma actual y con el algoritmo (tope 50 curvas).

En definitiva, este algoritmo nos permite juzgar si el patrón de funcionamiento *actual* de la bomba entra en su estándar (medido a través del pasado) o, por el contrario, por diferentes razones que mostramos en el cuadro 5.2, el comportamiento actual se aleja de ese estándar y, por lo tanto, existe riesgo de incidencias.

Problemas	**Avisos; riesgo de incidencia**
Valores de las temperaturas actuales extremos (< 0 ó > 65).	*Curva fuera de rango.*
Curva actual alejada del comportamiento estándar del pasado.	*Curva fuera de rango.*
Temperaturas actuales altas siendo enfriadora.	*Curva fuera de rango.*
Temperaturas actuales bajas siendo calentadora.	*Curva fuera de rango.*
Enfriadora calentando.	*T. impulsión por encima de la T. retorno.*
Calentadora enfriando.	*T. impulsión por debajo de la T. retorno.*
Demasiados valores faltantes en la curva actual	*Valores faltantes en la curva actual.*
Mediciones de las temperaturas actuales constantes.	*Temperaturas constantes.*
Datos del pasado no válidos por no estar bien medidos.	*No existen curvas estándar en el pasado.*

Cuadro 5.2: Problemas detectados por el algoritmo *Detección* que muestran posibles riesgos de incidencias en el futuro.

Capítulo 6

Análisis de datos y resultados

En este capítulo presentamos el análisis de datos reales llevado a cabo para verificar el rendimiento de los tres algoritmos creados. El análisis consiste en aplicar los algoritmos, uno a uno, a las dos muestras mencionadas en el capítulo 4. Para ello, se procedió de la siguiente forma:

1. Apartamos dos bombas de cada muestra para poder realizar una clasificación supervisada.
2. Aplicamos el algoritmo *Clasificador* a la muestra completa [38]. De esta forma conseguimos datos como los grupos de bombas, o las curvas medianas funcionales, necesarios para una posterior clasificación supervisada.
3. En base a los grupos creados, aplicamos el algoritmo *Clasificación.S* a las bombas apartadas en el primer paso. Obtenemos los grupos que más se asemejan al comportamiento de las nuevas bombas.
4. Partiendo de la suposición de que el último día del 2018 [39] es el día *actual* en el que queremos estudiar la posibilidad de incidencias futuras, aplicamos el algoritmo *Detección* a todas las bombas (tanto las originales como las nuevas). Con esto detectamos si las curvas actuales de temperaturas, de cada bomba, muestran valores o comportamientos alejados de la media; lo cual podría significar la existencia de futuros problemas en la instalación.

Para llevar a cabo el proceso de análisis de datos acabado de presentar se utilizaron los ficheros: *Clasificador*, *Clasificador.S*, *Detección*, *datos*, *BOMBAS* y *BOMBAS2*. En los cuatro primeros ficheros están los códigos, implementados en R, relativos a cada uno de los tres algoritmos creados y al script con las funciones creadas y la llamada de los datos. En los otros dos estarían los datos de las dos muestras de bombas de agua utilizadas en este estudio. A continuación mostramos los resultados a los que se llegaron con la implementación de los algoritmos; de forma esquemática, presentamos la información que pueden proporcionar.

6.1. Clasificación no supervisada.

En esta sección vamos a mostrar los grupos, en función del patrón de funcionamiento, resultantes de aplicar el algoritmo *Clasificador* a las dos muestras de bombas de agua.

6.1.1. Muestra 1. Península Ibérica

Los valores de los argumentos con los que hemos trabajado en este análisis son los siguientes: *corte*= 0.75, *nb*=250 y *regg*= secuencia desde 1 a 60, con una longitud de 60. De esta forma, con la muestra 1 obtenemos los grupos expuestos en las figuras 6.1 y 6.2. Además, para poder apreciar las diferencias entre ellos, tanto en niveles de temperatura como en forma de las curvas, también mostramos los grupos individualmente; es decir, las curvas medianas, pertenecientes a cada grupo, en gráficos diferentes (figuras 6.3, 6.4 y 6.5). Por último, destacar que el algoritmo ha conseguido llegar a estos resultados en, aproximadamente, 11 minutos.

[38] Sin las bombas apartadas en el anterior paso.
[39] Hasta donde llega la muestra de datos.

- ENFRIADORAS

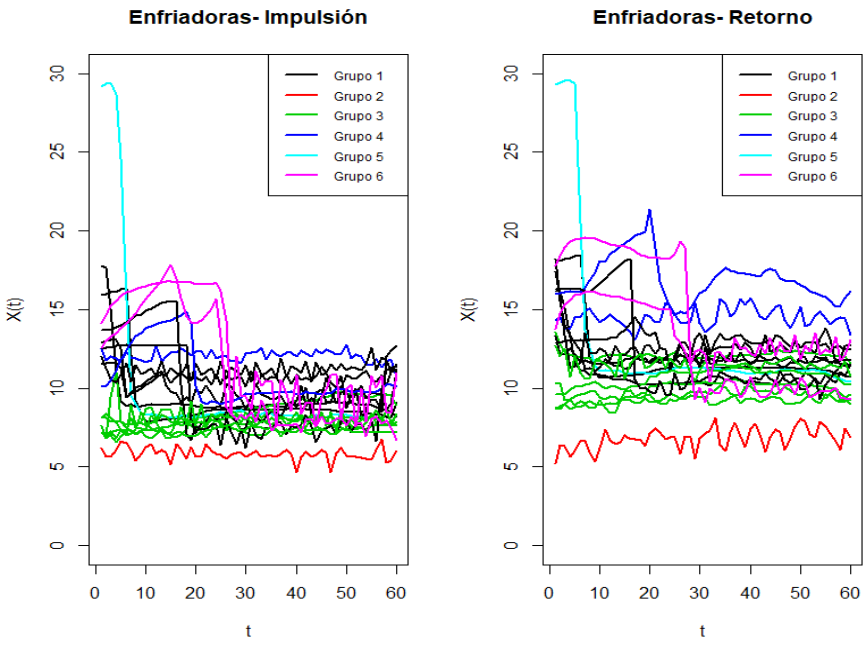


Figura 6.1: Curvas medianas de las enfriadoras separadas por grupos según su patrón de comportamiento.

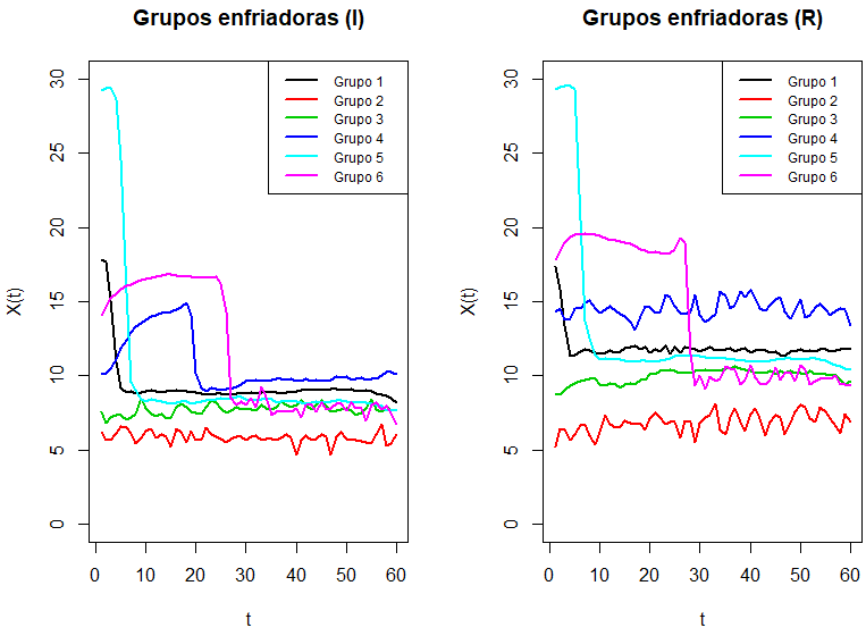
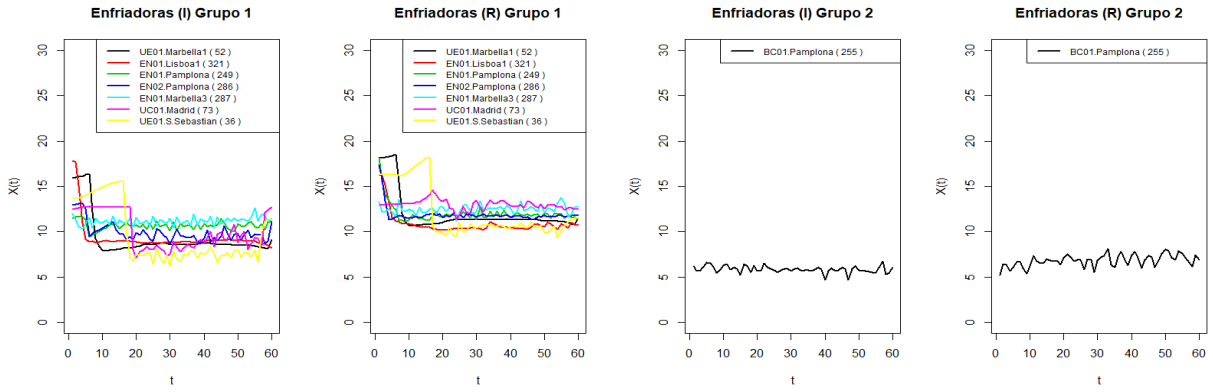


Figura 6.2: Curvas medianas de cada grupo de enfriadoras creado.

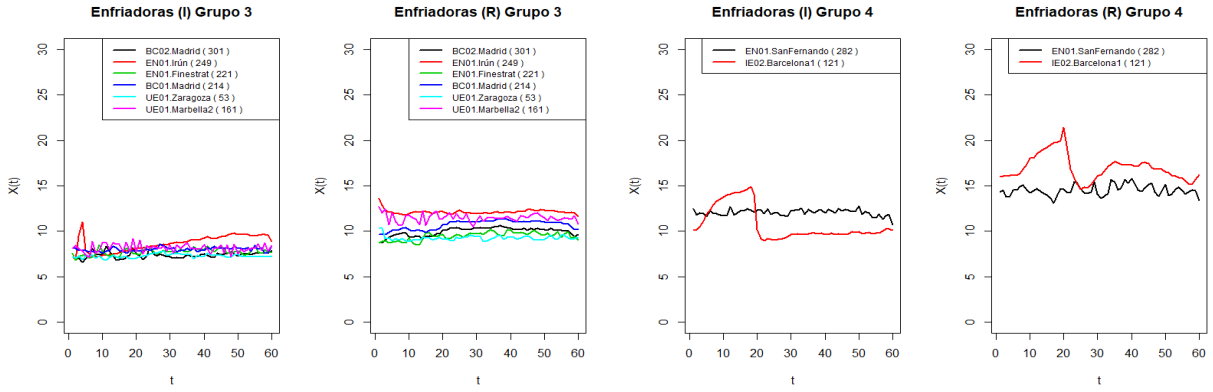
En este caso, el resultado que nos reporta el algoritmo es que la solución óptima es agrupar las 19 bombas de agua enfriadoras en 6 grupos diferentes. A continuación presentamos las bombas que forman cada uno de estos grupos.



(a) Grupo 1. Formado por UE01.Marbella1, EN01.Lisboa1, EN01.Pamplona, EN02.Pamplona, EN01.Marbella3, UC01.Madrid y UE01.S.Sebastián.

(b) Grupo 2. Formado por BC01.Pamplona

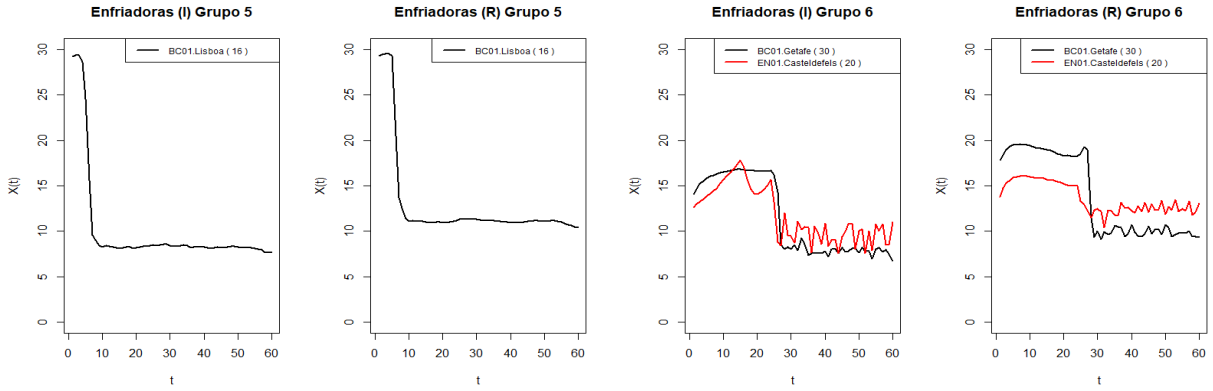
Figura 6.3: Grupos creados por el algoritmo. Curvas medianas de temperaturas de impulsión (I) y retorno (R).



(a) Grupo 3. Formado por BC02.Madrid, EN01.Irún, EN01.Finestrat, BC01.Madrid, UE01.Zaragoza y UE01.Marbella2.

(b) Grupo 4. Formado por EN01.SanFernando y IE02.Barcelona1.

Figura 6.4: Grupos creados por el algoritmo. Curvas medianas de temperaturas de impulsión (I) y retorno (R).



(a) Grupo 5. Formado por BC01.Lisboa.

(b) Grupo 6. Formado por BC01.Getafe y EN01.Castedefels.

Figura 6.5: Grupos creados por el algoritmo. Curvas medianas de temperaturas de impulsión (I) y retorno (R).

- CALENTADORAS

  Debido a que en esta muestra el número de calentadoras es pequeño (2 calentadoras), mostramos directamente las curvas medianas de cada bomba de agua para poder observar sus diferencias y/o similitudes. Las bombas de agua calentadoras, en este caso, con C01.S.Sebastián y CAL1.Madrid.

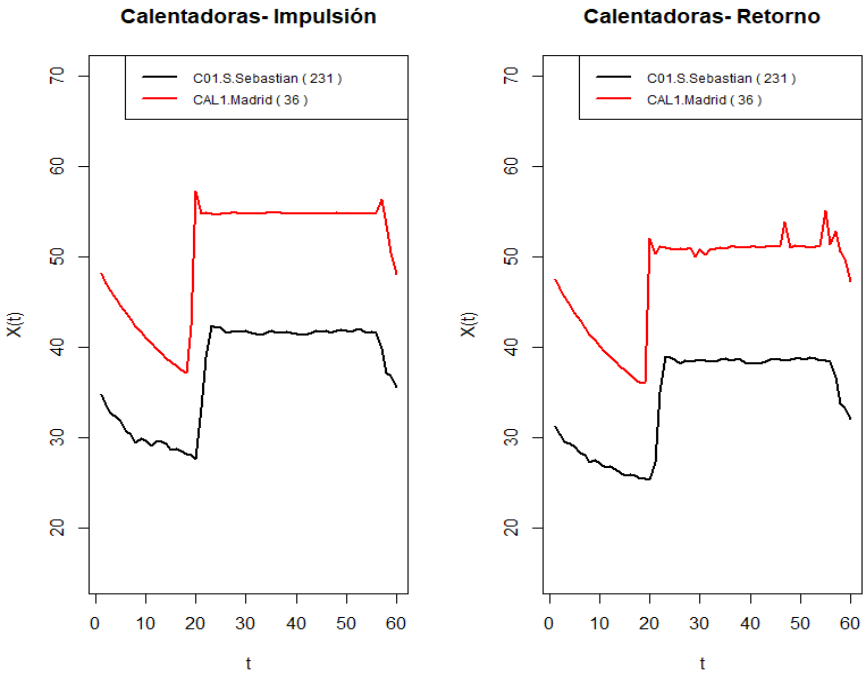


Figura 6.6: Curvas medianas de las calentadoras separadas por grupos según su patrón de comportamiento.

- BOMBAS MULTIFUNCIÓN

  Al igual que las calentadoras, contamos con pocas bombas multifunción (2 bombas multifunción) en la muestra 1. Por este motivo, también vamos a mostrar las curvas medianas de cada bomba de agua (UE01.Valencia y BC01.CiudadReal). Lo peculiar de este grupo de bombas es que las curvas medianas se separan en días calentando y días enfriando.

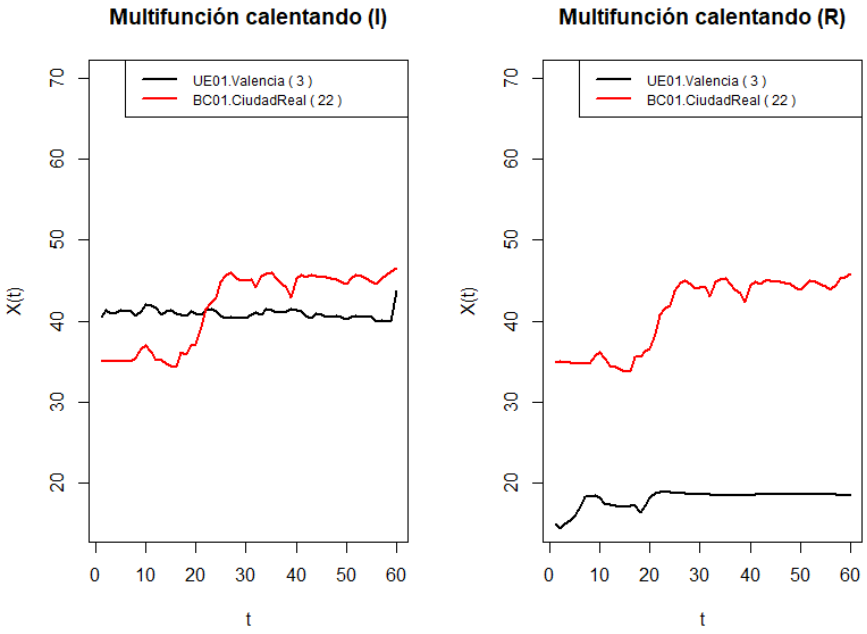


Figura 6.7: Curvas medianas de las bombas multifunción calentando, separadas por grupos según su patrón de comportamiento.

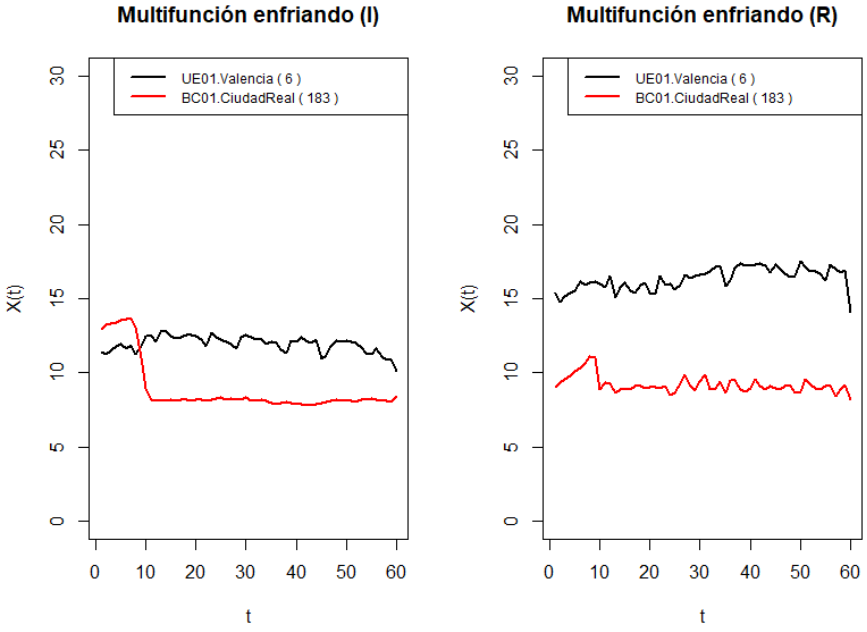


Figura 6.8: Curvas medianas de las bombas multifunción enfriando, separadas por grupos según su patrón de comportamiento.

6.1.2. Muestra 2. Europa central

Los valores de los argumentos con los que hemos trabajado en este caso son los siguientes: *corte*= 0.70, *nb*=250 y *regg*= secuencia desde 1 a 60, con una longitud de 60. Los grupos obtenidos con la muestra 2, basados en los patrones de funcionamiento de las bombas, se presentan en las figuras 6.9 y 6.10.. Además, para llegar a esta clasificación el algoritmo ha tardado, aproximadamente, 11 minutos.

- ENFRIADORAS

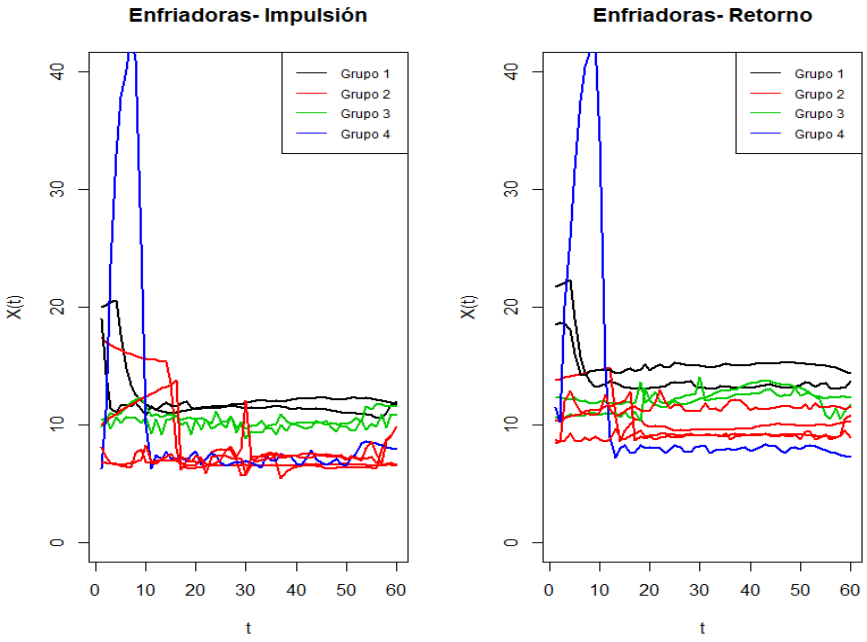


Figura 6.9: Curvas medianas de las enfriadoras separadas por grupos según su patrón de comportamiento.

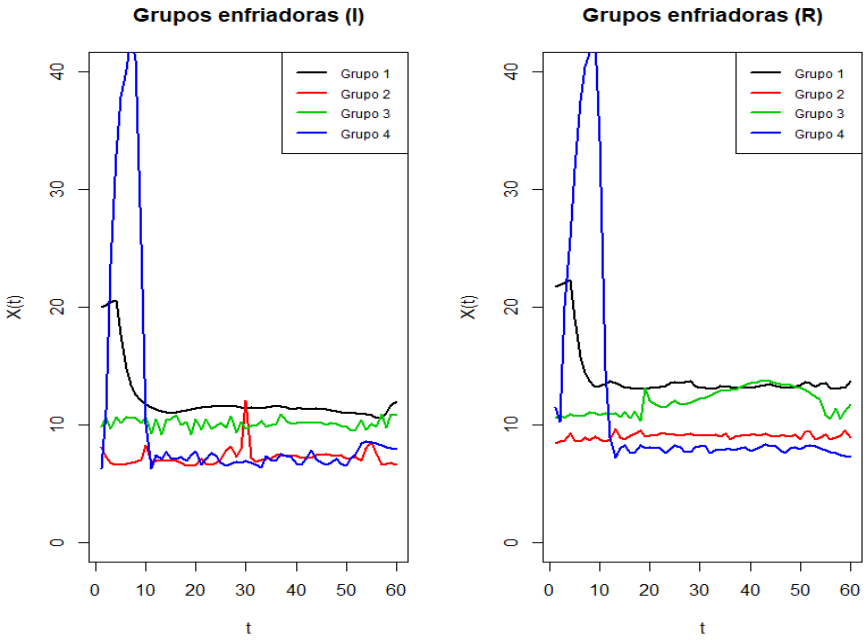
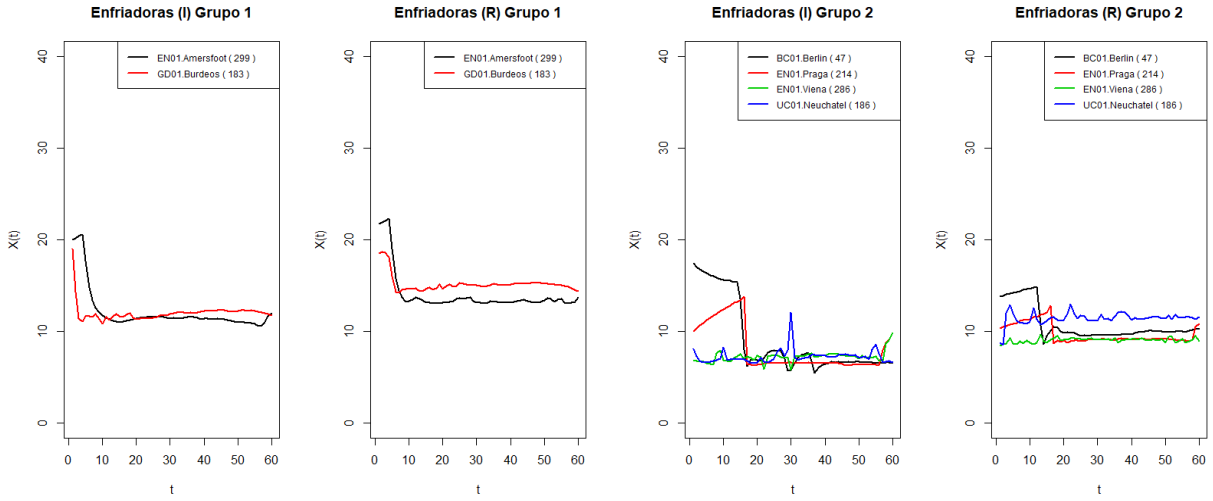


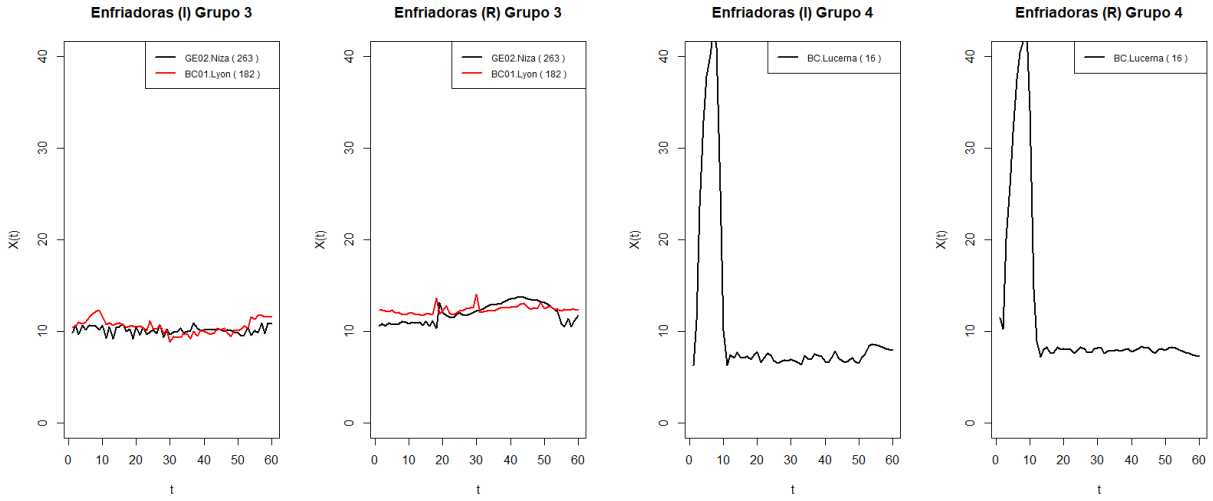
Figura 6.10: Curvas medianas de cada grupo de enfriadoras creado.

La solución óptima a la que llega el algoritmo es agrupar las 9 bombas de agua enfriadoras en 4 grupos. A continuación, en las figuras 6.11 y 6.12, presentamos las bombas que forman cada uno de estos grupos y el número de datos *correctos* con los que se efectuó el análisis.



(a) Grupo 1. Formado por EN01.Amersfoot y GD01.Burdeos. (b) Grupo 2. Formado por BC01.Berlín, EN01.Praga, EN01.Viena y UC01.Neuchatel.

Figura 6.11: Grupos creados por el algoritmo. Curvas medianas de temperaturas de impulsión (I) y retorno (R) acompañadas del número de datos estudiados (entre paréntesis).



(a) Grupo 3. Formado por GE02.Niza y BC01.Lyon. (b) Grupo 4. Formado por BC.Lucerna.

Figura 6.12: Grupos creados por el algoritmo. Curvas medianas de temperaturas de impulsión (I) y retorno (R) acompañadas del número de datos estudiados (entre paréntesis).

- CALENTADORAS

Del mismo modo que con las bombas enfriadoras, la agrupación óptima forma 4 grupos. La clasificación resultante de las 9 bombas de agua calentadoras aparecen en las figuras 6.13 y 6.14. Posteriormente, en las figuras 6.15 y 6.16, desglosamos los 4 grupos formados para ver las bombas de agua que los forman y el número de datos *correctos* con los que cada bomba llegó al análisis final.

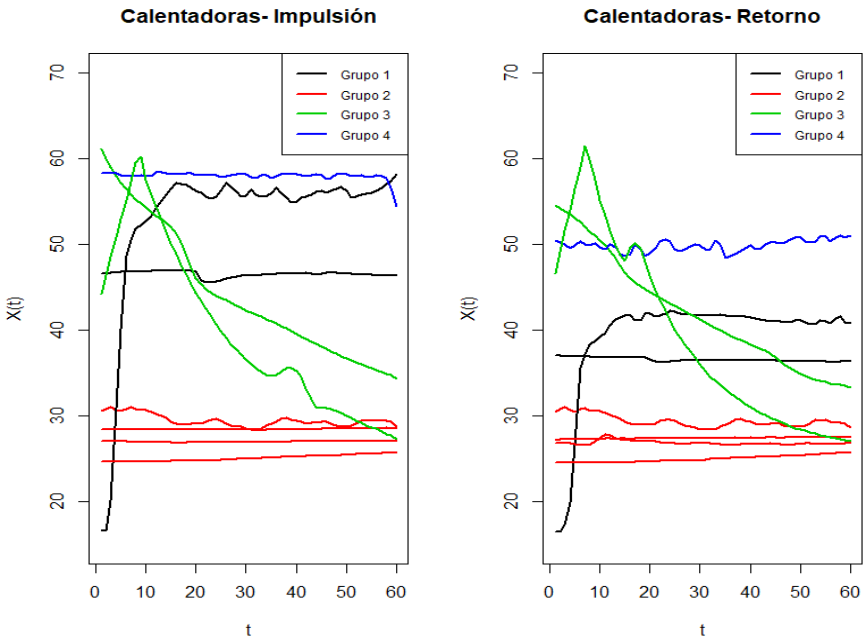


Figura 6.13: Curvas medianas de las calentadoras separadas por grupos según su patrón de comportamiento.

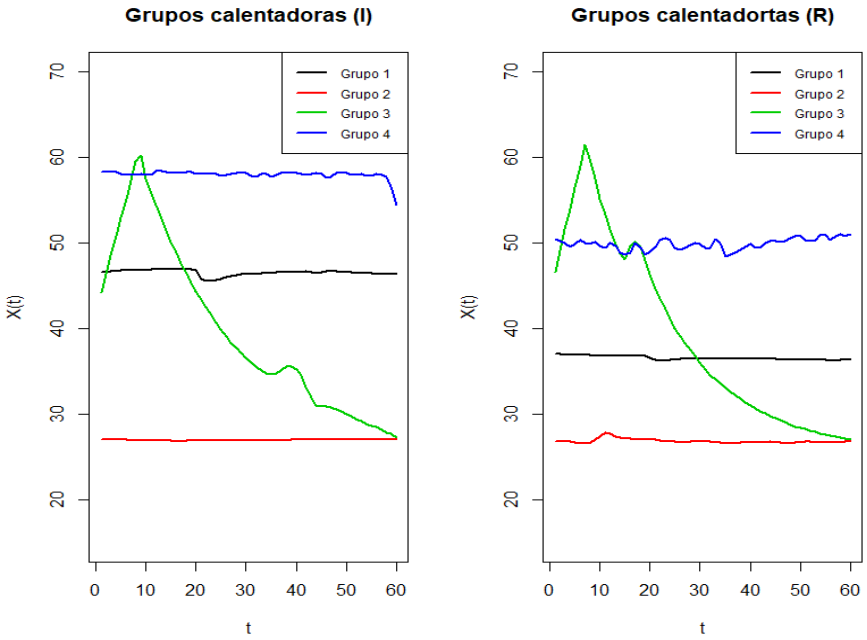
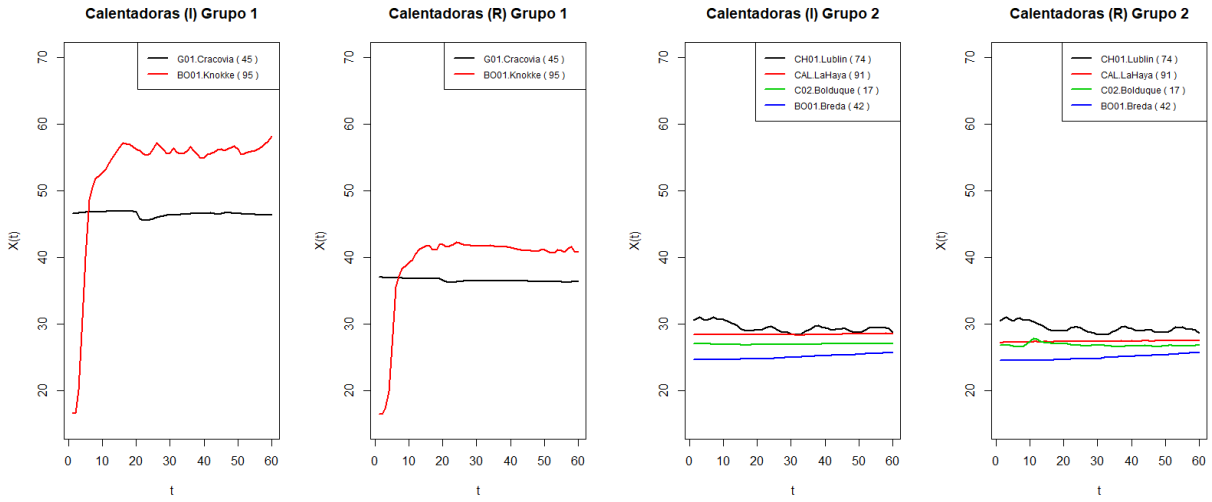


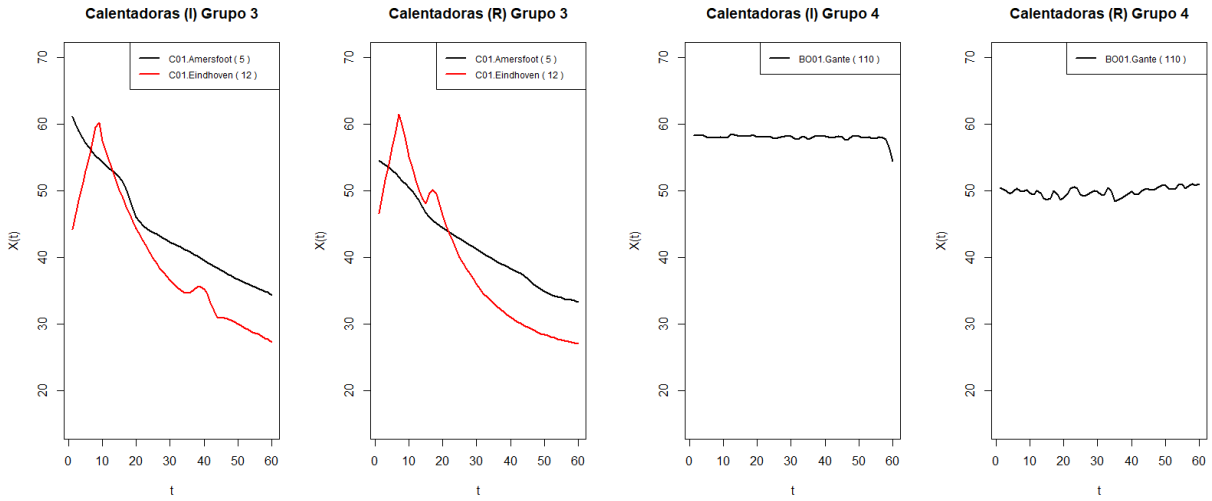
Figura 6.14: Curvas medianas de cada grupo de calentadoras creado.



(a) Grupo 1. Formado por GD1.Cracovia y BC01.Knokke.

(b) Grupo 2. Formado por CH01.Lublín, CAL.LaHaya, C02.Bolduque y BC01.Breda.

Figura 6.15: Grupos creados por el algoritmo. Curvas medianas de temperaturas de impulsión (I) y retorno (R) acompañadas del número de datos estudiados (entre paréntesis).



(a) Grupo 3. Formado por C01.Amersfoot y C01.Eindhoven.

(b) Grupo 4. Formado por BO01.Gante.

Figura 6.16: Grupos creados por el algoritmo. Curvas medianas de temperaturas de impulsión (I) y retorno (R) acompañadas del número de datos estudiados (entre paréntesis).

- BOMBAS MULTIFUNCIÓN

En el caso de bombas de agua multifunción hay que tener en cuenta que habrá, por un lado, una división en grupos para las bombas cuando estén calentando y, por otro, una división en grupos cuando estén enfriando. En esta muestra, el número de bombas que realizan ambas funciones es 3.

[Calentando]
En base a los días en que las bombas multifunción estuvieron calentando, la división ha resultado en 2 grupos mostrados en las figuras 6.17 y 6.18. Individualmente, mostramos los grupos y las bombas que los forman,en las figuras 6.19 y 6.20.

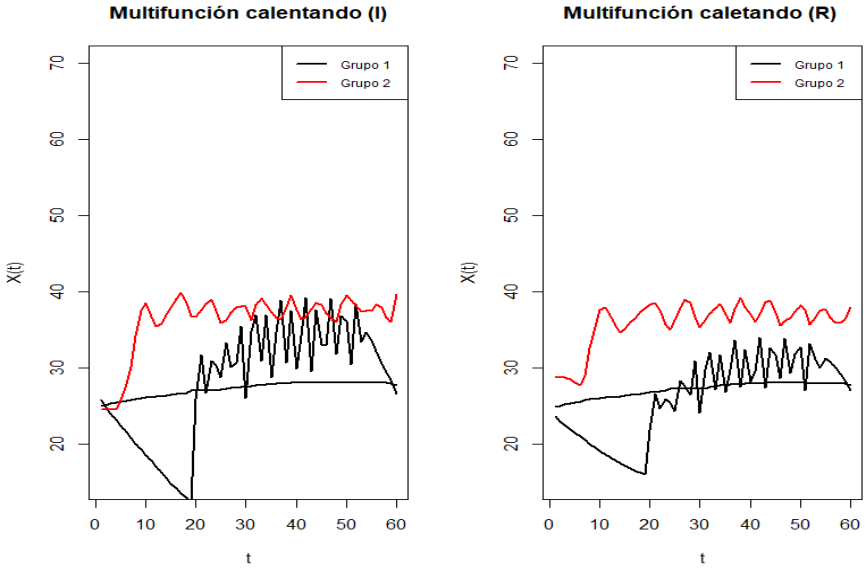


Figura 6.17: Curvas medianas de las bombas multifunción calentando, separadas por grupos según su patrón de comportamiento.

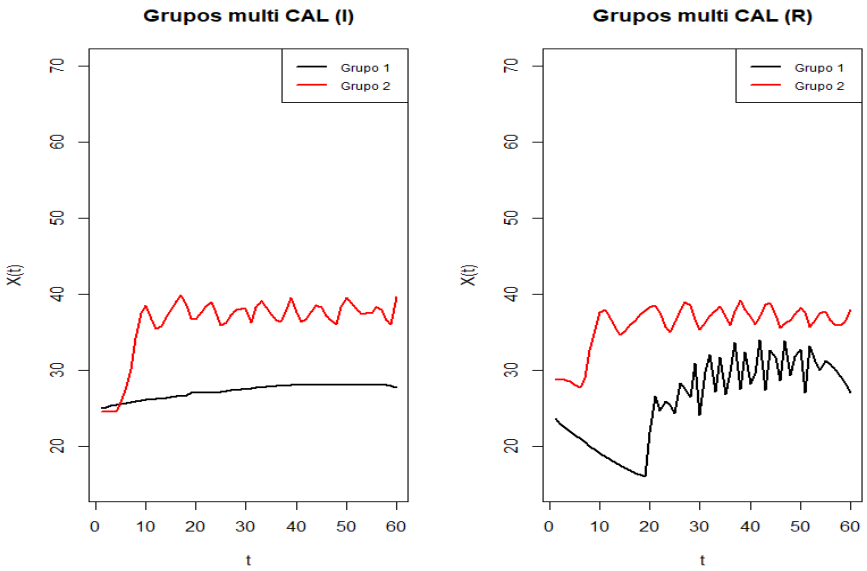
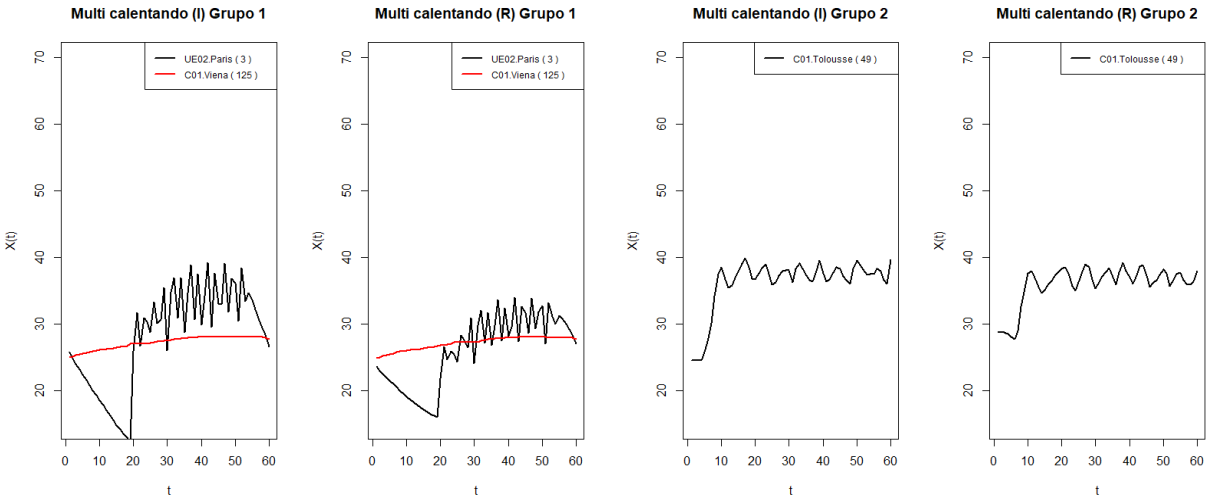


Figura 6.18: Curvas medianas de cada grupo de bombas multifunción calentando.



(a) Grupo 1. Formado por UE02.París y C01.Viena.

(b) Grupo 2. Formado por C01.Tolousse.

Figura 6.19: Grupos creados por el algoritmo. Curvas medianas de temperaturas de impulsión (I) y retorno (R) acompañadas del número de datos estudiados (entre paréntesis).

[Enfriando]
En base a los días en que las bombas multifunción estuvieron enfriando, la división ha resultado en 2 grupos mostrados en las figuras 6.17 y 6.18. Individualmente, mostramos los grupos y las bombas que los forman,en las figuras 6.19 y 6.20.

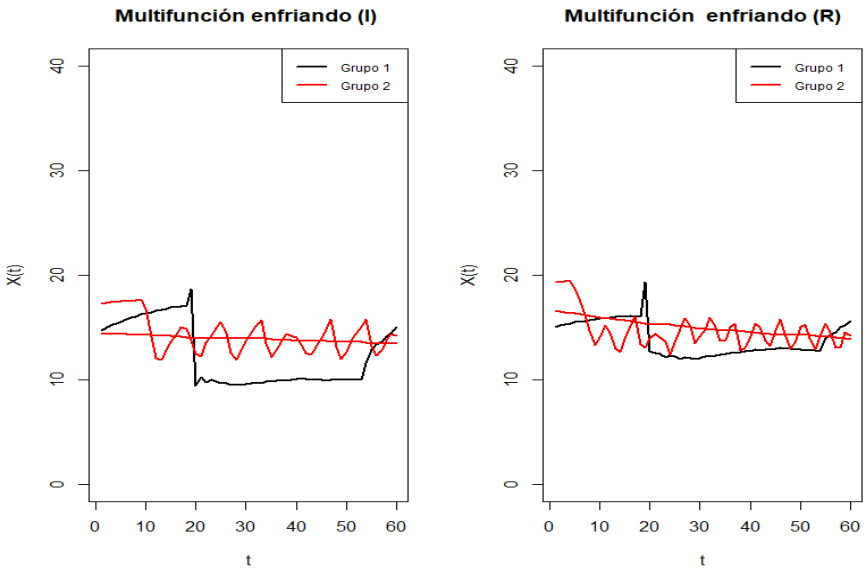


Figura 6.20: Curvas medianas de las bombas multifunción calentando, separadas por grupos según su patrón de comportamiento.

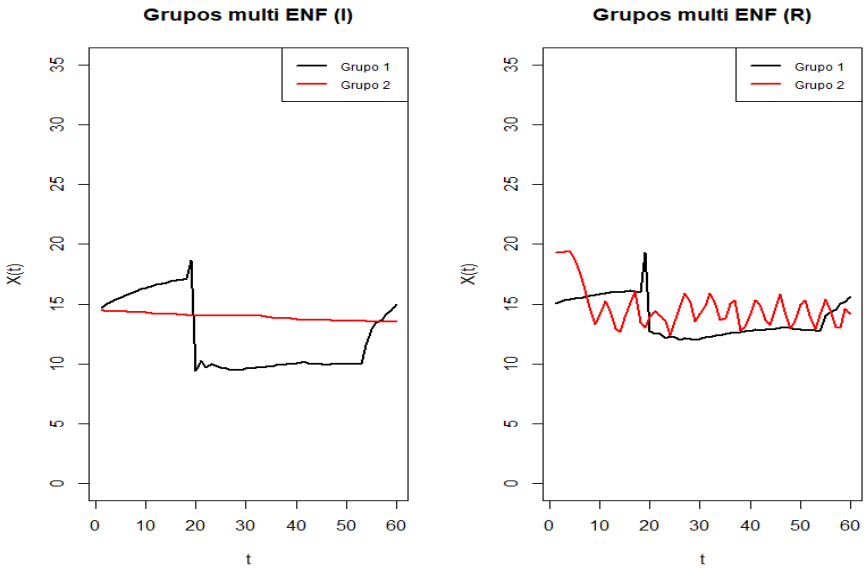
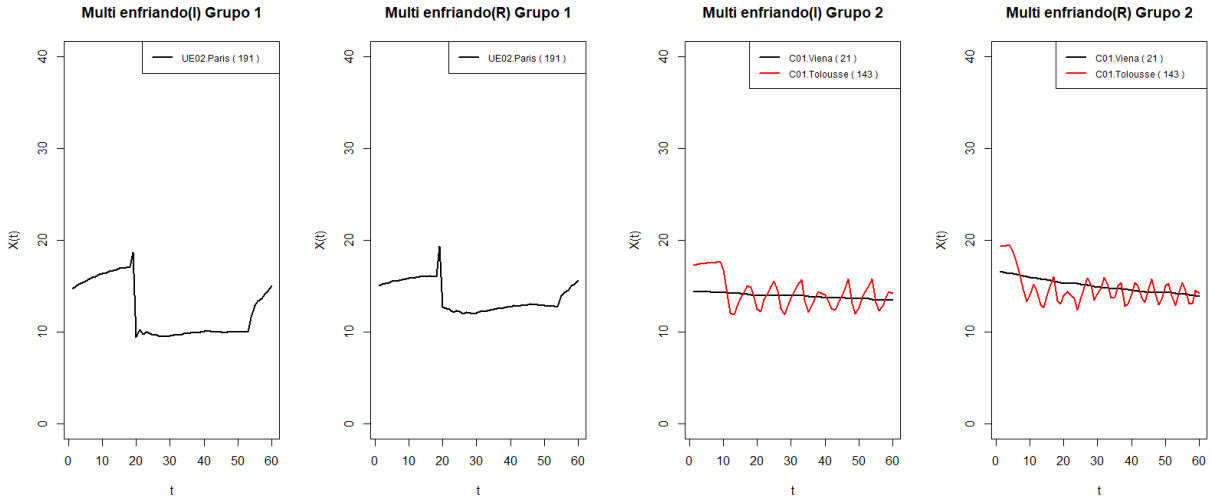


Figura 6.21: Curvas medianas de cada grupo de bombas multifunción calentando.



(a) Grupo 1. Formado por UE02.París.

(b) Grupo 2. Formado por C01.Viena y C01.Tolousse.

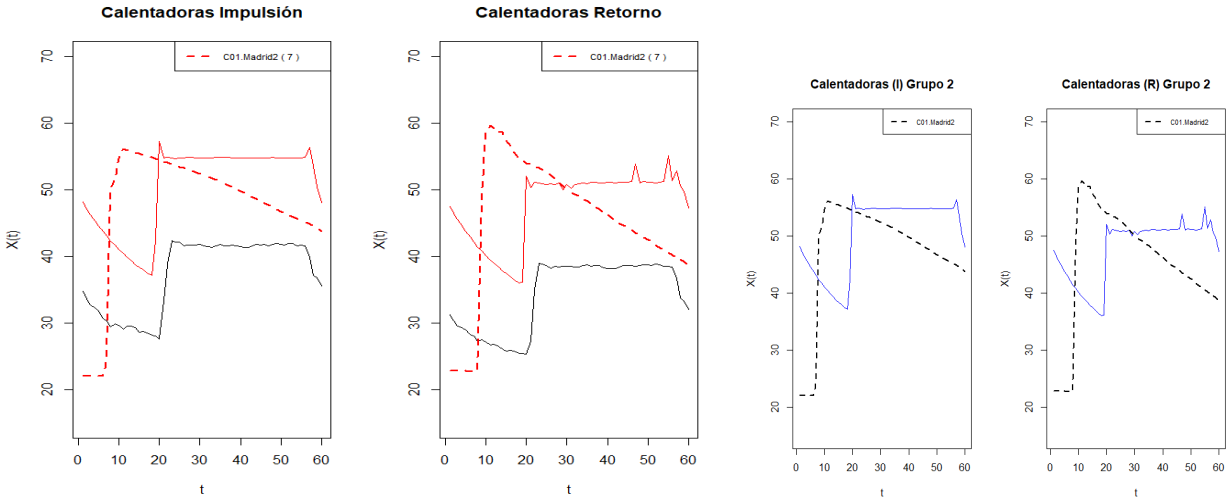
Figura 6.22: Grupos creados por el algoritmo. Curvas medianas de temperaturas de impulsión (I) y retorno (R) acompañadas del número de datos estudiados (entre paréntesis).

6.2. Clasificación supervisada

Para poder realizar una clasificación supervisada y poner a prueba el algoritmo *Clasificador.S* habíamos apartado dos bombas de agua de cada muestra. A continuación mostramos los grupos que el algoritmo ha predicho como los que más se ajustan a el comportamiento de las nuevas bombas.

6.2.1. Muestra 1. Península Ibérica

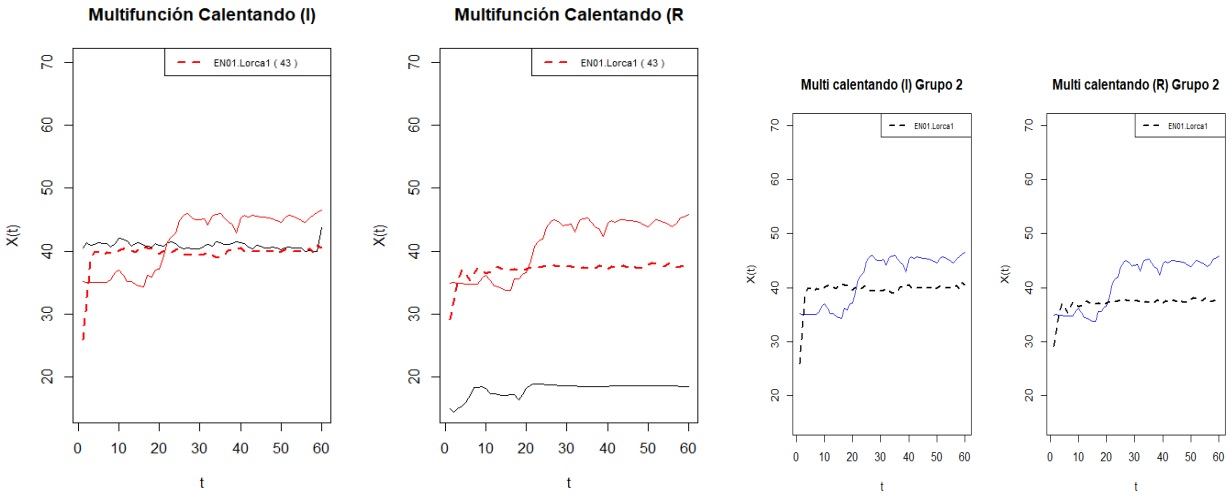
En la muestra 1, el resultado para las dos bombas de agua apartadas se muestran en las figuras 6.23, 6.24 y 6.25. Para alcanzar estos resultados, el algoritmo *Clasificador.S* ha tardado aproximadamente 1 minuto.



(a) C01.Madrid2 dentro de las calentadoras.

(b) Grupo predicho para C01.Madrid2; el 2.

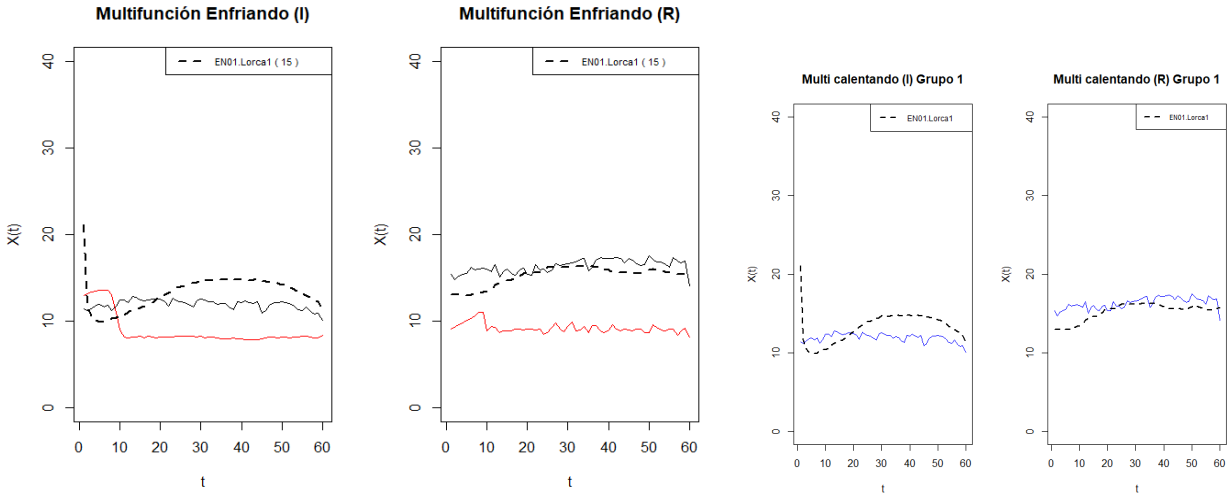
Figura 6.23: Resultados para la bomba de aguaC01.Madrid2, que está dentro de las calentadoras de la muestra 1.



(a) EN01.Lorca1 dentro de las multifunción calentando.

(b) Grupo predicho para EN01.Lorca1; el 2.

Figura 6.24: Resultados para la bomba de aguaEN01.Lorca1, que está dentro de las bombas multifunción de la muestra 1.



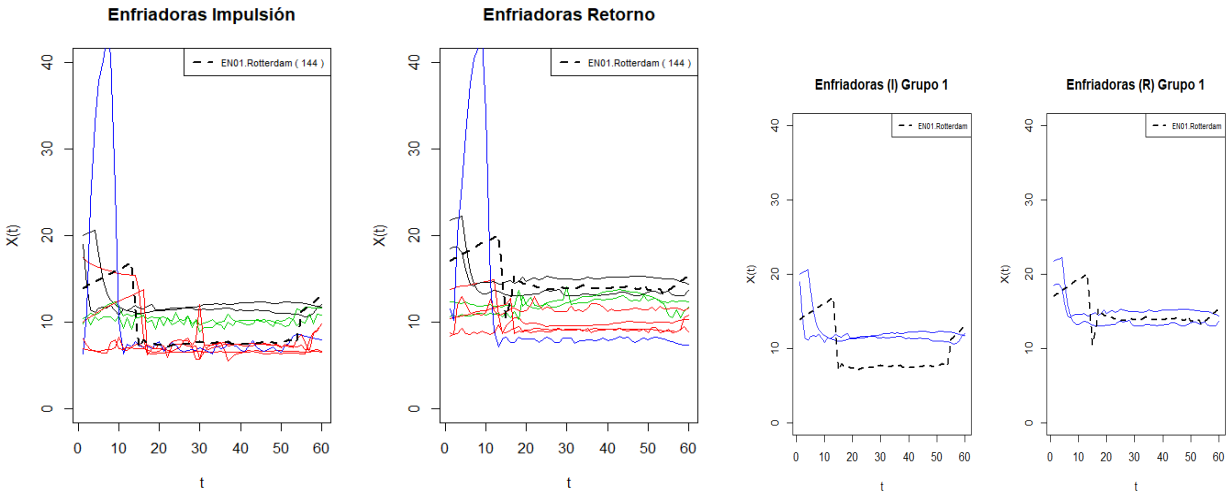
(a) EN01.Lorca1 dentro de las multifunción enfriando.

(b) Grupo predicho para EN01.Lorca1; el 1.

Figura 6.25: Resultados para la bomba de agua EN01.Lorca1, que está dentro de las bombas multifunción de la muestra 1.

6.2.2. Muestra 2. Europa central

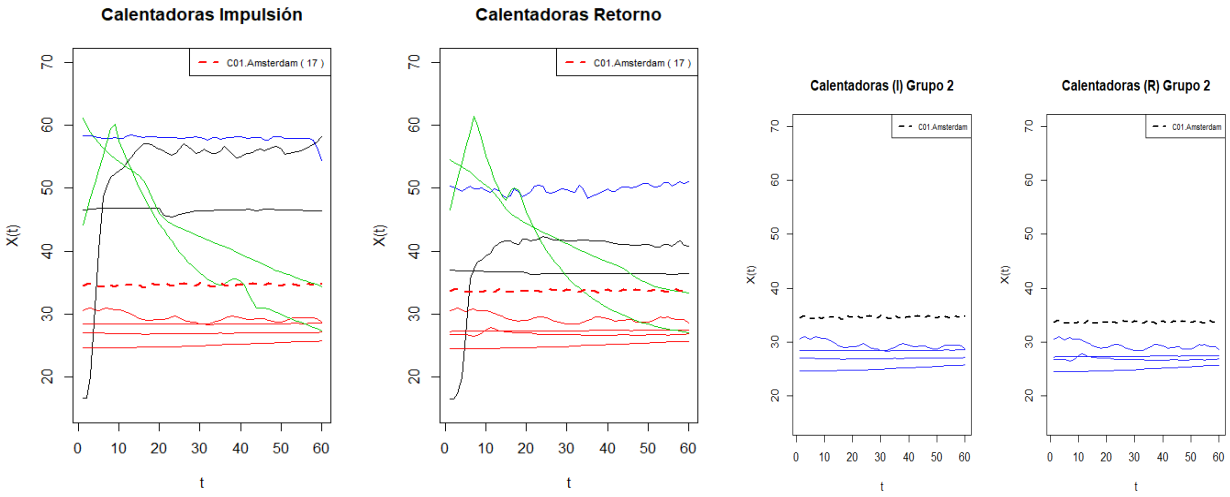
En el caso de la muestra 2, los resultados para las dos bombas de agua apartadas se muestran en las figuras 6.26 y 6.27. Para alcanzar estos resultados, el algoritmo *Clasificador.S* ha tardado aproximadamente 1 minuto.



(a) EN01.Rotterdam dentro de las enfriadoras.

(b) Grupo predicho para EN01.Rotterdam; el 1.

Figura 6.26: Resultados para la bomba de agua EN01.Rotterdam, que está dentro de las enfriadoras de la muestra 2.



(a) C01.Amsterdam dentro de las calentadoras.

(b) Grupo predicho para C01.Amsterdam; el 2.

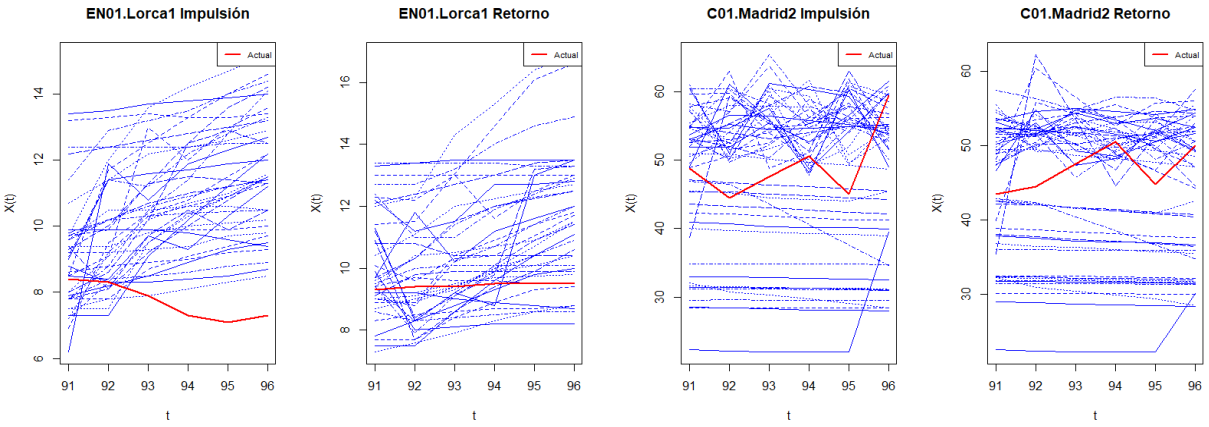
Figura 6.27: Resultados para la bomba de agua C01.Amsterdam, que está dentro de las calentadoras de la muestra 2.

6.3. Detección de incidencias

En este apartado, con el objetivo de mostrar el rendimiento del algoritmo *Detección*, hemos elegido 10 bombas de agua al azar de cada muestra y le hemos aplicado este algoritmo[40]. Para ello, hemos tenido en cuenta el último día en los datos disponibles, que en este caso es un lunes, y hemos cogido un intervalo de tiempo de hora y media hacia atrás (*long*). En los siguientes apartados se expone que posibles problemas ha detectado el algoritmo.

6.3.1. Muestra 1. Península Ibérica

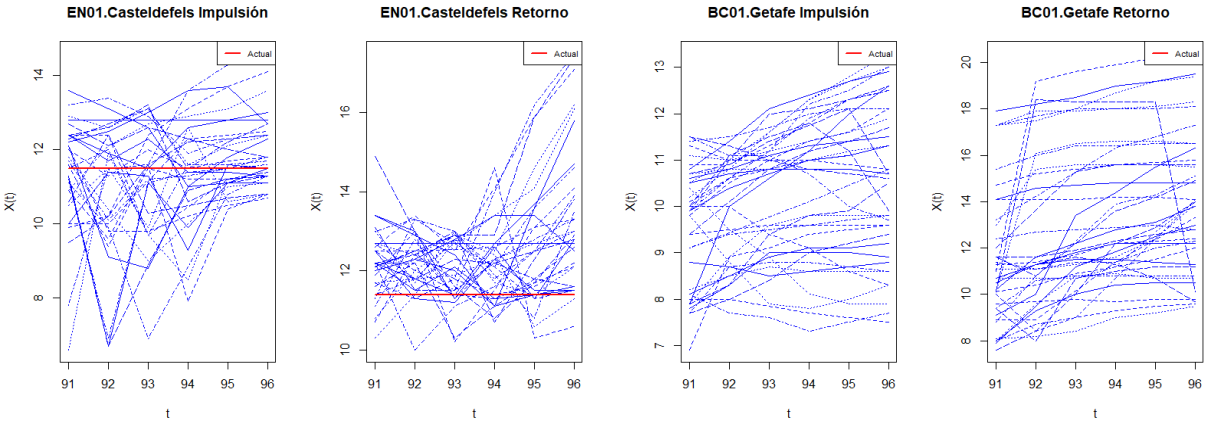
El análisis de detecciones para la muestra 1 se muestra en las figuras 6.28, 6.29, 7.30, 7.31 y 7.32. En los siguientes gráficos exponemos lo que ha detectado el algoritmo *Detección* en una muestra aleatoria de 10 bombas de agua y tardando aproximadamente 2 minutos.



(a) Curva de impulsión fuera de rango.

(b) Curva de impulsión fuera de rango. Curva de retorno fuera de rango.

Figura 6.28: Detección de incidencias en las bombas: EN01.Lorca1 y C01.Madrid2.

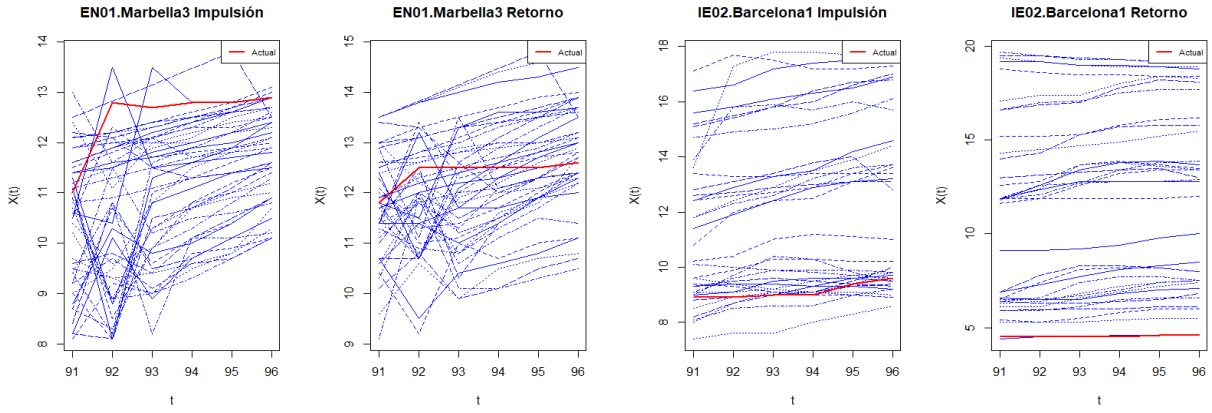


(a) Temperatura de impulsión por encima de la de retorno siendo enfriadora. Temperatura de impulsión constante. Temperatura de retorno constante.

(b) Curva de impulsión fuera de rango. Curva de retorno fuera de rango. Temperatura de impulsión por encima de la de retorno siendo enfriadora.Temperatura de impulsión constante. Temperatura de retorno constante.

Figura 6.29: Detección de incidencias en las bombas: EN01.Casteldefels y BC01.Getafe.

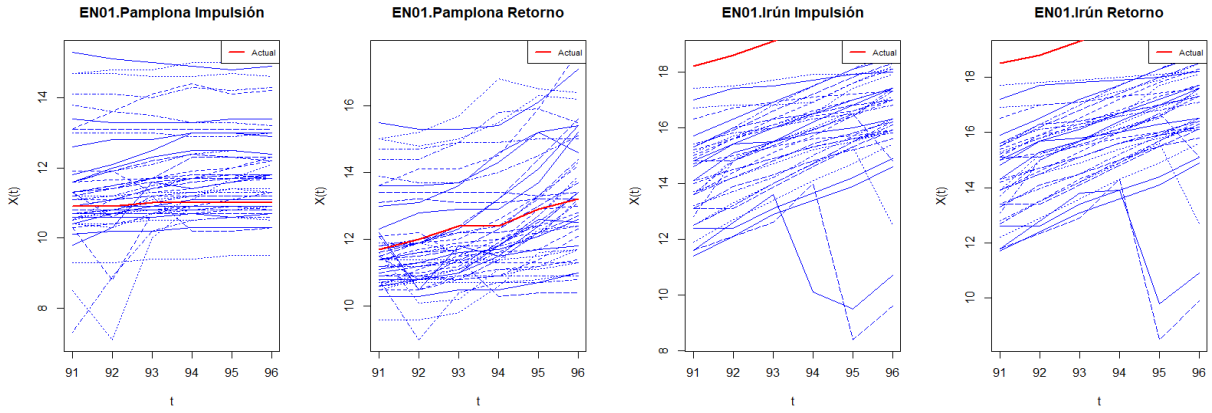
[40]Las demás bombas aparecen en el anexo: Detección 2



(a) Curva de impulsión fuera de rango. Temperatura de impulsión por encima de la de retorno siendo enfriadora.

(b) Curva de retorno fuera de rango. Temperatura de impulsión por encima de la de retorno siendo enfriadora

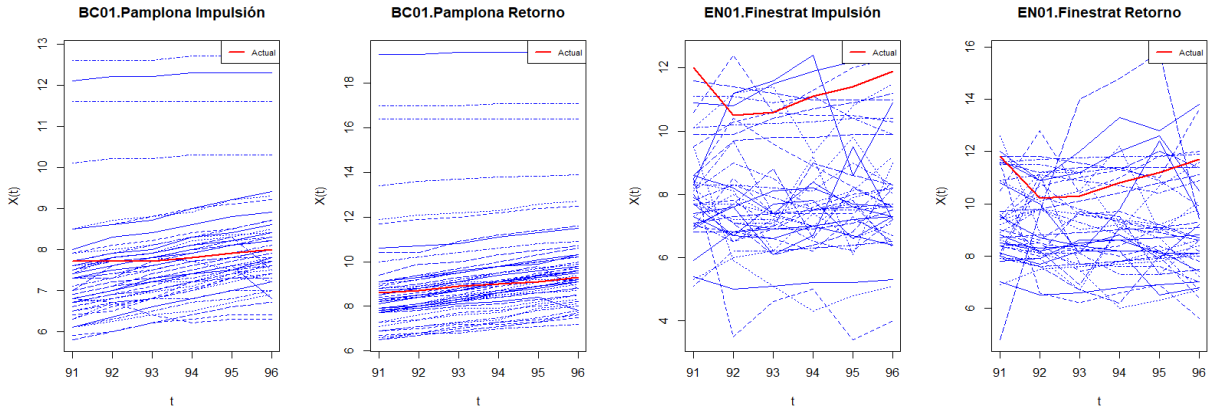
Figura 6.30: Detección de incidencias en las bombas: EN01.Marbella3 y IE02.Barcelona1.



(a) No se detectó ningún problema.

(b) Curva de impulsión fuera de rango. Curva de retorno fuera de rango.

Figura 6.31: Detección de incidencias en las bombas: EN01.Pamplona y EN01.Irún.



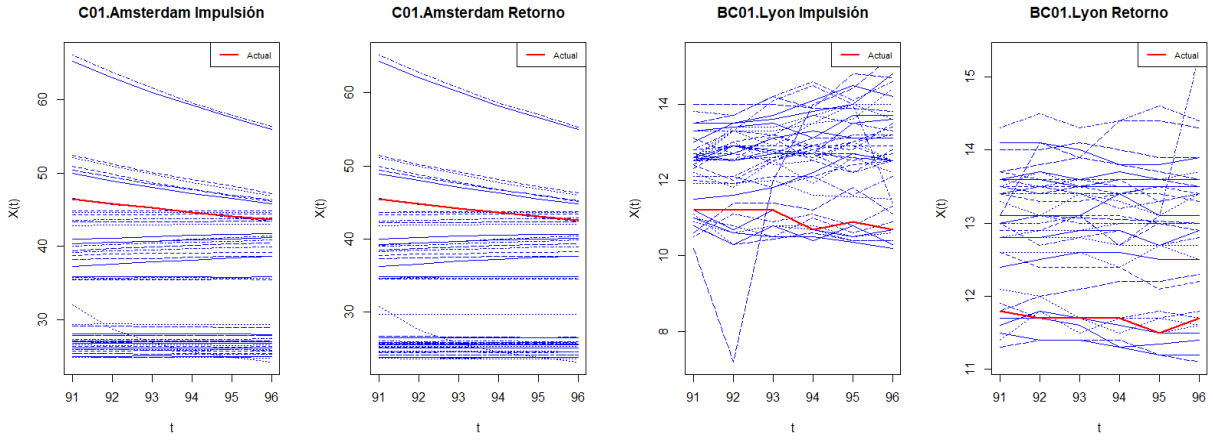
(a) No se detectó ningún problema

(b) Curva de impulsión fuera de rango. Temperatura de impulsión por encima de la de retorno siendo enfriadora.

Figura 6.32: Detección de incidencias en las bombas: BC01.Pamplona y EN01.Finestrat.

6.3.2. Muestra 2. Europa central

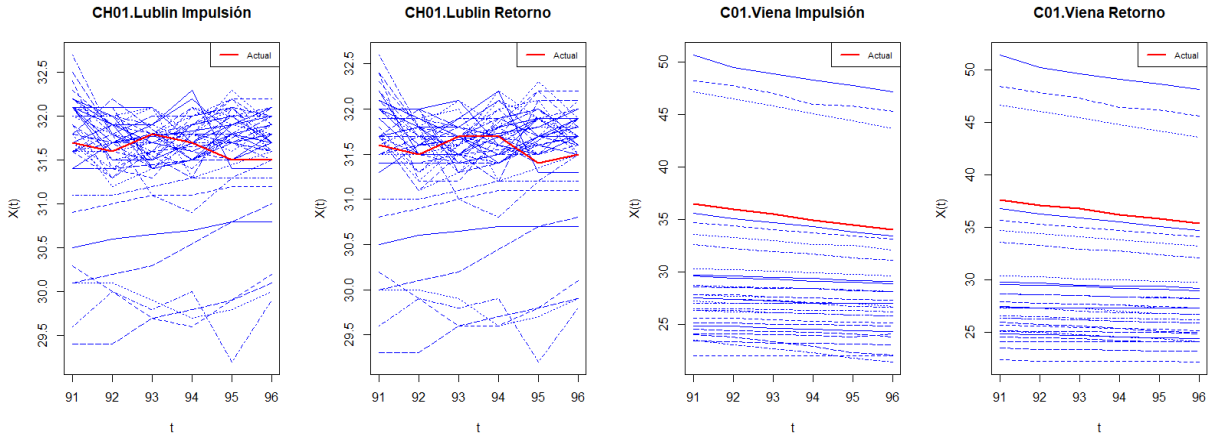
El análisis de detecciones para la muestra 2 se refleja en las figuras 6.33, 6.34, 6.35, 6.36 y 6.37. En ellas exhibimos lo que ha detectado el algoritmo *Detección* en 2 minutos aproximadamente.



(a) Curva de impulsión fuera de rango. Curva de retorno fuera de rango.

(b) Curva de impulsión fuera de rango. Curva de retorno fuera de rango.

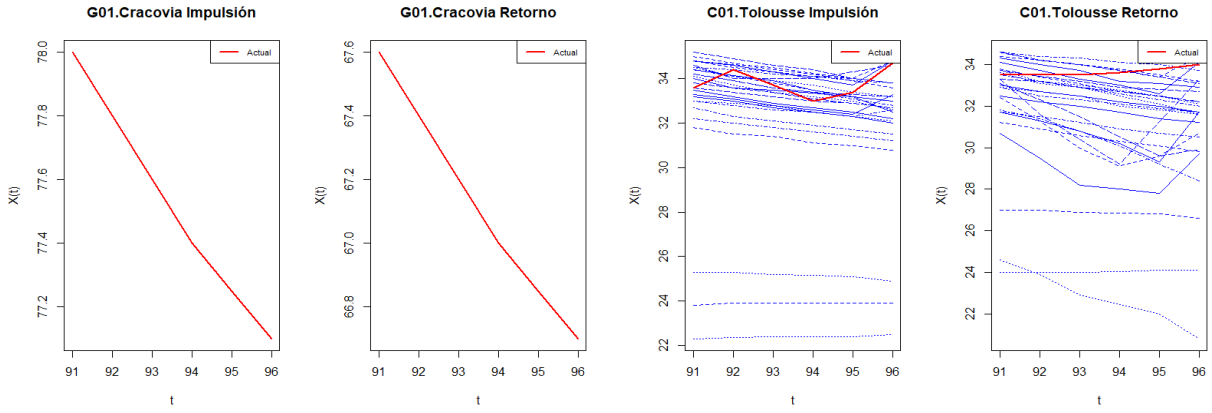
Figura 6.33: Detección de incidencias en las bombas: C01.Amsterdam y BC01.Lyon.



(a) No se detecta ningún problema.

(b) Curva de impulsión fuera de rango. Curva de retorno fuera de rango. Actualmente está enfriando a temperaturas muy altas.

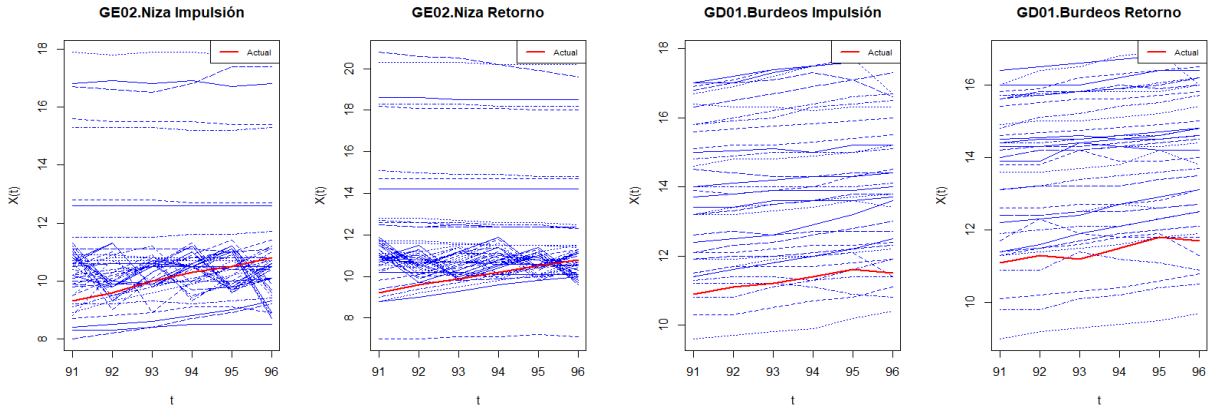
Figura 6.34: Grupos creados por el algoritmo. Curvas medianas de temperaturas de impulsión (I) y retorno (R) acompañadas del número de datos estudiados (entre paréntesis).



(a) Curva de impulsión fuera de rango.

(b) Curva de impulsión fuera de rango.

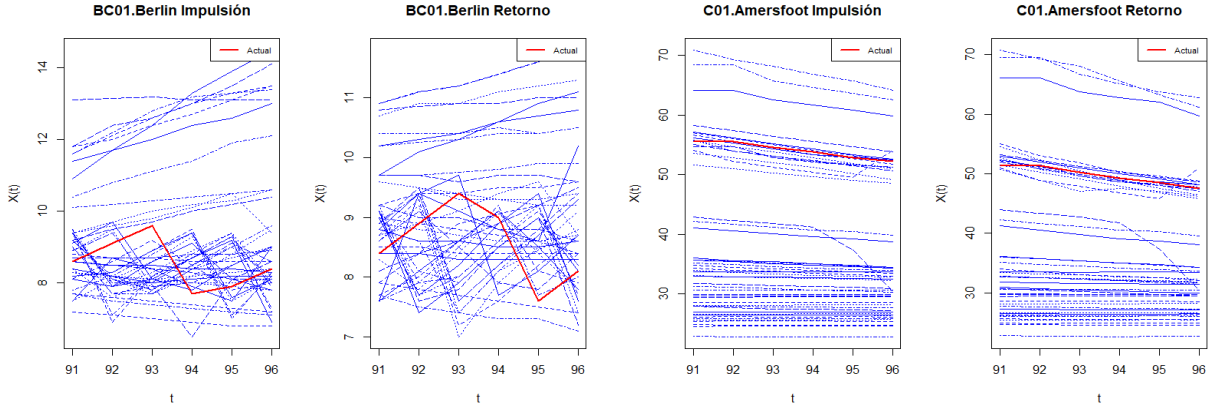
Figura 6.35: Detección de incidencias en las bombas: G01.Cracovia y C01.Tolousse.



(a) No se detecta ningún problema.

(b) Curva de retorno fuera de rango.

Figura 6.36: Detección de incidencias en las bombas: GE02.Niza y GD01.Burdeos.



(a) Temperatura de impulsión por encima de la de retorno siendo enfriadora.

(b) Curva de impulsión fuera de rango. Curva de retorno fuera de rango.

Figura 6.37: Detección de incidencias en las bombas: BC01.Berlín y C01.Amersfoot.

Capítulo 7

Conclusiones

La principal conclusión a la que se quería llegar con este trabajo de fin de máster era que los algoritmos creados reportaran una utilidad significativa a la empresa que recurriera a ellos. Una vez finalizado este trabajo queda claro que los tres algoritmos son útiles para cualquier empresa, que trabaje en el sistema retail y con instalaciones de climatización basadas en bombas de agua. Los dos primeros algoritmos, relativos a clasificaciones de bombas de agua, significan un avance en el contexto de bombas de agua; ya que hasta ahora no se conoce ningún otro método de agrupación de bombas de agua desde un enfoque de datos funcionales. Desde otro punto de vista, ha quedado también justificado que, el algoritmo *Detección* proporciona una mejora en la detección de posibles problemas futuros en este tipo de máquinas. Se ha demostrado que es capaz de anticipar incidencias reales, ahorrándole a la empresa todos los costes que supondrían las incidencias no detectadas a tiempo.

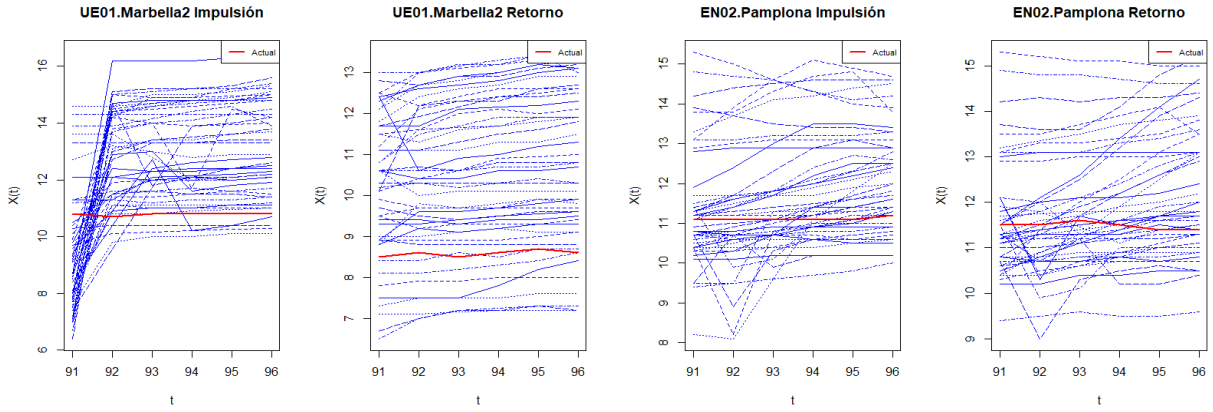
Por otro lado, destacar que afrontar este análisis desde el enfoque de datos funcionales ha sido un éxito. La información que reportan los datos, como curvas diarias, es incomparable con la que me podría proporcionar análisis del tipo: series de tiempo o modelos de regresión (ambos testeados con pésimos resultados). Los datos funcionales no solo nos han ayudado a conseguir alcanzar el objetivo de este trabajo, sino que le han dado a este TFM un carácter innovador; imprescindible en la época de la información y tecnología en la que vivimos.

Por último, la elaboración de este trabajo permite ver que aún queda mucho por hacer. A pesar de que el rendimiento de los algoritmos es bueno, se podría llevar a cabo clasificaciones en base a más variables, proporcionar más información de los grupos creados o afinar más la detección de futuras incidencias. Dicho de otra forma, aunque los algoritmos aquí presentados están completos y son operativos, es posible seguir mejorándolos para reportar mayor utilidad a la empresa.

Apéndice A

Detección 2

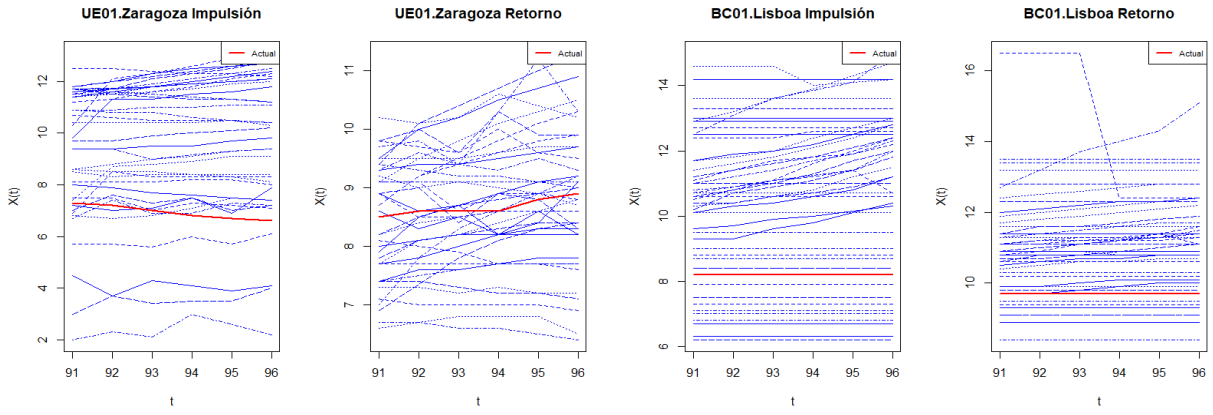
A.1. Muestra 1



(a) Curva de impulsión fuera de rango. Temperatura de impulsión por encima de la retorno siendo enfriadora.

(b) No se detecta nada.

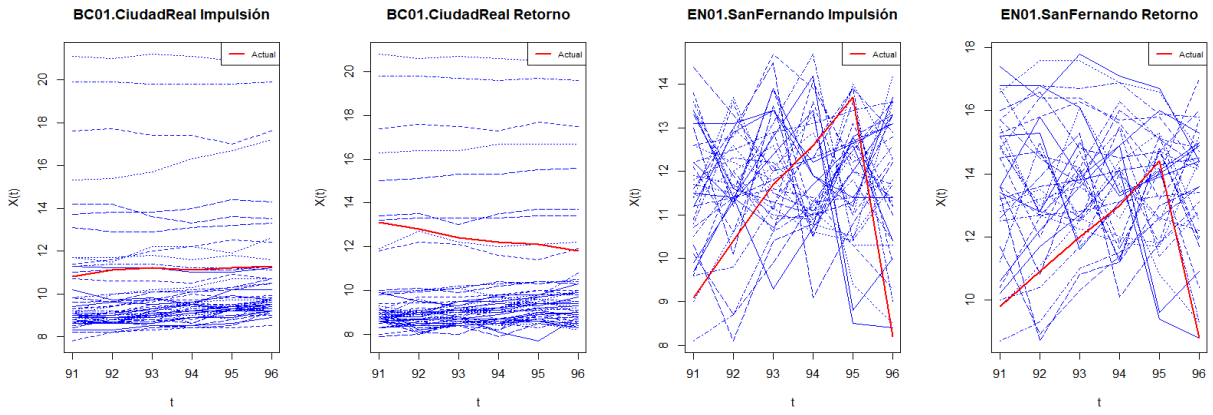
Figura A.1: Detección de incidencias en las bombas: UE01.Marbella2 y EN02.Pamplona.



(a) Curva de impulsión fuera de rango.

(b) Temperatura de impulsión constante. Temperatura de retorno constante.

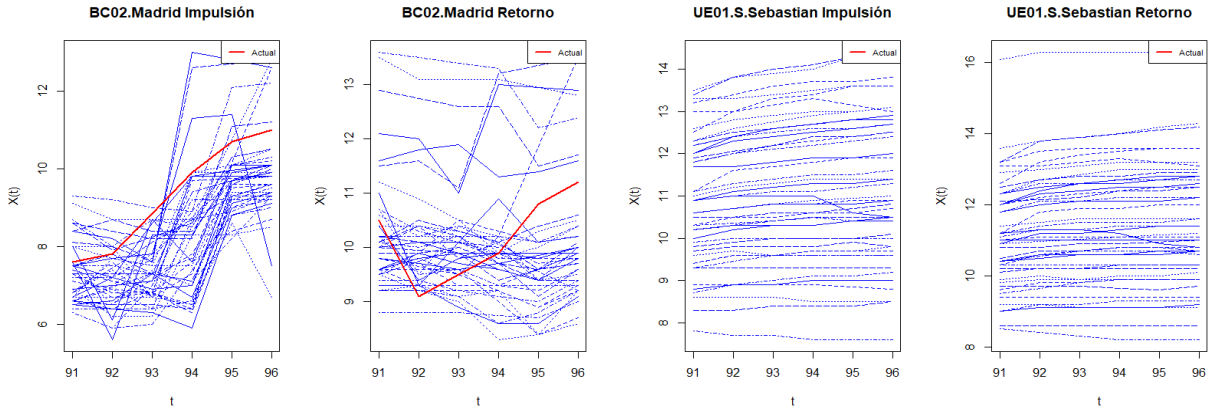
Figura A.2: Detección de incidencias en las bombas: UE01.Zaragoza y BC01.Lisboa.



(a) Curva de impulsión fuera de rango. Curva de retorno fuera de rango.

(b) Curva de impulsión fuera de rango. Curva de retorno fuera de rango.

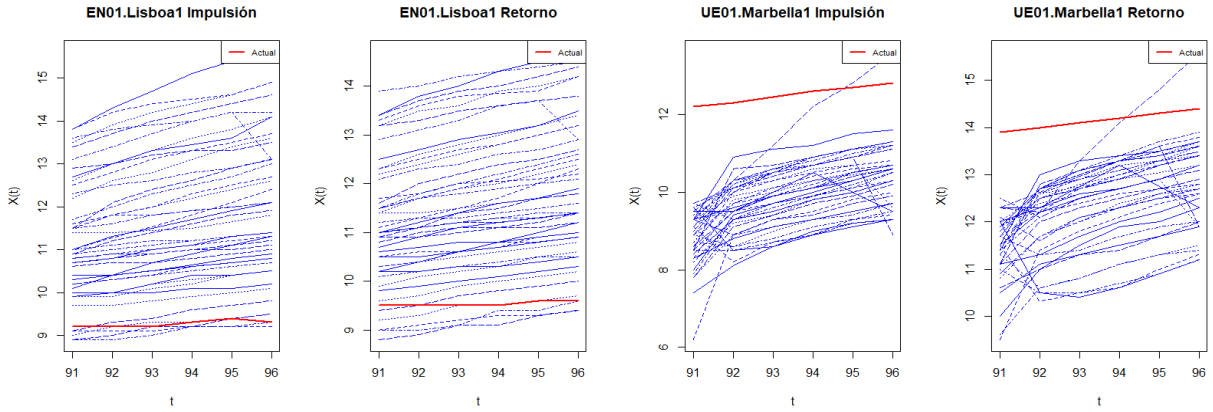
Figura A.3: Detección de incidencias en las bombas: BC01.CiudadReal y EN01.SanFernando.



(a) Curva de impulsión fuera de rango. Curva de retorno fuera de rango.

(b) Curva de impulsión fuera de rango. Curva de retorno fuera de rango.

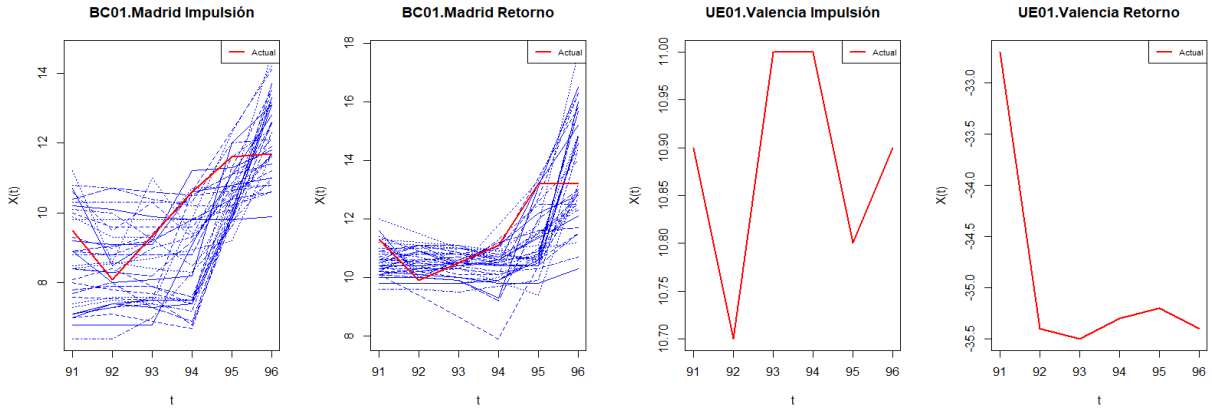
Figura A.4: Detección de incidencias en las bombas: BC02.Madrid y UE01.S.Sebastián.



(a) Curva de retorno fuera de rango.

(b) Curva de impulsión fuera de rango. Curva de retorno fuera de rango.

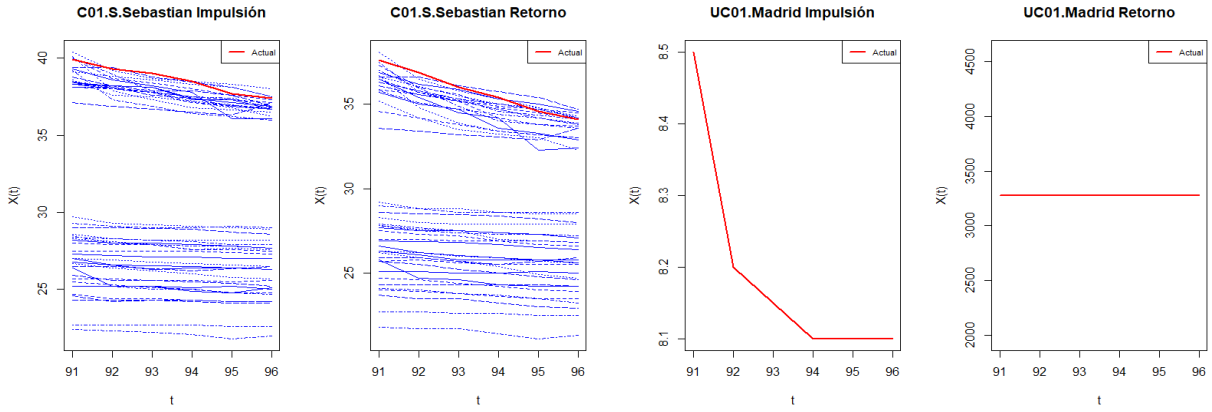
Figura A.5: Detección de incidencias en las bombas: EN01.Lisboa1 y UE01.Marbella1.



(a) Curva de retorno fuera de rango.

(b) Curva de retorno fuera de rango.

Figura A.6: Detección de incidencias en las bombas: BC01.Madrid y UE01.Valencia.



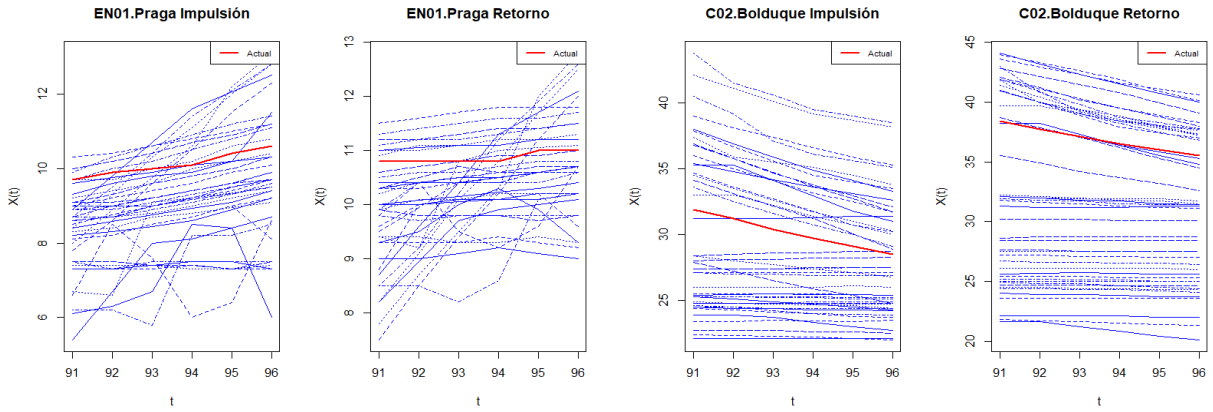
(a) Curva de impulsión fuera de rango. Curva de retorno fuera de rango.

(b) Curva de retorno fuera de rango.

Figura A.7: Detección de incidencias en las bombas: C01.S.Sebastián y UC01.Madrid.

A.2. Muestra 2

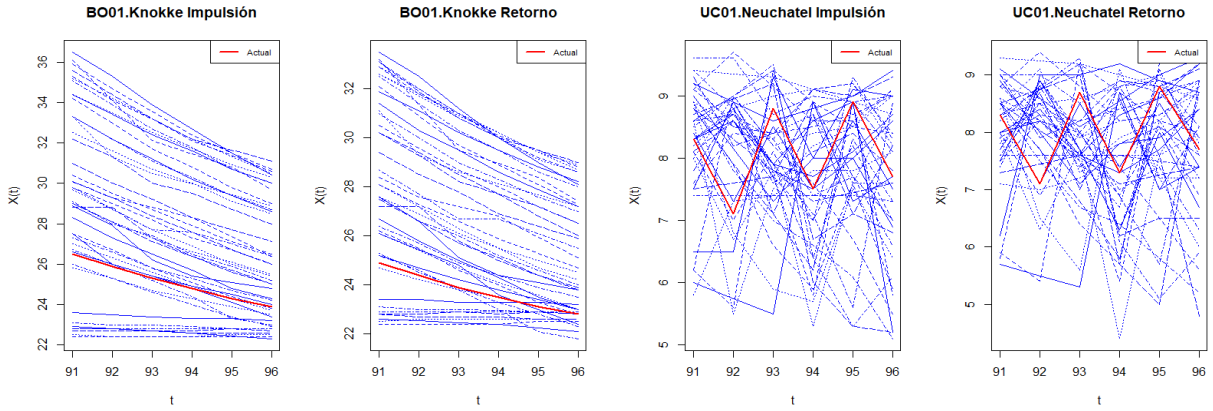
Al margen de las bombas de agua aquí mostradas, dentro de esta muestra, en 4 bombas se han detectado demasiados valores faltantes en la curva actual. Por este motivo, el algoritmo no ha proporcionado ningún gráfico. Estas bombas son: BC.Lucerna, BO01.Breda, EN01.Viena y EN01.Rotterdam.



(a) No se detectó ningún problema.

(b) Curva de impulsión fuera de rango. Temperatura de impulsión por debajo de la de retorno siendo calentandora.

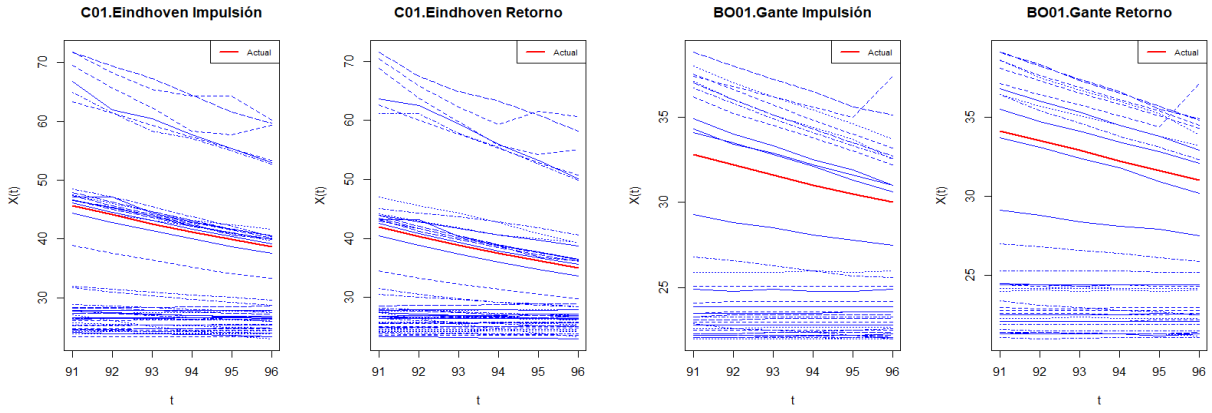
Figura A.8: Detección de incidencias en las bombas: EN01.Praga y C02.Bolduque.



(a) No se detecta ningún problema.

(b) No se detecta ningún problema.

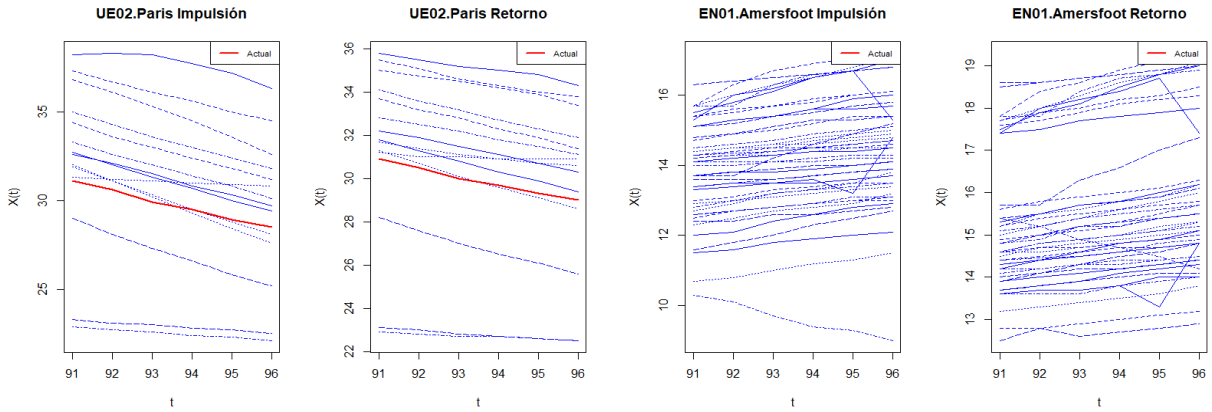
Figura A.9: Detección de incidencias en las bombas: BC01.Knokke y UC01.Neuchatel.



(a) Curva de impulsión fuera de rango. Curva de retorno fuera de rango.

(b) Curva de impulsión fuera de rango. Curva de retorno fuera de rango. Temperatura de impulsión por debajo de la de retorno siendo calentadora.

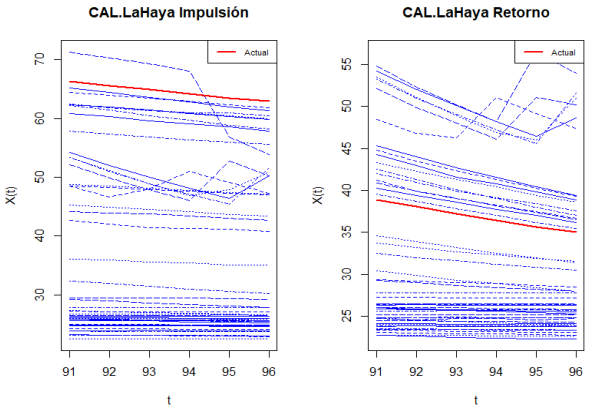
Figura A.10: Detección de incidencias en las bombas: C01.Eindhoven y BO01.Gante.



(a) No se detecta ningún problema.

(b) Curva de impulsión fuera de rango. Curva de retorno fuera de rango.

Figura A.11: Detección de incidencias en las bombas: UE02.París y EN01.Amersfoot.



(a) Curva de impulsión fuera de rango. Curva de retorno fuera de rango.

Figura A.12: Detección de incidencias en la bomba CAL.LaHaya.

Bibliografía

[1] Cuevas A. 2014. A partial overview of the theory of statistics with functional data. Journal of Statistical Plannning and Inference 147: 1 - 23.

[2] Febrero M, Galeano P, González - Manteiga W. 2008. Outlier detection in functional data by depth measures, with applcation to indentify abnormal NOx levels. Environmetrics 19: 331 - 345.

[3] Ferreira L, Hitchcock D.B. 2009. A comparison of Hierarchical Methods for Clustering Functional Data, Communications in Statistics - Simulation and Computation, 38:9, 1925 - 1949.

[4] Li J, Cuesta-Albertos J.A, Liu R.Y. 2012. DD- Classifier: Nonparametric Classification Procedure Based on DD-plot. Journal of the American Statistical Association - Junio 2012.

[5] Baíllo A, Cuevas A. 2008. Supervised Classification for Functional Data: A Theoretical Remark and Some Numerical Comparisons. In: Functional and Operatorial Statistics. Contributions to Statistics. Physica-Verlag HD.