Estudio de técnicas de clustering y detección de anomalías aplicado a fresadoras CNC

Patricia Lucía Barreno Recio

Anibal Bregón Bregón
y
Miguel Angel Martinez Prieto

Grado en Ingeniería Informática de Servicios y Aplicaciones
Escuela de Ingeniería Informática de Segovia
Universidad de Valladolid

Agradecimientos

Gracias a todos aquellos que creísteis en mí y me habéis ayudado a llegar hasta aquí, y muy en especial a los que más me han sufrido, a mis padres que siempre están ahí cuando les necesito y a mi hermano que ha sabido ser un buen hermano pequeño y convertirse en un gran hermano mayor.

y gracias a Anibal y Miguel Ángel
no solo por no darme por perdida con este TFG
sino por todo lo que me habéis enseñado estos años
y al resto profesores de la escuela por vuestra dedicación
y por abrirme la puerta a esta apasionante
y a veces abrumadora profesión
y sin olvidar en el camino a mis compañeros
que no solo me habéis acompañado y apoyado estos años
sino de los que también he aprendido mucho.

Sin todos vosotros el inicio de este viaje no hubiera sido el mismo.

En resumen, fácilmente se observará que
los ingeniosos son siempre dotados de fantasía y
los verdaderamente imaginativos jamas dejan de ser analíticos.
Edgar Allan Poe

Resumen

Este trabajo fin de grado presenta una introducción a la Industria 4.0 y a los métodos de pronóstico de fallos y realiza un estudio de métodos de clustering y de detección de anomalías que se puedan aplicar en los datos de monitorización de procesos de mecanizado. En este estudio se considerará el caso de que estos métodos se apliquen en datos de series temporales incluyendo diferentes distancias, representaciones y enfoques que se pueden aplicar en este tipo de datos.

Para completar este trabajo se realizara una parte práctica donde se probaran algunos de los métodos estudiados. En ella se aplicaran métodos de clustering en datos de dos fresadoras CNC (control numérico por computadora) , comparando los resultados obtenidos con diferentes algoritmos y distancias aplicadas. Finalmente, los resultados de estos modelos se podrán visualizar a través de un dashboard para facilitar una comprobación y comparación visual del trabajo realizado.

Palabras clave

Palabras claves: Industria 4.0, máquinas CNC, métodos de pronóstico , RUL (vida util restante), clustering, PAA (aproximación agregada por partes), DTW (distorsión de tiempo dinámico), PLR (Representación lineal por partes) , motivos de series temporales, discordias, shapelets, perfil de matriz, detección de anomalías, K-Means, DBSCAN, OPTICS, clustering jerárquico.

Abstract

This final degree project presents an introduction to Industry 4.0 and failure prediction methods and a study of clustering and anomaly detection methods that can be applied to machining process monitoring data. This study will consider the case where these methods are applied to time series data including the different distances, representations and approaches that can be applied to this type of data.

To complete this work, a practical part will be carried out where some of the methods studied will be tested. The clustering methods will be applied to data from two CNC milling machines (computer numerical control), comparing the results obtained with different algorithms and distances applied. Finally, the results of these models can be visualised using a dashboard to facilitate the visual verification and comparison of the work performed.

Keywords

Keywords: Industry 4.0, CNC machine, Prognostics Methods, RUL (Remaining Useful Life), clustering, PAA (piecewise aggregate approximation), DTW (dynamic time warping), PLR (piecewise linear representation), time series motifs, discords, shapelets, matrix profile, anomaly detection, K-Means, DBSCAN, OPTICS, hierarchical clustering.

Índice general

Resumen **ix**

Índice de figuras **xix**

Índice de tablas **xxii**

1 Introducción **1**

1.1 Motivación 2

1.2 Objetivos 3

1.3 Alcance del Sistema 4

1.4 Identificación del entorno tecnológico 5

1.4.1 Lenguaje de Programación 5

1.4.2 Framework 5

1.4.3 Librerías 6

1.4.4 Entorno de desarrollo 8

1.5 Organización del documento 8

2 Plan de Proyecto **11**

2.1 Metodología 11

2.2 Fase de Trabajo y estimación Temporal 12
2.3 Estimación 17
2.4 Presupuesto 21

I Dominio del problema 23

3 Industria 4.0 25

3.1 Características de la Industria 4.0 26
3.2 Desafíos de la Industria 4.0 28
3.3 Aplicaciones de la Industria 4.0 28
3.3.1 Diagnóstico y Pronóstico 30
3.3.2 CBM y PHM 33

4 Mecanizado, Fresado y Máquinas CNC 37

4.1 Mecanizado 37
4.1.1 Mecanizado por Arranque de viruta 38
4.2 Fresado 39
4.2.1 Fresadora 40
4.2.2 Herramientas de corte: Fresa 42
4.3 Máquinas CNC 44

5 Estado del Arte 51

II Estudio Teórico 55

6 Aprendizaje no supervisado y clustering 57

6.1 Clustering 58
6.2 Tipos de Algoritmos 61
6.2.1 Clustering basado en partición o en representantes 61
6.2.2 Clustering basado en densidad 63
6.2.3 Clustering jerárquico 64
6.2.4 Clustering basado en cuadrícula 66
6.2.5 Clustering basado en modelos 68
6.2.6 Otros algoritmos de clustering 71
6.3 Medidas de Evaluación 71
6.3.1 Índices Internos 71
6.3.2 Índices Externos 76
6.4 Clustering en aprendizaje semi-supervisado 78
6.5 Clustering de Series de Tiempo 79
6.5.1 Enfoques de clustering de series temporales 81
6.6 Representación de series temporales 82
6.6.1 PAA 83
6.6.2 APCA 84
6.6.3 SAX 84
6.6.4 DFT Transformada discreta de Fourier 84
6.6.5 DWT Transformada discreta de wavelet 85
6.6.6 Otras Representaciones 85
6.6.7 Resumen 85
6.7 Medidas de Similitud de series temporales 86

6.7.1 Distancia DTW 89
6.7.2 Distancia de edición 91
6.7.3 LCSS 92
6.7.4 TWED 93
6.7.5 Resumen 94
6.8 Prototipos de Series temporales 94
6.9 Clustering de subsecuencias de series temporales 95
6.9.1 Segmentación de series temporales 95
6.10 Clustering Significativo de subsecuencias de series temporales 99
6.11 Motivos, Shapelets, Discordias y Perfil de matriz 100

7 Detección de Anomalías **105**
7.1 Aprendizaje no supervisado para la detección de anomalías 108
7.1.1 Basados en métodos de vecinos más cercanos KNN 109
7.1.2 Basadas en métodos estadísticos 110
7.1.3 Clustering para detección de anomalías 111
7.1.4 Otros enfoques 113
7.2 Detección de anomalías en series temporales 113
7.3 Aprendizaje profundo para detección de anomalías 115

III Estudio Práctico **117**

8 Algoritmos de Clustering Aplicados **119**
8.1 Basados en Partición 119

8.1.1 K-Medias . . . 119
8.1.2 Fuzzy C-means . . . 123
8.2 Basados en Densidad . . . 124
8.2.1 DBSCAN . . . 124
8.2.2 OPTICS . . . 129
8.3 Clustering Jerárquico . . . 134
8.3.1 Aglomerativo . . . 134

9 Casos de estudio: Fresadoras 137

9.1 Caso de Estudio: Fresadora CNC (CIDAUT) . . . 137
9.1.1 Análisis de los Datos . . . 138
9.1.2 Preparación y limpieza de datos . . . 144
9.1.3 Análisis de los Modelos Utilizados . . . 149
9.1.4 Resumen y conclusiones . . . 159
9.2 Caso de Estudio: Fresadora Kaggle . . . 159
9.2.1 Análisis datos . . . 159
9.2.2 Preparación de datos . . . 161
9.2.3 Análisis Modelos . . . 162
9.2.4 Resumen y conclusiones . . . 168

10 Caso de Estudio: Dashboard 171

10.1 Análisis . . . 172
10.1.1 Descripción de Actores . . . 172
10.1.2 Requisitos de Usuario . . . 172

10.1.3 Requisitos Funcionales 177
10.1.4 Requisitos de Información 179
10.1.5 Requisitos No Funcionales 185
10.2 Diseño 185
10.2.1 Arquitectura 186
10.2.2 Diagrama Lógico de Datos 186
10.2.3 Diagramas de Clases 187
10.2.4 Diagramas de Secuencia 187
10.2.5 Diseño de la Interfaz 188
10.3 Detalles de Implementación 189
10.4 Pruebas realizadas 190
10.5 Manual de Instalación 191
10.6 Manual de Usuario 191
10.7 Visualización de los datos y los resultados 192

IV Parte Final **195**

11 Conclusiones **197**

11.1 Conclusiones CIDAUT 197
11.2 Conclusiones KAGGLE 199
11.3 Líneas de Trabajo Futuro 200

Bibliografía **201**

Índice de figuras

2.1 Metodología CRISP-DM 12
2.2 Adaptación al proyecto 13
2.3 Gantt del proyecto 16
2.4 Tareas Gantt 17
2.5 Mapa guía para la comprensión del entorno del proyecto 23

3.1 Fases de un programa de pronósticos 31
3.2 Ciclo de mantenimiento PHM 34

4.1 Fresadora componentes 41
4.2 Componentes de una máquina CNC 47
4.3 Componentes de la unidad de control de un sistema CNC 48

5.1 Mapa guía del estudio 55

6.1 Grupos Compactos y encadenados 59
6.2 Componentes del clustering 60
6.3 Proceso de clustering 60
6.4 Ejemplo de clustering basado en partición 62
6.5 Enlaces Jerárquico 65

6.6 Dendograma clustering jerárquico 66

6.7 Impacto del número de celdas clustering cuadrícula 67

6.8 Mapa Autoorganizado 70

6.9 Gráfico de silueta 73

6.10 Características ST 79

6.11 Enfoques clustering de series temporales 82

6.12 Muestreo PAA 84

6.13 SAX 84

6.14 Diferencia entre DTW y distancia euclidiana 89

6.15 Camino DTW 90

6.16 Restricciones Globales 91

6.17 LCSS 93

6.18 Segmentación de una serie temporal 96

6.19 Segmentación por aproximación lineal 97

6.20 Ejemplo de segmentación obtenida por ventana deslizante, los segmentos obtenidos se representan con diferentes colores junto con las rectas a las que se aproxima cada uno de ellos. 98

6.21 Interpretación Perfil de Matriz 101

6.22 Perfil de Matriz 102

6.23 PMP 102

7.1 Tipos de Anomalías 107

7.2 Mapa guía datos utilizados 117

8.1 Diagrama Codo 121

8.2 Diagrama de Voronoi 122

8.3 Punto Central, Punto Borde y ruido 125

8.4 Distancia al núcleo y distancia de alcance 130

8.5 Diagrama Alcance y extracción de clusters 132

8.6 Umbral de distancia para determinar los grupos 135

9.1 Corriente todos los días examinados 139

9.2 Comportamientos inusuales 139

9.4 Corriente 05-02 Forma habitual de la corriente durante su arranque y calentamiento 140

9.3 Corriente en el arranque de la fresadora 140

9.5 Corriente Incrementos pequeños de corriente a periodos constantes de tiempo 141

9.6 Corriente 29-01 Picos de Corriente Habituales 141

9.7 Corriente diversos días Final 142

9.8 Acelerómetro 142

9.9 Acelerómetro rotura del 30 de Enero 143

9.10 Corriente, Acelerómetro, Sonido y Temperatura de los días 10, 23 y 28 de enero 143

9.11 Muestreo PAA en fresadora CIDAUT 144

9.12 Segmentación Puerta 146

9.13 Segmentación Aproximación Lineal 146

9.14 Segmentación propia CB 147

9.15 Segmentación propia CB 148

9.16 Segmentación con Pelt 148

9.17 Segmentación Arranque 149

9.18 Fases Fresadora 149

9.19 Clustering de puntos de serie de tiempo 150

9.20 Dendrograma Jerárquico Arranque Completo 151

9.21 Clustering jerárquico del arranque completo 151

9.22 Clustering jerárquico enlace completo con distancia DTW 152

9.23 Diagrama del codo y silueta 152

9.24 Arranque K-means 153

9.25 OPTICS con DTW aplicado al arranque completo 153

9.26 Diagrama de alcance arranque 154

9.27 OPTICS con DTW aplicado al arranque 154

9.28 DBSCAN y OPTICS aplicado al apagado 155

9.29 Dendrogramas Clustering Jerárquico Euclidiana 155

9.30 Clustering Activa Jerárquico con dist euclidiana y enlace ward 156

9.31 Clustering obtenido aplicando clustering jerárquico con enlace completo y distancia DTW a la fresadora en activo 156

9.32 Clusters identificados de la fresadora activa aplicando clustering jerárquico con enlace promedio y distancia DTW 157

9.33 Diagramas de codo e índices de silueta y clusters obtenidos con K-means . 157

9.34 Resultado de K-means y distancia euclidiana con la fresadora en activo . . 158

9.35 Diagramas de alcance obtenido con distancia euclidiana y DTW 158

9.36 Resultado de DBSCAN y distancia DTW con la fresadora en activo 158

9.37 Resumen Experimentos 161

9.38 Resumen Fases por experimentos 162

9.39 Comparación enlaces Jerárquicos en Experimentos 163

9.40 Resultados K-Means experimentos 164
9.41 Diagrama alcance Optics Distancia Eucliana y DTW 164
9.42 Comparación enlaces Jerárquicos en Fases 166
9.43 Resultados K-Means fases 166
9.44 Diagrama alcance de las Fases con distancia Euclidiana y DTW 167

10.1 Diagrama casos de uso 172
10.2 Diagrama entidad-relación 179
10.3 Diagrama de clases 187
10.4 Diagrama de secuencia de CU-05 188
10.5 Visualización dashboard 190
10.6 Visualización dashboard: Inicio 192
10.7 Visualización dashboard: Análisis CIDAUT 193
10.8 Visualización dashboard: Clustering CIDAUT 193
10.9 Visualización dashboard: Análisis Kaggle 194
10.10Visualización dashboard: Clustering Kaggle 194

Índice de tablas

6.1 Clustering duro y difuso 63

6.2 Matriz de contingencia 76

9.1 Clustering Experimentos distancia Euclidiana 165

9.2 Clustering Experimentos distancia DTW 165

9.3 Clustering Fases distancia Euclidiana 167

9.4 Clustering Fases distancia DTW 168

10.1 Actores 172

10.2 CU-1 Mostrar Fresadora CIDAUT 173

10.3 CU-02 Cambiar día a mostrar 173

10.4 CU-03 Mostrar Fresadora Kaggle 174

10.5 CU-04 Ir a clustering 174

10.6 CU-05 Seleccionar Clustering 175

10.7 CU-06 Ir a clustering 176

10.8 CU-07 Mostrar resultados clustering Kaggle 176

10.9 CU-08 Cambiar Tabla Análisis Inicial 177

10.10RFuncional 178

10.11RI-01 ResultadoClustering 180
10.12RI-02 subsecuencia relevante 181
10.13RI-03 Fase 181
10.14RI-04 Experimento 182
10.15RI-05 Dato Agrupado 182
10.16RI-06 medición fresadora CIDAUT 183
10.17RI-R1 conforma 183
10.18RI-R2 pertenece 184
10.19RI-R3 agrupa 184
10.20RNFuncional 185
10.21RNegocio 185
10.22DI-01 Diseño de la ventana de resultados fresadora CIDAUT 189
10.23P-01 prueba visualización de algoritmo aplicado con solo una distancia . . 191

Capítulo 1

Introducción

La Industria 4.0 es una nueva etapa industrial que se basa en la revolución que ha supuesto la integración de las tecnologías de la información y la comunicación (TIC) en los sistemas de fabricación convencionales.

Con la Industria 4.0 aumenta el número de sensores y con ello los datos provenientes de la monitorización de las máquinas de fabricación. Esto implica una oportunidad de aprovechar estos datos para mejorar la eficiencia de la producción. Una de estas formas es mediante su uso para detectar patrones inusuales, esto puede permitir entre otros, detectar un mal funcionamiento de la máquina o un desgaste en la herramienta de corte. Además, esta información se puede aprovechar posteriormente para conseguir programar de forma más adecuada las tareas de mantenimiento y aprovechar de manera más óptima los recursos.

En este trabajo se realizará una introducción al mecanizado y sus implicaciones y aplicaciones en la Industria 4.0, así como su relación con el aprendizaje automático. Tras ello se realizará un estudio de técnicas clustering y detección de anomalías que se pueden aplicar a datos de monitorización de procesos de mecanizado.

Para acabar se realizará un estudio práctico de técnicas de clustering aplicados a datos de monitorización de varias fresadoras CNC. Los resultados obtenidos podrán ser visualizados mediante un dashboard que servirá de apoyo para la comparación y evaluación de dichos resultados.

1.1 Motivación

Dentro de los procesos de mecanizado convencional se encuentra el fresado, una forma de mecanizado que se realiza por arranque de viruta mediante una herramienta rotativa llamada fresa. Las máquinas que realizan este tipo de mecanizado se conocen como fresadoras. Estas máquinas permiten obtener piezas de gran precisión a través del movimiento longitudinal y transversal de la mesa, del carro transversal y del brazo soporte a través de los ejes X,Y y Z, así como con el movimiento rotativo de la fresa.

Las fresadoras CNC utilizan un sistema con control numérico por computador para realizar automáticamente una serie de operaciones siguiendo las instrucciones establecidas desde un ordenador. Dicho de otra forma, trabajan automáticamente mediante un programa de control que maneja los movimientos y posiciones de los elementos que componen la fresadora de forma automática, haciendo innecesario que un operario maneje la máquina manualmente. El uso de estas máquinas ha permitido aumentar la precisión y productividad al reducir tiempos de trabajo.

Las fresadoras son muy utilizadas en industria [45] y como todas las herramientas sufren desgastes y pueden producirse roturas que afecten a su uso habitual y por tanto afectar a la producción industrial. Por ello, aprovechar los datos que se recopilan durante su uso para detectar comportamientos inusuales puede permitir corregir estos comportamientos lo más pronto posible. La corrección de estos comportamientos, por ejemplo sustituyendo la fresa o ajustando parámetros de la máquina, puede reportar un aumento en la productividad ahorrando tanto tiempo como dinero.

Una de las aplicaciones del aprendizaje automático es la detección de anomalías. Esta tarea requiere poder identificar un comportamiento anómalo de uno que no lo es, lo cual no siempre es trivial. Las condiciones normales de funcionamiento de la fresadora pueden variar, las fresadoras permiten trabajar con multitud de piezas así como con distintos materiales y con distintos tamaños de producción que producirán datos de monitorización diferentes. Esto impide conocer a priori que datos escapan del comportamiento normal. Además los datos provenientes de este control del funcionamiento de la máquina son desequilibrados, teniendo muchos más datos correspondientes a comportamientos de funcionamiento normales que de funcionamiento inusual, lo que complica su análisis. Esta situación hace que dentro del aprendizaje automático, la detección de anomalías se trate normalmente como un problema de aprendizaje no supervisado o semi supervisado (teniendo solo algunos ejemplos que suelen ser del comportamiento habitual).

El aprendizaje no supervisado evita la necesidad de conocer cuáles de los datos recopilados son anómalos y requiere de menos datos para entrenar. Estos algoritmos permiten definir los datos inusuales dinámicamente y evitan la necesidad de un conocimiento amplio del dominio de aplicación, es decir, evitan requerir un conocimiento en profundidad del funcionamiento de una fresadora CNC.

Además, no solo se necesita poder identificar comportamientos atípicos. Como se ha mencionado, las fresadoras realizan diferentes piezas a lo largo de su funcionamiento que obtendrán mediciones diferentes y no siempre los datos contienen información al respecto. Esto hace que sea de utilidad poder identificar patrones comunes que se repitan y que se correspondan con firmas de elaboración de piezas concretas. Una de las técnicas de aprendizaje automático no supervisado más utilizadas es el clustering, donde los datos se agrupan en función de una medida de similitud.

Las técnicas de clustering han sido ampliamente utilizadas tanto para datos estáticos como dinámicos. Dentro de los datos dinámicos nos encontramos las series de tiempo. Este tipo de datos es muy común en multitud de dominios, incluido la industria. Las características de las series de tiempo, pueden variar la forma de afrontar el problema.

No existe una técnica de aprendizaje que sea mejor que otra para cualquier problema lo que implica que se requiera comprobar que técnica es más eficaz para cada problema.

1.2 Objetivos

Muchos trabajos de detección de anomalías en entornos industriales presentan problemas debido a un elevado número de falsos negativos [40] que pueden impedir, por ejemplo que se detecte que la fresa está a punto de romperse o un elevado número de falsos positivos [70] que incrementan el coste y los tiempos de producción entre otras formas debido a tareas de mantenimiento innecesario .

Con este estudio se espera descubrir el comportamiento habitual (o diferentes comportamientos habituales para diferentes piezas) de una fresadora CNC, así como los valores que escapan a este comportamiento. Para ello dado que no existe conocimiento previo se estudiarán algoritmos de aprendizaje no supervisado para series temporales, con el objetivo de comprobar que técnica es más efectiva en las fresadoras estudiadas.

Para lograrlo, se estudiarán varios enfoques, representaciones de datos, distancias de si-

militud y algoritmos de clustering con el fin de comparar resultados.

Para la detección de anomalías no solo son importantes los factores anteriormente mencionados, sino también las características utilizadas. No todas las mediciones recopiladas de la monitorización del fresado son igual de buenas para poder detectar valores inusuales. En el caso de la fresadora del CIDAUT [1] se estudiará la capacidad de la corriente para realizar esta tarea.

Además de la fresadora mencionada, se realizarán pruebas con datos provenientes de otra fresadora CNC publicados por la universidad de Michigan [54]. Estos segundos datos, permitirán comprobar la efectividad de las técnicas aplicadas en la fresadora del CIDAUT en la detección de desgaste, así como, comprobar su validez en datos multivariados.

1.3 Alcance del Sistema

El trabajo se dividirá en varias partes:

- La primera parte del trabajo, consistirá en la familiarización con el problema. Se incluye entender qué es, en qué consiste y las implicaciones que trae el concepto de Industria 4.0, junto con la investigación del marco teórico para abordar el problema centrado en el clustering de series temporales y la detección de anomalías.

- En una segunda parte, se aplicarán algunos de los métodos investigados en dos fresadoras CNC una de ellas con datos temporales etiquetados y multivariantes y otra con datos temporales sin etiquetar y univariantes. En cada una de ellas se compararán las diferentes técnicas aplicadas.

- La parte final, consistirá en la elaboración de un dashboard que permita la visualización de los datos analizados, así como, la de los modelos realizados y los resultados obtenidos.

[1] Fundación para la Investigación y Desarrollo en Transporte y Energía (CIDAUT)

1.4 Identificación del entorno tecnológico

1.4.1 Lenguaje de Programación

Tanto el análisis de los datos, los modelos de aprendizaje automático, como el dashboard se realizarán en Python.

Python es un lenguaje de programación interpretado y de propósito general. Cuenta con una sintaxis simple y es multiparadigma permitiendo tanto programación estructurada, como orientada a objetos o funcional.

Actualmente es uno de los lenguajes de programación más utilizados y cuenta con una gran variedad de librerías para tareas de ciencias de datos.

1.4.2 Framework

Para la elaboración del dashboard, donde se mostrarán los datos analizados y los modelos realizados se utilizará el framework **Dash** [2] . Se trata de un framework de Python que se encuentra bajo licencia MIT. Dash permite crear aplicaciones web multiplataformas y adaptadas a dispositivos móviles de forma fácil para el análisis de datos sin necesidad de tener que escribir código HTML y Javascript y que está escrito sobre Flask, Plotly.js y React.js.

Dash permite crear tableros personalizados, su código es funcional, declarativo y contiene diversos componentes que dotan de funcionalidad a las aplicaciones. Además, utiliza la librería Plotly de Python, que contiene herramientas para la visualización de los datos permitiendo hacer gráficos más complejos y más interactivos que los que permite hacer otras librerías como Maplotlib.

Además permite crear nuevos componentes a través de React (librería Javascript para desarrollo de interfaces de usuario), mediante programación dinámica genera automáticamente clases de python a partir de React, las clases generadas serán el componente.

Para el estilo de las aplicaciones cuenta con la biblioteca dash-bootstrap-components que permite usar componentes Bootstrap en los tableros facilitando diseños adaptativos.

[2] Página oficial de Dash Plotly

1.4.3 Librerías

Se utilizarán diversas librerías entre las que se encuentran pandas, Matplotlib, Datetime, Numpy, Stlearn y Sklearn para la preparación de datos y la elaboración de los modelos realizados.

Sklearn [3]

Es una biblioteca de aprendizaje automático escrita en python y de código abierto. Sklearn cuenta con algoritmos de clasificación, regresión, clustering, reducción de dimensionalidad y preprocesamiento (como estandarización y normalización), así como métodos para comparar, validar y elegir parámetros para los modelos. Esta librería incluye Numpy, Pandas, Scipy, Maplotlib, Ipython y Sympy. Los algoritmos de clustering que incluye son K-Means, propagación de afinidad, Mean-Shift, clustering espectral, método Ward, clustering aglomerativo, DBSCAN, OPTICS, mezclas gausianas y Birch, e índices internos como el coeficiente de silueta, Calinski-Harabasz e índice Daves-Bouldin.

Tslearn [4]

Es un paquete de python para aprendizaje automático de series temporales que utiliza Sckit-learn, Numpy y Scipy. Contiene varios módulos incluidos entre otros, métricas para series temporales que incluye DTW y variantes, otro para clustering donde incluye K-means, uno para preprocesamiento que incluye representaciones para la serie temporal como PAA y SAX y otro paquete para algoritmos basados en Shapelets que requiere de Keras.

Pandas [5]

Es un paquete de Python que permite trabajar con datos estructurados proporcionando estructuras de datos rápidas, flexibles y expresivas, su estructura básica es el dataframe y contiene métodos para poder modificarlos.

[3] Guía de usuario y página oficial de la librería Sklearn
[4] Página oficial de la librería Tslearn
[5] Página oficial del paquete Pandas

Numpy [6]

Es un paquete que proporciona procesamiento de matriz de propósito general,es decir, proporciona un array multidimensional de alto rendimiento y métodos para manejarlos que permiten realizar cálculos de forma sencilla y que supone una alternativa a las listas en python.

Datetime [7]

Es un módulo que permite manejar y manipular fechas y que contiene diferentes tipos de fechas como time, date,datetime o timedelta.

Maplotlib [8]

Es una librería que permite realizar gráficos en 2D, contiene una gran diversidad de gráficos.

Ruptures[9]

Es una librería para la detección de puntos de cambio fuera de línea que incluye detección exacta y aproximada para modelos paramétricos y no paramétricos. Entre los métodos que incluye se encuentra Pelt.

MatrixProfile [10]

Proporciona algoritmos exactos y aproximados para calcular el perfil de matriz de una serie temporal así como para determinar discordias y motivos de dicha serie a partir de él y herramientas para visualizar los resultados.

[6] Página oficial del paquete Numpy
[7] Documentación del módulo Datetime
[8] Página oficial de la librería Maplotlib
[9] Guía de usuario de la librería Ruptures [58]
[10] dirección de la fundación matrix profile autora de los paquetes probados

Plotly [11]

Librería para visualización interactiva que posee una amplia variedad de gráficos avanzados.

1.4.4 Entorno de desarrollo

Se ha utilizado el entorno de trabajo interactivo Jupyter Notebook, tanto para realizar el análisis de datos, como el preprocesamiento y la elaboración de los modelos. Jupyter es una aplicación web de código abierto para programación interactiva y paralela que permite crear y compartir documentos con código (tanto en python como en otro lenguajes) , fórmulas matemáticas, gráficos y texto llano. La arquitectura es independiente del lenguaje, y consiste en protocolos de mensajería y clientes interactivos, estos clientes están conectados a núcleos que implementan las instalaciones de programación interactivas centrales.

El dashboard se ha realizado con el entorno de desarrollo Spyder, se trata de un entorno de desarrollo interactivo, integrado y multiplataforma de código abierto.

1.5 Organización del documento

Este documento se estructura en nueve capítulos sin contar el resumen, manuales de usuario y anexos que acompañan a la memoria. El contenido de cada capítulo se especifica a continuación.

- **Capítulo 1: Introducción** Se explica brevemente el trabajo realizado, la motivación para llevarlo a cabo, los objetivos que se pretendían conseguir, el alcance del sistema y el entorno de trabajo incluyendo lenguaje y librerías utilizadas entre otros. También se incluye la estructura de este documento.

- **Capítulo 2 Plan del proyecto** En este apartado se explica la metodología que se ha seguido en este trabajo así como la distribución temporal y el presupuesto.

[11] Página de la librería Plotly

- **Capítulo 3: Industria 4.0** Desarrolla el concepto de la Industria 4.0 pasando de explicar que es, sus características y el cambio que supone, así como las oportunidades que genera. Mencionando más en detalle sus aplicaciones, para a partir de los datos recopilados obtener diagnósticos y pronósticos sobre el estado de salud de las máquinas.

- **Capítulo 4: Mecanizado,Fresado y Máquinas CNC** Este apartado, explica las técnicas de mecanizado convencional centrándose en el fresado, las máquinas CNC y lo que suponen estas máquinas para la industria actual. De estas máquinas se explican los componentes en que se dividen, ventajas, desventajas que tienen y sus fallos más comunes .

- **Capítulo 5 Estado del Arte** Se comenta brevemente estudios realizados previamente sobre detección de anomalías y clustering de series temporales en entornos industriales.

- **Capítulo 6: Aprendizaje no supervisado** Presenta la primera parte de la base teórica del estudio. Habla brevemente del aprendizaje automático para detallar y profundizar en el aprendizaje no supervisado centrado en el clustering. Este capítulo se podría subdividir en 3 partes:

  - **Clustering genérico**: Definiciones y conceptos generales y se explica los tipos de algoritmos, así como las medidas utilizadas para evaluar los resultados.
  - **Clustering para series temporales**: Explica brevemente en qué consiste la minería de series temporales para centrarse en el clustering indicando las formas de aplicar clustering con este tipo de datos, así como las representaciones y medidas de similitud utilizadas para realizar este agrupamiento.
  - **Clustering de subsecuencias de series temporales** Relacionado con el clustering de series temporales un problema particular es el clustering de subsecuencias de series temporales. En este apartado se explica el problema de agrupar partes de una misma serie de tiempo indicando las dificultades que plantea así como algunas soluciones que se han aplicado. También se explican otros aspectos relacionados, como son la segmentación de la serie y el uso de una estructura de datos llamada perfil de matriz para el descubrimiento de patrones repetidos e inusuales dentro de la serie de tiempo.

- **Capítulo 7 Detección de Anomalías** Se explica en qué consiste la detección de

anomalías, indicando también cómo se aplica para series temporales y que técnicas y enfoques se utilizan.

- **Capítulo 8: Algoritmos de Clustering** Se explican en detalle los algoritmos de clustering probados con los datos de las fresadoras. De cada algoritmo, también se indica cómo aplicarlos mediante la librerías utilizadas.

- **Capítulo 9: Caso de estudio: Clustering aplicado a datos de fresado** Se aplican varios algoritmos de clustering en datos con diferentes características provenientes de la monitorización de dos fresadoras CNC distintas. De cada una de ellas se comparan los resultados de los distintos algoritmos y métricas utilizadas.

- **Capítulo 10: Caso de estudio: Dashboard** Incluye el análisis y modelado de la herramienta de visualización realizada para mostrar y consultar los resultados del análisis de las dos fresadoras.

- **Capítulo 12: Conclusiones** En la parte final del documento se resume el trabajo realizado y se comentan los resultados obtenidos indicando líneas de trabajo futuro.

Capítulo 2

Plan de Proyecto

El plan adoptado de trabajo ha intentado seguir de forma flexible la metodología CRISP-DM [14].

2.1 Metodología

CRISP-DM, es una metodología típica para la minería de datos, que ofrece un resumen del ciclo de vida de un problema de minería de datos, el cual se divide en 6 fases. La transición entre las fases no es estricta pudiendo retroceder y avanzar entre ellas en caso de ser necesario. Las fases son:

- **Comprensión del negocio**: Se establecen los objetivos y requisitos desde la perspectiva de negocio, de esta forma al extraer la máxima información se reducen riesgos identificando problemas,objetivos y recursos . Además se evalúa la situación (como el conocimiento del problema) , se establecen objetivos del proceso de minería de datos y se genera el plan de proyectos.
- **Compresión de los datos**: Consiste en entender los datos, teniendo en cuenta los objetivos de negocio que se han determinado. Esta etapa incluye la recopilación , descripción , exploración y verificación de la calidad de los datos.
- **Preparación de los datos**: Extraer los datos a los que se aplicaran las tareas de minería de datos. En este apartado se incluye la selección, limpieza, construcción, integración y formateo de datos.

- **Modelado**: Se aplican las técnicas de minería. En esta parte se incluye la selección de la técnica a aplicar, el diseño de la evaluación, la construcción del modelo y su evaluación.
- **Evaluación**: Se estudian los resultados obtenidos para ver si cumplen las necesidades de negocio que se habían establecido y se revisa el proceso estableciendo la siguiente línea de trabajo.
- **Despliegue**: Planificar el despliegue, realizar la planificación de la monitorización, generar el informe final y revisar el proyecto.

En la figura 2.1 se representa la transición entre las fases de CRISP-DM.

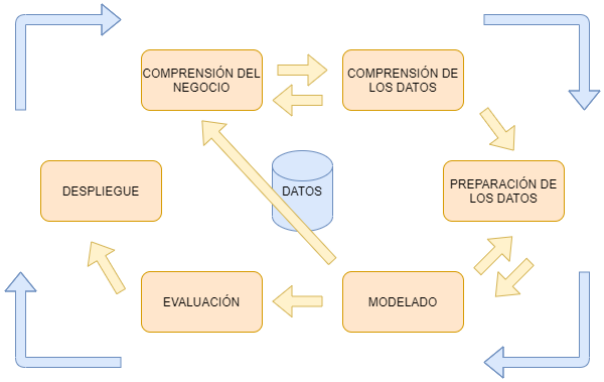


Figura 2.1: Metodología CRISP-DM

2.2 Fase de Trabajo y estimación Temporal

Resumiendo un poco, para la elaboración del proyecto se comenzó estudiando conceptos de mecanizado y comprendiendo cómo funciona una fresadora CNC, lo que condujo a comprender el contexto de la industria 4.0 y sus aplicaciones. Dentro de las aplicaciones, el trabajo se centró más en el apartado de pronósticos. Además se buscaron trabajos donde se aplicaron métodos de aprendizaje no supervisado en este tipo de máquinas para saber cómo actuar.

Tras la obtención de los datos, se pasó a su análisis inicial y al aprendizaje de técnicas de detección de anomalías y de clustering. La naturaleza de los datos y la forma de la señal llevó a dirigir el trabajo al aprendizaje del uso de estas técnicas con datos de series temporales.

Debido a un mayor interés de encontrar diferentes comportamientos de funcionamiento dentro de las series de tiempo diarias que de agrupar días enteros, (ya que el comportamiento es diferente cada día pudiendo realizar diferentes trabajos de mecanizado), el estudio se dirigió al clustering de subsecuencias. Esto condujo a buscar información sobre cómo lograr que el agrupamiento de subsecuencias obtenido fuese relevante y ampliar el estudio a otros conceptos relacionados con las subsecuencias. Entre estos conceptos se encuentran, los métodos segmentación y el descubrimiento de motivos u otros patrones relevantes de la serie de tiempo, así como formas de encontrar estos patrones, como el uso de una estructura de datos denominada perfil de matriz.

Este estudio se fue intercalando con pruebas para la representación, distancias y segmentación de los datos así como aprender y en algunos casos implementar los algoritmos de clustering para finalmente agrupar los datos de la fresadora. Ante las dudas surgidas de los datos obtenidos se analizaron y se aplicaron los mismos algoritmos de clustering a los datos de otra fresadora proporcionados por Kaggle [54].

Finalmente tras realizar el clustering de ambas fresadoras se pasó a realizar un dashboard para facilitar la comparación de resultados (obtenidos con distintos algoritmos y distancias).

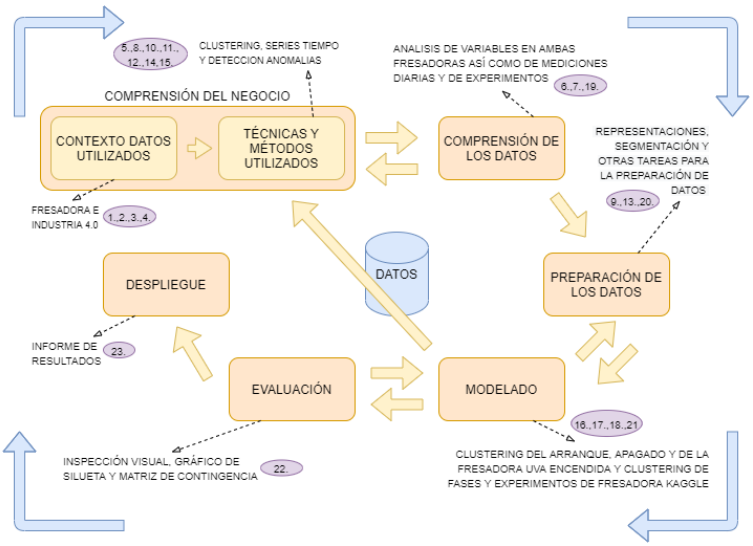


Figura 2.2: Adaptación al proyecto

Las fases de trabajo realizadas se pueden clasificar dentro de las fases de CRISP-DM. Dentro de la comprensión del negocio podemos dividir las fases de trabajo referentes a esta parte en dos grupos. En un primer grupo, las que se refieren a entender el contexto

de los datos de trabajo, es decir, comprender sobre fresadoras e industria y un segundo grupo, con las referentes a la investigación de las técnicas de minería de datos que se utilizarán o que se podrían llegar a utilizar.

Dentro del grupo de **comprensión del negocio - contexto de los datos utilizados** encontramos:

1. Entender el mecanizado, entender en qué consiste el fresado y que son las máquinas CNC y su importancia en la industria.

2. Investigar técnicas de aprendizaje automático y más concretamente de aprendizaje no supervisado que se han aplicado en trabajos anteriores para datos industriales, así como los diferentes objetivos que persiguen.

3. Comprender en qué consiste la industria 4.0 los cambios que supone, las dificultades a las que se enfrenta y las oportunidades que brinda. Se busca comprender el contexto en que se encuentran los trabajos investigados y la situación actual de la industria.

4. Aprender sobre los métodos de pronósticos que se emplean para determinar cuándo van a fallar las herramientas.

En el grupo de **comprensión del negocio - técnicas de minería de datos investigadas** encontramos:

5. Investigación de problemas de clustering, el tipo de algoritmos, las medidas de evaluación y otras consideraciones del clustering a tener en cuenta.

8. Clustering de series de tiempo investigación sobre cómo se aplica el clustering con datos de series temporales incluyendo componentes, enfoques, distancias y representaciones adoptadas.

10. Clustering de subsecuencias de series temporales, entender porque se han invalidado trabajos realizados para este problema y cómo conseguir resultados correctos.

11. Aprendizaje sobre métodos segmentación de series de tiempo.

12. Investigar sobre patrones de interés dentro de una serie de tiempo (motivos, discordias y shapelets) y cómo hallarlos (perfil de matriz).

14. Aprender cómo detectar anomalías identificando los tipos de anomalías existentes, los diferentes enfoques y cómo aplicar estas técnicas con datos de series temporales.

15. Aprendizaje de los algoritmos de clustering utilizados.

Dentro de la parte de **comprensión de los datos**, en el proyecto se han utilizado dos conjuntos de datos que han requerido de diferentes tareas para cada conjunto.

6. Análisis de las variables de los datos de la fresadora del CIDAUT, analizando cuales son las variables significativas, comprobando su comportamiento los días analizados y analizando este comportamiento tanto de forma independiente como comparándolo con el resto de variables (incluyendo el de las variables de monitorización con el valor booleano de la situación de la puerta de la máquina).

7. Análisis detallado del comportamiento diario de la corriente distinguiendo diferentes situaciones e incluyendo un análisis de las fases del arranque de la fresadora.

19. Analizar variables de la fresadora Kaggle diferentes fases, datos de mediciones y de los experimentos junto con sus resultados.

Tras el análisis de los datos se observó una serie de dificultades y de tareas de preparación necesarias para obtener datos que se puedan aplicar a los algoritmos de clustering en la etapa de modelado.

9. Comprobación de representaciones de los datos y obtención de un muestreo regular de la fresadora CIDAUT mediante muestreo PAA.

13. Realizar segmentación de los datos incluyendo varios intentos con la variable puerta, mediante aproximación lineal y a través de los valores estables de corriente (incluyendo segmentado específico para el arranque).

20. Preparación datos fresadora Kaggle, se eliminan variables y filas con mediciones no precisas o no relacionadas con la monitorización de la fresadora y se separan los datos de medición y resultados por muestras, fases y experimentos.

Dentro de la etapa de **modelado** se han realizado algoritmos de clustering y se han aplicado algoritmos a distintos conjuntos de datos de ambas fresadoras.

16. Clustering del arranque de la fresadora CIDAUT, al arranque completo, a segmentos, a puntos de tiempo y a representación en vector de características con métodos clásicos basados en densidad, partición y jerárquicos con distancias euclidiana y DTW y prueba de métodos implementados.

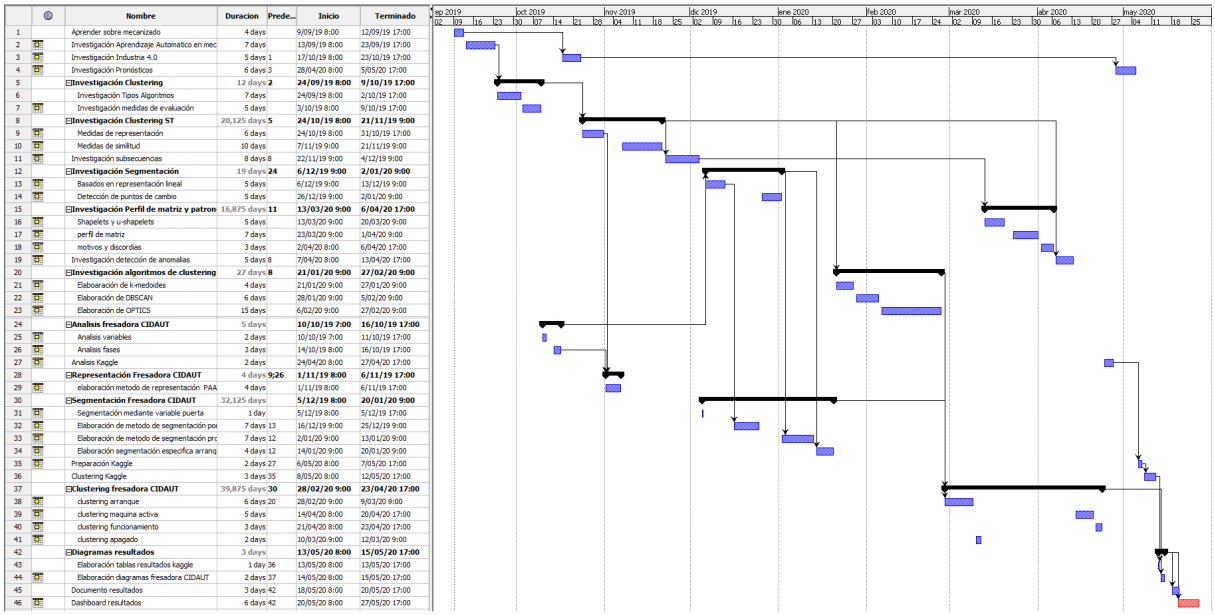
17. Clustering al apagado de la fresadora CIDAUT con los métodos K-Means, DBSCAN, OPTICS y jerárquico aglomerativo y con las distancias DTW y euclidiana.

18. Clustering a la fresadora CIDAUT encendida (considerando encendido completo y solo del funcionamiento) con los mismos algoritmos y distancias del apartado anterior.

21. Clustering por fases y por experimentos de la fresadora Kaggle.

En la parte de **evaluación** el clustering realizado sobre los datos de la fresadora CIDAUT se comprobaron mediante gráficos de las series de tiempo diarias distinguiendo los grupos formados, así como diagramas de los clusters obtenidos. En el caso de la fresadora Kaggle se usó la matriz de contingencia (22.), finalmente el **despliegue** ha consistido en el informe de la parte práctica de este trabajo (23.) y en el dashboard realizado (24.).

Las diferentes tareas se han ordenado según la realización temporal de las mismas, aunque algunas de ellas se han ido intercalando.



	Nombre	Duracion	Prede...	Inicio	Terminado
1	Aprender sobre mecanizado	4 days		9/09/19 8:00	12/09/19 17:00
2	Investigación Aprendizaje Automatico en mec	7 days		13/09/19 8:00	23/09/19 17:00
3	Investigación Industria 4.0	5 days	1	17/10/19 8:00	23/10/19 17:00
4	Investigación Pronósticos	6 days	3	28/04/20 8:00	5/05/20 17:00
5	**Investigación Clustering**	12 days	2	**24/09/19 8:00**	**9/10/19 17:00**
6	Investigación Tipos Algoritmos	7 days		24/09/19 8:00	2/10/19 17:00
7	Investigación medidas de evaluación	5 days		3/10/19 8:00	9/10/19 17:00
8	**Investigación Clustering ST**	20,125 days	5	**24/10/19 8:00**	**21/11/19 9:00**
9	Medidas de representación	6 days		24/10/19 8:00	31/10/19 17:00
10	Medidas de similitud	10 days		7/11/19 9:00	21/11/19 9:00
11	Investigación subsecuencias	8 days	8	22/11/19 9:00	4/12/19 9:00
12	**Investigación Segmentación**	19 days	24	**6/12/19 9:00**	**2/01/20 9:00**
13	Basados en representación lineal	5 days		6/12/19 9:00	13/12/19 9:00
14	Detección de puntos de cambio	5 days		26/12/19 9:00	2/01/20 9:00
15	**Investigación Perfil de matriz y patron**	16,875 days	11	**13/03/20 9:00**	**6/04/20 17:00**
16	Shapelets y u-shapelets	5 days		13/03/20 9:00	20/03/20 9:00
17	perfil de matriz	7 days		23/03/20 9:00	1/04/20 9:00
18	motivos y discordias	3 days		2/04/20 8:00	6/04/20 17:00
19	Investigación detección de anomalias	5 days	8	7/04/20 8:00	13/04/20 17:00
20	**Investigación algoritmos de clustering**	27 days	8	**21/01/20 9:00**	**27/02/20 9:00**
21	Elaboaración de k-medoides	4 days		21/01/20 9:00	27/01/20 9:00
22	Elaboración de DBSCAN	6 days		28/01/20 9:00	5/02/20 9:00
23	Elaboración de OPTICS	15 days		6/02/20 9:00	27/02/20 9:00
24	**Analisis fresadora CIDAUT**	5 days		**10/10/19 7:00**	**16/10/19 17:00**
25	Analisis variables	2 days		10/10/19 7:00	11/10/19 17:00
26	Analisis fases	3 days		14/10/19 8:00	16/10/19 17:00
27	Analisis Kaggle	2 days		24/04/20 8:00	27/04/20 17:00
28	**Representación Fresadora CIDAUT**	4 days	9,26	**1/11/19 8:00**	**6/11/19 17:00**
29	elaboración metodo de representación PAA	4 days		1/11/19 8:00	6/11/19 17:00
30	**Segmentación Fresadora CIDAUT**	32,125 days		**5/12/19 8:00**	**20/01/20 9:00**
31	Segmentación mediante variable puerta	1 day		5/12/19 8:00	5/12/19 17:00
32	Elaboración de metodo de segmentación poi	7 days	13	16/12/19 9:00	25/12/19 9:00
33	Elaboración de metodo de segmentación pr	7 days	12	2/01/20 9:00	13/01/20 9:00
34	Elaboración segmentación especifica arranq	4 days	12	14/01/20 9:00	20/01/20 9:00
35	Preparación Kaggle	2 days	27	6/05/20 8:00	7/05/20 17:00
36	Clustering Kaggle	3 days	35	8/05/20 8:00	12/05/20 17:00
37	**Clustering fresadora CIDAUT**	39,875 days	30	**28/02/20 9:00**	**23/04/20 17:00**
38	clustering arranque	6 days	20	28/02/20 9:00	9/03/20 9:00
39	clustering maquina activa	5 days		14/04/20 8:00	20/04/20 17:00
40	clustering funcionamiento	3 days		21/04/20 8:00	23/04/20 17:00
41	clustering apagado	2 days		10/03/20 9:00	12/03/20 9:00
42	**Diagramas resultados**	3 days		**13/05/20 8:00**	**15/05/20 17:00**
43	Elaboración tablas resultados kaggle	1 day	36	13/05/20 8:00	13/05/20 17:00
44	Elaboración diagramas fresadora CIDAUT	2 days	37	14/05/20 8:00	15/05/20 17:00
45	Documento resultados	3 days	42	18/05/20 8:00	20/05/20 17:00
46	Dashboard resultados	6 days	42	20/05/20 8:00	27/05/20 17:00

Figura 2.3: Gantt del proyecto

		Nombre	Duracion	Prede.
1		Aprender sobre mecanizado	4 days	
2		Investigación Aprendizaje Automatico en mec	7 days	
3		Investigación Industria 4.0	5 days	1
4		Investigación Pronósticos	6 days	3
5		**Investigación Clustering**	12 days	**2**
6		Investigación Tipos Algoritmos	7 days	
7		Investigación medidas de evaluación	5 days	
8		**Investigación Clustering ST**	20,125 days	**5**
9		Medidas de representación	6 days	
10		Medidas de similitud	10 days	
11		Investigación subsecuencias	8 days	8
12		**Investigación Segmentación**	19 days	**24**
13		Basados en representación lineal	5 days	
14		Detección de puntos de cambio	5 days	
15		**Investigación Perfil de matriz y patron**	16,875 days	**11**
16		Shapelets y u-shapelets	5 days	
17		perfil de matriz	7 days	
18		motivos y discordias	3 days	
19		Investigación detección de anomalias	5 days	8
20		**Investigación algoritmos de clustering**	27 days	**8**
21		Elaboaración de k-medoides	4 days	
22		Elaboración de DBSCAN	6 days	
23		Elaboración de OPTICS	15 days	
24		**Analisis fresadora CIDAUT**	5 days	
25		Analisis variables	2 days	
26		Analisis fases	3 days	
27		Analisis Kaggle	2 days	
28		**Representación Fresadora CIDAUT**	4 days	**9;26**
29		elaboración metodo de representación PAA	4 days	
30		**Segmentación Fresadora CIDAUT**	32,125 days	
31		Segmentación mediante variable puerta	1 day	
32		Elaboración de metodo de segmentación po	7 days	13
33		Elaboración de metodo de segmentación pr	7 days	12
34		Elaboración segmentación especifica arranq	4 days	12
35		Preparación Kaggle	2 days	27
36		Clustering Kaggle	3 days	35
37		**Clustering fresadora CIDAUT**	39,875 days	**30**
38		clustering arranque	6 days	20
39		clustering maquina activa	5 days	
40		clustering funcionamiento	3 days	
41		clustering apagado	2 days	
42		**Diagramas resultados**	3 days	
43		Elaboración tablas resultados kaggle	1 day	36
44		Elaboración diagramas fresadora CIDAUT	2 days	37
45		Documento resultados	3 days	42
46		Dashboard resultados	6 days	42

Figura 2.4: Tareas Gantt

La planificación que se plantearía para el proyecto se muestra a continuación, esta planificación no se ha ajustado finalmente a la elaboración del proyecto.

2.3 Estimación

Para la estimación se ha utilizado DMcomo (data mining cost model) [21, 50],una variante de COCOMO que fue introducida para proyectos de minería de datos. Al igual que COCOMO, se trata de un método paramétrico que utiliza ecuaciones analíticas y factores subjetivos para aproximar los meses por persona que requerirá el proyecto.

En DMcomo se utilizan una serie de factores de coste para realizar esta aproximación, los valores de estos factores se determinan mediante unas tablas que contienen las descripciones del esfuerzo.

Los factores utilizados se clasifican en:

- Datos: Factores relacionados con los datos que se vayan a utilizar, incluido la cantidad de datos (número de tablas, tuplas y atributos), los diferentes valores que pueden tomar los atributos, su calidad e información disponible sobre los modelos

de datos entre otros.

- Modelos: Referentes a los modelos que se van a aplicar y los datos que se pasarán a dichos modelos. Se incluyen tanto el número de modelos, como su tipo y características.
- Plataforma: relativos a la plataforma de desarrollo incluido el número de fuentes que se van a utilizar.
- Técnicas y Herramientas: Factores sobre las herramientas disponibles y las técnicas que incluyen dichas herramientas que pueden facilitar para la elaboración del proyecto.
- Proyecto: en este grupo se encuentra el factor referente a la documentación a entregar, también se incluirían otros factores sobre el número de departamentos de una organización que interfieren en el proyecto y la realización del proyecto en múltiples entornos y localizaciones.
- Personal: Referentes al equipo de trabajo del proyecto incluido la formación y experiencia.

Mediante los factores de estos grupos DMcomo presenta dos fórmulas para calcular el esfuerzo. Una fórmula que emplea 23 factores y que se puede emplear en proyectos bien definidos y otra con 8 factores como variables. Para realizar la estimación de este trabajo se utilizará la segunda fórmula.

$$\begin{aligned} MM8 = {} & 70{,}897 + 2{,}368 \cdot NTAB + 2{,}885 \cdot NATR + 4{,}792 \cdot DISP \\ & + 2{,}713 \cdot DEXT + 7{,}257 \cdot TMOD + 4{,}615 \cdot MATR \\ & - 3{,}842 \cdot NFOR - 3{,}275 \cdot MFAM \end{aligned} \tag{2.1}$$

Los factores que emplea esta fórmula pertenecen a las categorías de datos, modelos y técnicas . Dentro de la categoría de datos, se utiliza NTAB (número de tablas) que se establece a 0 al considerarse pocas tablas en el proyecto, NATR (número de atributos) establecido a 1 ya que los sensores de los que se recopilan datos son relativamente pocos respecto a otros dominios donde el número de variables puede ser considerablemente mayor, el valor de DISP (dispersión de los datos) se ha establecido como medio y se le asigna como valor 2 y el factor DEXT (necesidad de adquisición de datos externos) se ha establecido a su valor mínimo que es 2.

Dentro de la categoría de modelos utiliza, TMOD (tipo de modelo) aunque el valor asignado a los modelos de clustering es 2 se ha establecido a 3 al tratarse el problema de obtener un modelo descriptivo para patrones secuenciales y el MATR (número y tipo de atributos que se pasan al modelo) se establecerá también con el mínimo valor de la tabla que es 1.

Finalmente en los referentes a la técnica el NFOR (nivel de formación de los usuarios que requieren las herramientas) se establecerá a nivel medio 3, requiriendo conocer tanto las técnicas como la herramienta pero sin requerir un nivel experto en ambas, para terminar, para MFAM (familiaridad con el tipo de problema) el esfuerzo requerido se ha establecido como alto asignándole el valor de 4.

Con todo esto al aplicar la fórmula se estima el proyecto en 95 meses persona lo que lleva a concluir que esta estimación no es apta para proyectos tan pequeños.

Tras obtener una estimación fallida para el proyecto se ha intentado con el COCOMO tradicional, como se espera obtener una estimación medianamente precisa al desconocer todos los detalles y al tratarse de un proyecto pequeño con solo una persona se ha optado por el COCOMO básico orgánico.

El dashboard realizado es muy simple y contiene solo dos salidas (salida de la fresadora Kaggle y los días de la fresadora) y 4 consultas (consulta de resultados de cualquiera de las dos fresadoras y consulta de la segmentación y de los cluster de la fresadora uva). Al ser todas las salidas y consultas simples, se obtiene un total de 20 puntos de función no ajustados (PFNA).

Para ajustar los puntos de función solo se ha considerado el diseño para la eficiencia del usuario final, la complejidad del proceso lógico interno y la reusabilidad del código, todas con valor 1. Con ello y siguiendo la fórmula el factor de ajuste sería $FA = (0{,}01 \cdot \sum FC) + 0{,}65 = (0{,}01 \cdot (1+1+1)) + 0{,}65 = 0{,}68$. Una vez obtenido el factor de ajuste, obtenemos que los puntos de función ajustados son: $PF = PFNA \cdot FA = 20 \cdot 0{,}68 = 13{,}6$ puntos ajustados.

Como no se han encontrado las LDC/PF de python pero dado que en java es 53 y en los lenguajes de cuarta generación son 20, utilizaré 30 para python. Esto estima las líneas de código en 408.

Además de las líneas de código del dashboard se debería considerar aquellas requeridas para la preparación de los datos,los algoritmos realizados y los modelos aplicados.

Estimando todo ello en al menos 3 veces lo requerido para la aplicación, estimar las líneas de código en 1632. Al aplicar la fórmula del COCOMO orgánico básico (con parámetros a=2,4, b=1,05, c=2,5 y d=0,38), se obtiene que el esfuerzo requerido es de $Esfuerzo = a \cdot (KLDC)^b = 2{,}4 \cdot 1{,}632^{1{,}05} = 4{,}01$ personas-mes y que el tiempo estimado del proyecto es de $c \cdot Esfuerzo^d = 2{,}5 \cdot 4{,}01^{0{,}38} = 4{,}24$ meses.

Dando así una estimación más aproximada a lo que se espera que dure el proyecto. Teniendo en cuenta que esta estimación solo ha tenido en cuenta la parte de codificación y no la parte de estudio teórica creo que la estimación obtenida se acerca a lo que debería durar el proyecto.

Dado que el primer intento ha sido infructuoso se realizará una pequeña estimación similar a la anterior mediante puntos de caso de uso para comparar los resultados.

Teniendo un único usuario que interactúa con el tablero mediante interfaz gráfica se le asigna una complejidad alta teniendo por tanto $UAW = 1\ actor \cdot 3\ \frac{peso}{actor} = 3$.

Teniendo 8 casos de uso que se han establecido como 7 de ellos simples y uno de complejidad media obtengo que $UUCW = 10\ \frac{peso}{CU_{medio}} \cdot 1\ CU_{medio} + 5\ \frac{peso}{CU_{simple}} \cdot 7\ CU_{simple} = 45$. Finalmente los puntos de caso de uso sin ajustar serían $UUCP = UAW + UUCW = 3 + 45 = 48$.

Como factores de complejidad técnica se ha tenido en cuenta los objetivos de rendimiento,procesamiento complejo, código reutilizable y fácil de cambiar con influencia baja 1 y eficiencia respecto al usuario final y fácil utilización influencia media 2. El factor R_i se calcula multiplicando la influencia del factor por su peso, salvo la facilidad de utilización que tiene de peso 0,5 el resto tienen peso 1. Aplicando la fórmula se obtiene que el factor de complejidad técnica es $TDF = 0{,}6 + (0{,}01 \cdot \sum R_i) = 0{,}67$, estableciendo un factor de entorno cercano a 0,965 se obtiene finalmente que los puntos de caso de uso ajustados $UCP = UUCP \cdot TCF \cdot EF = 48 \cdot 0{,}67 \cdot 0{,}965 = 31{,}03$ puntos de caso de uso.

Usando Karner como factor de productividad y estimando 20 horas persona por UCP, obtengo que $Esfuerzo = 31{,}03\ UCP \cdot 20\ \frac{horas}{UCP} = 620{,}7\ horas$. Este tiempo se corresponde con el tiempo de codificación que se estima que es solo el 40 % el tiempo total de elaboración, sin embargo, para este dashboard las pruebas y la sobrecarga es mínima por lo que el porcentaje sería mayor por eso he considerado un 50 %. Considerando que las 620,7 horas se corresponden con el 50 %, el tiempo total estimado es de 1241,4 horas. Estableciendo 8 horas al día 6 días a la semana se obtiene un periodo de trabajo de 6,47 meses.

De igual modo que en la estimación conseguida mediante COCOMO, teniendo en cuenta que en esta estimación no se considera el estudio teórico, esta estimación aunque más cercana al resultado real es mayor a lo que debería durar el proyecto.

De las estimaciones realizadas creo que la más adecuada es la obtenida con el COCOMO orgánico básico ya que además de considerar más el conjunto del proyecto (no solo el dashboard sino también la preparación de los datos y el clustering) se obtiene una estimación bastante cercana a lo que debería durar el proyecto teniendo en cuenta el incremento que supondrá el estudio de la base teórica para realizarlo.

2.4 Presupuesto

Para el trabajo no se ha requerido invertir dinero en programas software ya que se han utilizado programas de código abierto o planes de uso personal, que no requerían de inversión. Esto deja como gasto el portátil empleado para trabajar estimando aproximadamente un coste de 700 euros para amortizar en 6 años en los cuales se ha utilizado para este proyecto finalmente alrededor de 18 meses se obtiene un gasto de $\frac{700 \text{ euros}}{6 \text{ años}} \cdot 1{,}5 \text{ años} = 175 \text{ euros}$.

Parte I

Comprensión del problema

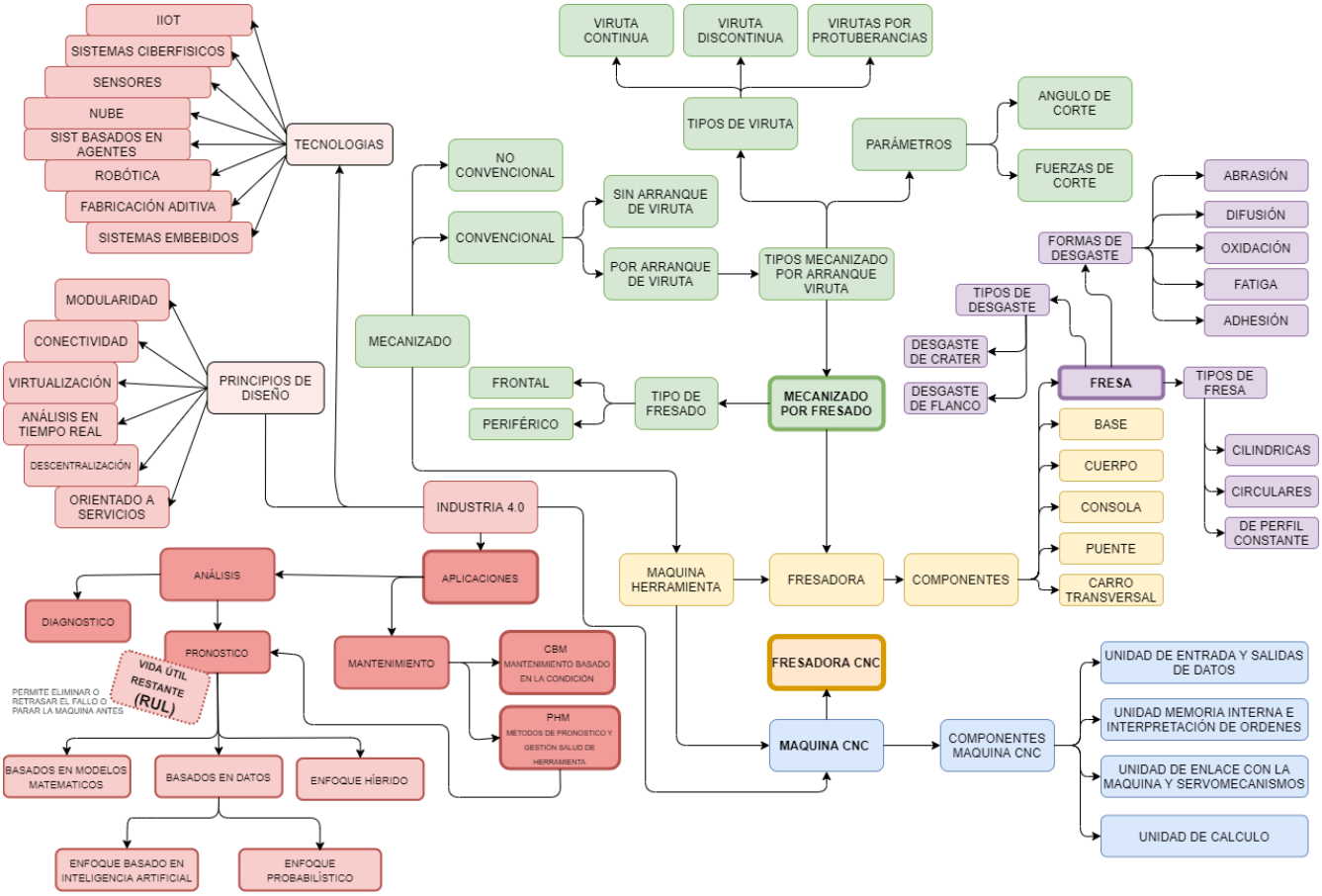


Figura 2.5: Mapa guía para la comprensión del entorno del proyecto

Contiene los conceptos que se explicaran en los capítulos de esta parte del trabajo. Incluye conceptos sobre la Industria 4.0 desde tecnologías implicadas, principios de diseño y aplicaciones, entre las que se encuentran los pronósticos (en rojo), sobre el mecanizado centrado en el fresado (verde) y sobre las maquinas que realizan este mecanizado donde se encuentran las maquinas CNC (amarillo, azul y verde). Finalmente se abordará el estado del arte revisando trabajos donde se aplicaron datos de mecanizado entre otros, para la detección de anomalías.

Capítulo 3

Industria 4.0

La Industria 4.0 surge como concepto en 2011 en Alemania, como parte de una ruta estratégica de fabricación para promover su digitalización. La Industria 4.0 consiste en la **automatización e integración de las TIC en todos los niveles de producción industrial**. Tiene como objetivo transformar los procesos de fabricación y aumentar la productividad, logrando así **fábricas inteligentes, autónomas y descentralizadas** capaces de ofrecer una producción más personalizada [51] .

Busca conseguir una integración a distintos niveles:

- **Integración horizontal** en toda la red de creación de valor (entre empresas intercambiando información y recursos en tiempo real). Esto ofrece como ventajas la combinación de recursos y la disminución de los riesgos así como una mejor adaptación de los cambios y una colaboración más activa con los clientes.
- **Integración vertical** integración en diferentes niveles de la organización tanto a nivel de producción como de gestión.
- **Integración extremo a extremo** integración en todo el ciclo de vida del producto.

Con todos estos cambios se pretende satisfacer las necesidades individuales de los clientes, logrando una mayor flexibilidad de la fabricación industrial para lograr que sea más productiva, con una mayor calidad y personalizada al cliente.

En la anterior revolución, gracias a la aplicación en la industria de los avances en los campos de la electrónica y la informática que se produjeron en el siglo XX, se logró

la automatización de procesos. Esta automatización supuso un incremento de los datos provenientes de estos procesos de fabricación. Ahora en esta nueva revolución se persigue la **digitalización** de la industria, de forma que se aprovechen estos datos, para mejorar la productividad y la calidad de los productos.

3.1 Características de la Industria 4.0

Existen varias tecnologías de fabricación avanzadas que se centran en diferentes conceptos relacionados con la Industria 4.0 [71]:

- **Fabricación Inteligente**: Centrada en la interacción entre los recursos de producción. Los sistemas de fabricación inteligente (IMS) usan una arquitectura orientada a servicios a través de internet para ofrecer servicios personalizados.
- **Fabricación Habilitada por IoT**: Se enfoca en los datos de los procesos de producción y de decisión. En ella las máquinas y otros recursos necesarios para la producción se transforman en objetos de fabricación inteligentes (SMO) que son capaces de recopilar datos, de interactuar y conectarse entre ellos para intercambiar información en tiempo real. Consiguiendo así no solo la automatización de la producción sino también conseguir que sea más adaptable.
- **Fabricación en la Nube**: Persigue modelar y configurar los servicios para tener un sistema de producción en paralelo, capaz de gestionar los recursos de producción del ciclo completo del producto a través de los conceptos de programación en la nube, el internet de las cosas, la virtualización y de tecnologías orientadas a servicios.

Para lograr estas fabricaciones, esta nueva etapa industrial se basa en la colaboración de una serie de tecnologías como la **programación en la nube**, el **análisis big data**, el **internet de las cosas** y los **sistemas ciberfísicos (CPS)**.

Estos **sistemas ciberfísicos CPS**, son mecanismos que interaccionan entre el entorno físico y virtual. Se encargan de integrar, controlar y coordinar simultáneamente procesos y operaciones y son capaces de procesar y dar acceso a los datos que utilizan, es decir, son sistemas integrados que intercambian datos en red y permiten la producción inteligente. A diferencia de los sistemas embebidos, los CPS tienen interacciones en red con salidas y entradas físicas.

Como un apartado del IoT donde se busca conseguir que los objetos puedan recopilar información, interactuar con su entorno e interconectarse con otros objetos a través del intercambio de datos, surge el concepto de **industrial internet of things** IIoT. En el IIoT los dispositivos requieren de más conexiones y más datos, son dispositivos más resistentes y autónomos, que necesitan recibir continuamente información de otros dispositivos. Esta necesidad de información se ha visto beneficiada por el aumento de redes de sensores y sistemas integrados. A través del IoT aplicado a la industria se persigue tanto optimizar procesos, como mejorar el uso de recursos y conseguir crear sistemas complejos.

Con ello se definen seis principios de diseño para la Industria 4.0:

- **Interoperabilidad**: Capacidad de integración y cooperación entre máquinas. Se intercambian y utilizan la información que comparten. Este término está relacionado con el IIoT.

- **Virtualización**: Poder realizar simulaciones y modelos a partir de la virtualización de los procesos monitorizados por los sistemas ciberfísicos CPS.

- **Descentralización**: Capacidad de tomar decisiones en un entorno distribuido.

- **Análisis en Tiempo Real**: Poder recoger y analizar datos para poder detectar problemas de producción de forma rápida.

- **Orientación al servicio**: Los CPS, las fábricas y también las personas se utilizan en el contexto de una arquitectura orientada a servicios (SOA) para facilitar la toma de decisiones.

- **Modularidad**: Facilidad para actualizar la fabrica añadiendo nuevos módulos y CPS sin necesidad de cambios en los módulos ya existentes.

Estas tecnologías y conceptos permiten conseguir procesos de fabricación adaptables a las necesidades individuales. Con todos los sensores y dispositivos con RFID (identificación por radiofrecuencia) se generan un gran número de datos, que se pueden utilizar para diagnóstico y pronóstico permitiendo así dar soporte y mejorar la toma de decisiones estratégicas y operativas.

3.2 Desafíos de la Industria 4.0

Sin embargo y a pesar de las ventajas y avances que nos supone la Industria 4.0 también presenta una serie de desafíos [32]. Los datos de monitoreo provenientes principalmente de los sensores, actuadores y del PLC son complejos y heterogéneos. Además estos datos no solo se deben recopilar sino también preprocesar y transmitir entre los CPS. Estos datos de diversos tipos se deben integrar en una sola plataforma que debe ser escalable y rápida. Esto implica que se necesite una comunicación en tiempo real entre las máquinas remotas y los CPS . Además de esto, las aplicaciones CPS son complejas y heterogéneas e integrar varios subsistemas diferentes es costoso.

Los sistemas actuales no están preparados y las máquinas de control numérico por computadora (CNC) se deben modernizar. Es necesario que estas máquinas interactúen con el resto de recursos de manera efectiva para realizar las tareas, a pesar de que estos dispositivos tengan diferentes tecnologías (diferentes protocolos de comunicación, sistemas de control, componentes eléctricos y mecánicos ...). Esta adaptación supone una inversión inicial.

La UE se fijó como objetivo en 2016 aumentar el peso de la industria en el PIB europeo del 15,3 % al 20 % en 2020 lo que requiere aumentar su productividad ofreciendo una alta calidad a un coste reducido [61], sin embargo, hoy en día gran parte de la producción final (aproximadamente el 12 %) se invierte en las piezas de corte y sus reemplazos [37] . En el sector de la fabricación se espera que esta digitalización suponga un crecimiento entre el 20 % y 30 % para 2030 [51].

3.3 Aplicaciones de la Industria 4.0

Como se comentó los fallos en las herramientas producen un importante gasto para la industria, no solo por el dinero de la reparación y/o sustitución de la pieza sino también por el tiempo que la máquina está inactiva sin producir y también por los recursos humanos que se emplean para identificar anomalías, fallos y para su reparación. Se ha comprobado que antes del fallo de la máquina su comportamiento generalmente cambia , presentando un comportamiento inusual antes del fallo [70]. Debido a la creciente utilización de las máquinas CNC y gracias a sus sensores podemos utilizar los datos de la herramienta para poder detectar anomalías y predecir cuándo va a fallar una herramienta (el tiempo

de vida útil restante). Detectar cuando va a fallar una pieza tiene un gran beneficio para la productividad ya que nos permitirá saber cuando sustituirla evitando sustituirla demasiado pronto lo que implicaría más gastos en sustituciones o demasiado tarde y que se pueda producir una rotura.

Esto permite eliminar retrasos en la producción debido a fallos en las máquinas que pueden llevar a tener que detener la producción, generando un desperdicio de materia prima o un mal funcionamiento del sistema. Los datos recopilados no solo se utilizan para la detección de estos fallos sino también para poder estimar y predecir la vida útil restante de la máquina (RUL) reduciendo tiempo y costes de operaciones debidos a mantenimiento innecesario.

Para poder determinar el desgaste que conduce al fallo de la herramienta se supone que esta degradación de los componentes físicos es monótona y no reversible. Como el desgaste o degradación de la pieza es difícil de medir y en ocasiones no es directamente observable, se utiliza una variable aleatoria para medirla, conocida como RUL.

Los datos recopilados de la monitorización permiten extraer conocimiento significativo para la industria. Se pueden realizar distintos tipos de análisis [17]:

- **Descriptivo**: Donde se analizan los datos recopilados para identificar tendencias o patrones. Se trata de obtener información de los datos sin supervisión. Este tipo de análisis permite detectar puntos de interés y conseguir detectar fallos y comprobar el estado de la herramienta.
- **Predictivo**: Analiza cual es probablemente el comportamiento futuro a través de modelos estadísticos y técnicas de pronóstico. En la industria puede ayudar a estimar la vida útil de las herramientas, como la probabilidad de ocurrencia de fallos aumenta al inicio y fin de la vida útil de la herramienta conocer esta estimación, permite regular de forma más adecuada los cambios de herramienta y el mantenimiento.
- **Prescriptivo** permite recomendar acciones y sus posibles resultados mediante algoritmos de simulación y optimización.Pudiendo así la máquina gestionar sus tareas de mantenimiento.

Sin embargo estas aplicaciones se enfrentan a una serie de complicaciones. En muchos casos es difícil interpretar los datos recopilados así como usarlos para identificar el grado de degradación. Además se suele recopilar datos de múltiples sensores los cuales no siempre

son relevantes para medir el desgaste y se pueden dejar de medir otros más relevantes. Por si esto no fuera poco se suelen requerir de una gran cantidad de datos para poder apreciar la degradación del sistema.

3.3.1 Diagnóstico y Pronóstico

Normalmente se desconoce cómo afecta el funcionamiento de la máquina a su estado y como debido a su uso se degrada con el tiempo, por eso se utilizan métodos de diagnóstico y pronóstico para determinar su condición y evolución.

El diagnóstico es el proceso de identificación y determinación de la relación entre los fallos y sus causas. En los pronóstico se utilizan los resultados del diagnóstico para predecir el comportamiento futuro.

Las tareas de diagnóstico se dividen en detección, aislamiento e identificación de fallos .

Pronósticos

Los métodos de pronóstico de fallo son una predicción del aumento de fallos a partir de unas condiciones operativas. Se basan en intentar determinar el tiempo que falta para que se produzca un error y estimar la vida útil restante (RUL) de la herramienta de corte. Junto al valor de RUL se deben estimar unos límites de confianza.

La predicción de la vida útil restante permite aplicar medidas para mejorar la productividad eliminando el origen del posible fallo, retrasarlo o deteniendo la máquina y cambiando la herramienta de corte antes de que se rompa o deje de ser útil.

Un programa de pronósticos se compone de 4 partes (figura 3.1) :

1. **Adquisición de Datos**: Consiste en la captura y almacenamiento de los datos. Los datos recopilados se pueden dividir en:
    - Datos de eventos: Indican fallos, cambios de piezas, mantenimientos, fases de descanso, ...
    - Datos de monitorización: Datos sobre la condición del sistema y que están relacionados con el estado del mismo como valores de corriente, vibración, temperatura, potencia, ...

2. **Construcción del indicador de salud (HI)**: Los datos de monitorización contienen mucha información así como ruido por lo que se suelen extraer indicadores que permitan mejorar la precisión de la predicción. Los indicadores de salud se pueden clasificar en:

   - **Indicadores físicos (PHI)** Contienen significado físico y se obtiene directamente de los datos de monitorización utilizando métodos estadísticos o métodos de procesamiento de señales. El PHI más utilizado es el RMS (valor cuadrático medio) pero se han aplicado otros como la curtosis o variables relacionadas con la entropía o con las transformadas en frecuencia de los datos.
   - **Indicadores de salud virtuales (VHI)** Se obtienen de la fusión de varios indicadores físicos. Se puede aplicar PCA u otros métodos de reducción de dimensionalidad así como SOM (mapas autoorganizados) o la distancia de mahalanobis entre otros.

   Estos indicadores deben ser monótonos ya que la degradación de la maquinaria es irreversible, robustos ante el ruido y a la variación de las condiciones de funcionamiento y tener una tendencia de degradación suave. Si se distingue entre varias etapas de salud (HS) un indicador adecuado debe ser capaz identificar cada uno de ellos.

3. **División Etapa de Salud (HS)**: Consiste en dividir los procesos de degradación continua de la herramienta en diferentes estados según las tendencias variables de los HI. El enfoque más simple es utilizar dos etapas una cuando la herramienta es utilizable y la otra cuando está desgastada .

4. **Predicción de RUL**: Se estima la vida útil restante (RUL) junto con sus límites de confianza (debido a la incertidumbre del proceso de degradación).

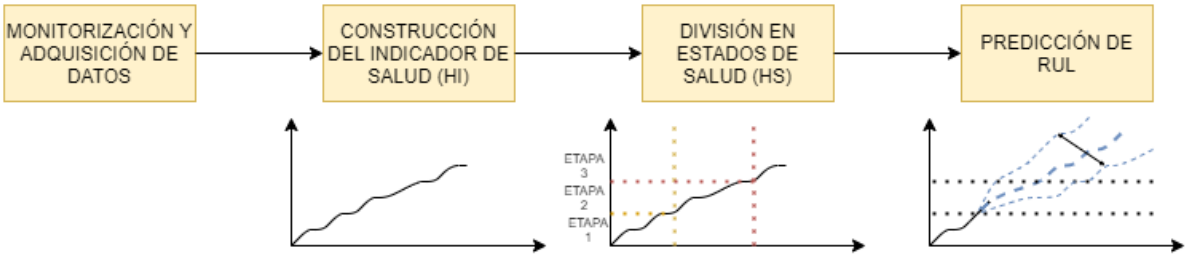


Figura 3.1: Fases de un programa de pronósticos

Existen varios métodos para determinar RUL que se clasifican en:

- **Pronóstico basado en modelos matemáticos**: Se establece un modelo matemático que permite explicar la degradación de la herramienta. Estos métodos relacionan matemáticamente las características físicas del sistema con los modos de fallo. Para ello se basan en la mecánica y la dinámica, no requieren de datos históricos y son más precisos. Sin embargo, son difíciles de aplicar debido a su dificultad para construirlos ya que requieren de conocimiento del dominio como las especificaciones de los componentes que no siempre son conocidas. Este método permite identificar modos de fallo más relevantes.

- **Pronóstico basado en datos**: Utiliza los datos de monitorización para construir el modelo de comportamiento y evalúa el estado actual comparándolo con el modelo aprendido. Se requiere entrenar el modelo con las condiciones adecuadas para que pueda detectar fallos. Este método requiere de menos suposiciones que el basado en la física y no requiere de conocimientos sobre cómo se produce el desgaste. Se utiliza cuando se disponen de suficientes datos de monitorización y es difícil o no es confiable aplicar un método de pronóstico basado en modelos, sin embargo carece de interpretación física. Dependiendo del método empleado se distingue en:

  - **Pronóstico basado en Estadística**: Se utilizan los datos para construir un modelo probabilístico. Consisten en determinar la distribución estadística que mejor se ajusta a los datos recopilados para obtener la estimación de la vida útil. Algunas de las técnicas empleadas en este tipo de pronósticos son máquinas de vectores de soporte, procesos gaussianos o modelos ocultos de Markov.
  - **Pronóstico basado en inteligencia artificial**: Se utilizan para sistemas complejos donde es difícil relacionar los procesos de degradación con modelos físicos o probabilísticos. Para este tipo de pronósticos se emplean redes neuronales y lógica difusa.

- **Pronóstico basado en enfoque híbrido**: Combina varios tipos de pronósticos, según los enfoques que combinen podemos encontrar:

  - Basado en experiencia y en datos, combina conocimiento experto para mejorar el modelo basado en datos. Este enfoque no es recomendable para predicciones a largo plazo.
  - Basado en experiencia y en la física, combina conocimiento experto para mejorar el modelo basado en la física.

- Varios modelos basados en datos, este tipo de pronósticos permite usar un modelo para estimar el estado de salud y otro para predecir la vida residual.
- Basado en datos y en la física de esta forma consigue mejorar el rendimiento obtenido con estos enfoques por separado.
- Basado en experiencia, en datos y en la física, este enfoque es difícil de aplicar por la dificultad de combinar los resultados de los diferentes pronósticos.

El enfoque de pronóstico más adecuado dependerá del caso de estudio específico, no existiendo un enfoque que funcione mejor que el resto para cualquier problema.

Entre algunas de las dificultades que presentan estos métodos, se encuentra que para obtener los pronósticos se requieren datos de todo el proceso de degradación hasta que la herramienta se rompe o falla. Esto complica la obtención de estos programas ya que el proceso de degradación es un proceso largo que requiere de muchos datos y las máquinas rara vez se dejan funcionar hasta que la herramienta se rompe. Además los datos de monitorización suelen ser ruidosos y no solo contienen datos de funcionamiento, contienen datos de arranque, apagados, descansos y otros periodos de inactividad que presentan comportamientos distintos. A todo esto se le añade la dificultad para medir el grado de desgaste necesario para realizar los pronósticos, debido a que no suele ser lineal al funcionamiento, tiempo de uso y otras condiciones del entorno. Además en algunos componentes es difícil de observar esta degradación sin destruir la pieza o los cambios que se producen en ella son a escalas tan pequeñas que requieren de instrumentos específicos para observarlas. Además aunque se pueda no siempre se puede detener la máquina para poder observar este deterioro.

3.3.2 CBM y PHM

El almacenamiento, procesamiento y transmisión de grandes volúmenes de datos permite que las máquinas puedan autoevaluar su estado actual y así mejorar su rendimiento y gestionar su mantenimiento.

Todos estos métodos y técnicas han permitido cambiar el mantenimiento tradicional de las herramientas basados en gestión de fallos o en intervalos fijos por otros enfoques preventivos. En el mantenimiento basado en la condición (condition based maintenance, CBM) se utilizan los datos de monitorización para evaluar el estado de la máquina y gestionar las tareas de mantenimiento. Mientras CBM permite actuar cuando es necesario a través

del diagnóstico, la gestión de salud y pronósticos (prognostics and health management, PHM) administra la salud de la máquina previniendo cuando serán necesarias tareas de mantenimiento y cuándo se producirá un fallo a través de pronósticos. Una vez determinada la vida útil restante (RUL) se pueden tomar decisiones que pueden ser intentar eliminar el origen del fallo, retrasar o detener la máquina antes de que se produzca.

PHM junta los pronósticos con la gestión de salud. La gestión de salud consiste en el proceso continuo de acciones de mantenimiento adecuadas que se toman en base al diagnóstico y a los pronósticos y que intentan evaluar y minimizar el impacto de los fallos. De esta forma PHM intenta optimizar las actividades de mantenimiento durante todo el ciclo de vida del recurso, en este caso de la máquina CNC. Aunque se suele usar con monitorización basada en condición, PHM se puede usar con otros tipos como la monitorización de la salud estructural del sistema.

En un modelo PHM se tiene por tanto que estimar el estado actual de la máquina, predecir la evolución del estado hasta el fallo y determinar el impacto del fallo en el rendimiento. Se considera que el modelo PHM es eficiente cuando es capaz de identificar y predecir fallos y de elaborar un calendario de mantenimiento y de gestión de recursos.

Con todo ello PHM ofrece un marco que permite gestionar la salud de la máquina con soluciones integrales e individualizadas, consiguiendo así aportar una visión integrada del estado de la máquina .

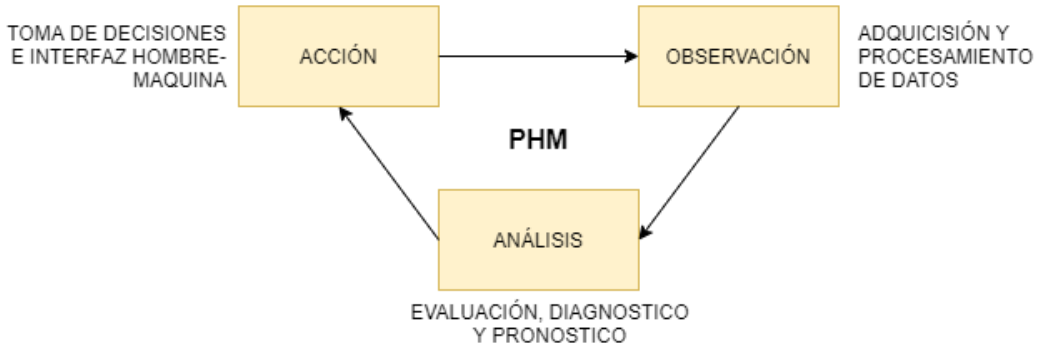


Figura 3.2: Ciclo de mantenimiento PHM

El enfoque PHM se puede dividir en tres partes (figura 3.2) : **observación** donde se recopilan y procesan los datos, **análisis** donde se evalúa la degradación y se realiza el diagnóstico y el pronóstico y una etapa de **acción** que incluye el soporte de decisiones que puede ser fuera de línea para acciones de mantenimiento o en línea configurando el sistema y la interfaz de usuario que permita visualizar los datos, el estado y controlar el

mantenimiento.

Normalmente se usan las metodologías PHM a nivel de componentes pero también se puede usar a nivel de sistemas.

Capítulo 4

Mecanizado, Fresado y Máquinas CNC

Dentro de los procesos de fabricación, el **mecanizado** sigue siendo actualmente uno de los procesos de producción más utilizados.

Las máquinas-herramientas son herramientas capaces de realizar operaciones de mecanizado y que utilizan en su uso habitual una fuente de energía distinta del movimiento humano, normalmente energía eléctrica . Estas máquinas disponen de una plataforma cinemática capaz de dotarla de movimiento a lo largo de los ejes necesarios para realizar el proceso de mecanizado concreto que realice la máquina, así como de proporcionar rigidez para soportar las fuerzas involucradas en el proceso, amortiguamiento capaz de absorber las vibraciones generadas y precisión en cada uno de los ejes.

Dependiendo del tipo de proceso y otras características de la fabricación como el tamaño de las piezas, el volumen de producción o la complejidad de las piezas hay una amplia variedad de máquinas-herramientas. La evolución de estas máquinas ha llevado a los centros de mecanizado que se utilizan actualmente.

4.1 Mecanizado

El mecanizado es el proceso de fabricación por el cual una materia prima se transforma en una pieza con una forma y tamaño específico mediante una serie de operaciones. Existen distintos tipos de mecanizado:

- Mecanizado convencional, que se basa en la rotación para la fabricación de la pieza.

- Mecanizado no convencional, que aplica métodos no mecánicos como productos químicos o electricidad para obtener la pieza.

Dentro del mecanizado convencional podemos distinguir dos tipos en función de si la pieza se obtiene mediante el arranque de viruta o sin él.

- Los procesos de **mecanizado sin arranque de viruta**, tienen como objetivo deformar la materia prima mediante procesos físicos para obtener la forma deseada. Entre los procesos de mecanizado sin arranque de viruta encontramos entre otros el prensado, laminado y trefilado.
- En los procesos de **mecanizado por arranque de viruta** permiten obtener distintas piezas a través del uso de herramientas de corte, los movimientos de la herramienta de corte podrán ser de rotación y translación entre otros dependiendo de la máquina y del tipo de proceso utilizado. Entre los procesos de mecanizado por arranque de viruta se encuentran el fresado, torneado y taladrado.

4.1.1 Mecanizado por Arranque de viruta

El mecanizado por arranque de viruta es de los más utilizados debido a su facilidad de automatización, alta precisión y la diversidad de formas que permite realizar, además de requerir poco tiempo de preparación. Sin embargo tiene una serie de inconvenientes, entre los que se encuentra el desecho de gran parte del material, un gran consumo de energía durante el proceso, tiempos y costes de producción elevados y tamaños de las piezas limitado por el tamaño de la máquina utilizada para realizar el mecanizado.

Dependiendo de la velocidad de corte o del tipo de material sobre el que se realice este mecanizado podemos distinguir tres tipos de viruta:

- **Viruta Discontinua** se trata de pequeñas fracturas de la materia prima y se produce en materiales frágiles o con materiales dúctiles a baja velocidad de corte.
- **Viruta con Protuberancias** se produce cuando existe una elevada fricción y se produce una adhesión que hace que la viruta se deslice sobre el material adherido en vez de sobre la herramienta lo que puede llevar a desprendimientos en la pieza que produzcan un acabado deficiente, este tipo de viruta se da con materiales de elevada ductilidad a velocidades de corte bajas.

- **Viruta Continua** es la que se obtiene en un proceso de mecanizado normal y permite obtener buenos resultados con un correcto acabado.

Entre los parámetros de este mecanizado que afectan al acabado son los ángulos y las fuerzas de corte.

Los **ángulo de corte** afectan al resultado del mecanizado así como al desgaste de la herramienta de corte. Un gran ángulo de desprendimiento de la viruta implica una disminución de las fuerzas de corte y un mayor desgaste en la herramienta debido al aumento de la fricción y de la velocidad relativa de la viruta sobre la superficie de la herramienta de corte. En el caso del ángulo de incidencia, un ángulo grande puede aumentar el riesgo de fractura de la herramienta de corte, mientras que uno pequeño aumenta su desgaste debido al rozamiento. Otros ángulos son el ángulo de inclinación que determina la dirección de la viruta y el ángulo de posición que sirve para controlar fuerzas de impacto, sobre todo en el inicio del corte y para aprovechar el filo de la herramienta.

Las **fuerzas de corte** afectan al calentamiento de la máquina y de la pieza que se fabrica, al desgaste de la herramienta de corte y a la calidad de la pieza entre otros. Estas fuerzas se producen por la presión ejercida entre la herramienta y el material con el que se trabaja.

4.2 Fresado

El **fresado** es una forma de mecanizado convencional que se realiza por arranque de viruta mediante una herramienta rotativa con forma circular y de cortes múltiples conocida como fresa; esta herramienta puede tener diversas formas lo que permite realizar diferentes acabados. Las máquinas que usualmente realizan este tipo de mecanizado se conocen como fresadoras. El fresado consiste en una operación de corte interrumpido, ya que los dientes de la fresa entran y salen durante cada revolución. El movimiento principal en el fresado es el que se consigue mediante el giro de la fresa sobre su propio eje y el movimiento de avance se produce por el desplazamiento de la fresa, la profundidad del corte se debe a la aproximación de la fresa a la materia prima con la que se trabaje.

Este tipo de mecanizado permite obtener una gran diversidad de piezas, pudiendo realizar cualquier geometría, con bastante precisión y con diferentes materiales. Además se puede utilizar tanto para obtener una única pieza como para fabricar en serie lo que flexibiliza la producción. Sin embargo es un proceso caro, donde no se puede utilizar para cual-

quier material, no permitiendo trabajar con materiales duros. Debido a su versatilidad, el fresado es uno de los mecanizados más usados.

Existen dos tipos básicos de fresado según la posición de la fresa respecto a la superficie con la que se está trabajando:

- **Fresado Periférico**: cuando el eje de la fresa es paralelo a la superficie y el mecanizado se realiza con el borde de la fresa. Permite realizar planeados, ranuras y escalonados en las piezas. En función de la dirección de giro se habla de fresado en concordancia (o convencional), si la dirección de avance de la pieza y de la fresa es la misma (gira con el avance), o en oposición (o contraposición), si la dirección de giro de la fresa es contraria a la del avance (gira contra el avance), según el tipo de fresado empleado la presión de corte y la formación de viruta varia.

- **Fresado Frontal**: cuando el eje de la fresa es perpendicular a la superficie, al igual que en el periférico se utiliza los extremos de la fresa para realizar el mecanizado, se habla de fresado convencional cuando la fresa es más grande que la pieza y si solo sobrepasa el ancho de la pieza en un lado se trata de fresado parcial. Con este fresado podemos realizar perfilados, ranuras, fresado de cajeras (con cavidades de profundidad limitada en piezas plana y contorneado.

El movimiento de desplazamiento, tamaño y forma de la fresa son los que determinan la profundidad y anchura del corte teniendo también que tener en consideración la velocidad de mecanizado y el material a transformar. Los movimientos que se realizan en el fresado se pueden dividir en dos tipos el movimiento de avance que generalmente posee 3 grados de libertad y el movimiento de corte o movimiento principal que consiste en el giro de la herramienta de corte y es un movimiento que alcanza mayor velocidad y potencia que el de avance. Ambos movimientos pueden ser lineales o circulares.

Entre las operaciones más comunes que permite este mecanizado se encuentra el planeado, escuadrado, fresado de perfiles y fresado de ranuras.

4.2.1 Fresadora

Las máquinas que realizan un mecanizado por fresado se conocen como fresadoras, estas máquinas surgieron ante la necesidad de crear máquinas herramientas capaces de fabricar

piezas intercambiables en el armamento, donde la falta de uniformidad exigía piezas de distintos tamaños, materiales y tolerancias.

Las fresadoras tienen el eje de rotación de la herramienta de corte perpendicular a la dirección de avance. A diferencia de otras máquinas como el prensado, las fresadoras permiten mayor variedad de piezas debido a que permite movimientos a lo largo de los ejes X,Y,Z pudiendo realizar movimientos de corte verticales, longitudinales y transversales. Esto permite realizar con precisión una o varias superficies mecanizadas en la misma pieza, esta versatilidad hace que las fresadoras sean una de las máquinas de mecanizado más utilizadas.

Las fresadoras se pueden clasificar en verticales, horizontales y universales (estas últimas permiten hacer cortes helicoidales, para lo que cuentan con un componente llamado divisor que recibe el movimiento del husillo y la inclinación la consigue mediante una mesa orientable o con un eje orientable). También se pueden clasificar por la forma de dar la profundidad del corte según esto pueden ser de consola (si la consola y la mesa se aproximan a la fresa), de bancada (si es la fresa la que se aproxima a la consola y a la mesa) o mixta.

Su funcionamiento consiste en un motor de accionamiento que transmite a la cadena cinemática de transmisión que lleva potencia al cabezal de la máquina que es el encargado de realizar el movimiento principal (el de rotación) alrededor del husillo o eje principal que se encuentra en el eje Z (eje vertical) , los ejes X (eje longitudinal) e Y (eje transversal) son los que se encuentran en paralelo y en perpendicular a la pieza de trabajo, según el tipo de fresadora la mesa de trabajo puede desplazarse por los ejes X y Z en una fresadora horizontal o por los ejes X e Y en una fresadora vertical.

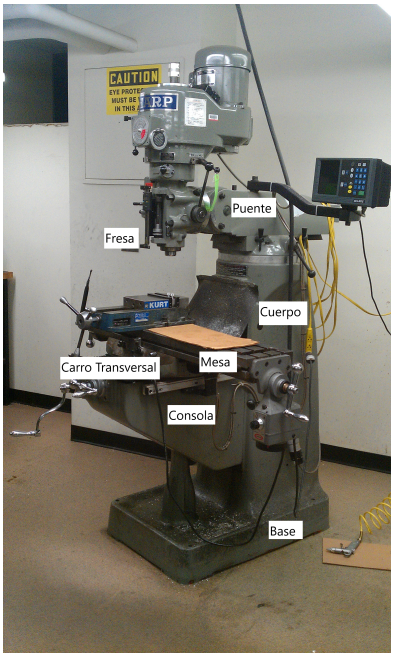


Figura 4.1: Fresadora componentes

Los componentes principales de estas máquinas son (figura 4.1): La **base** que permite dar rigidez a la estructura y fijar la máquina al suelo, un **cuerpo** o columna que contiene unas guías

en su lateral por donde se mueve la consola y que en su parte superior se aloja el puente de la máquina, un **puente** (o brazo) que sujeta y sirve de guía al husillo donde se sitúa la herramienta giratoria de corte que se denomina **fresa**, una **consola** que es el carro vertical que está acoplado a las guías verticales de la columna (en caso de ser una fresadora de consola se desplaza para realizar el mecanizado), el **carro transversal** que se desplaza por las guías de la parte superior de la consola y una **mesa** que es una plataforma de fundición aleada con forma rectangular que puede realizar movimientos horizontales y que es donde se coloca la pieza que se quiere transformar.

En el caso de la fresadora universal existe otro componente, la placa giratoria que es la que posibilita poder girar la mesa.

La precisión de las piezas que se fabriquen depende en gran medida de la máquina, algunos de los parámetros de la fresadora que pueden afectar a la calidad del resultado final son la precisión de giro del husillo, así como su resistencia a las vibraciones y su rigidez.

Al igual que ocurre en otras máquinas las características del entorno influyen en su estado.

4.2.2 Herramientas de corte: Fresa

En el mecanizado por arranque de viruta se utiliza una herramienta de corte para eliminar material de la materia prima y obtener la pieza deseada, en el fresado la herramienta que se utiliza se conoce como fresa.

Estas herramientas de corte requieren poder soportar grandes esfuerzos mecánicos y disponer de una alta resistencia al desgaste y a la fractura y poder soportar las elevadas temperaturas que se alcanzan en el mecanizado. Las fuerzas a las que esté sometida la herramienta varían dependiendo del proceso de mecanizado que se realice. El material de la herramienta de corte debe ser más duro que el del material que trabaje, por ello algunos de los materiales más utilizados para su elaboración son el acero rápido o diferentes tipos de metal duro que se clasifican en 6 grupos según el material que pueden mecanizar. Para la elección del material se debe considerar la relación entre dureza y tenacidad, por ello muchas veces se utiliza una combinación de materiales que permita conseguir una mejor relación de estas características. Se utiliza un material base que de tenacidad y se recubre la superficie que está en contacto con la herramienta y que es la más expuesta al rozamiento y a altas temperaturas con un material más duro, esto además permite proteger la herramienta y reducir el rozamiento. Este recubrimiento suele estar hecho de

algún compuesto basado en nitruro de titanio (TiN).

Las herramientas de corte poseen de rompevirutas en la cara de desprendimiento y consisten en resaltes para forzar la rotura de la viruta continua así como facilitar la evacuación de viruta en la zona de corte, su diseño permite marcar la zona óptima de trabajo

Las fresas son herramientas de corte que están formadas por una superficie que puede ser plana, cilíndrica o cónica y que disponen en su periferia o cara frontal de una serie de cuchillas. Las fresas pueden estar fabricadas por distintos materiales y pueden tener diversas formas y tamaños.

Debido a su diversidad existen multitud de clasificaciones diferentes según el método de fresado, el tipo de superficie o el perfil de incidencia de corte, la fijación de la fresa en la máquina, la forma de sus dientes o la dirección de corte de la fresa.

Algunos de los tipos más comunes de fresas son las **fresas cilíndricas**, que poseen un mango cilíndrico y permiten realizar cortes planos a una buena velocidad de corte; **fresas circulares** que son discos de acero rápido con dientes que les permiten cortar de forma frontal y lateral simultáneamente (este tipo alcanza mayor velocidad y permite generar un mayor volumen de viruta) o las **fresas de perfil constante** con dientes tallados que dejan la forma en el material cuando lo cortan.

Desgaste en las herramientas de corte

Todas las fresas sufren desgaste con el paso del tiempo y se deben cambiar cuando se produce. Este desgaste es debido principalmente al rozamiento, a las elevadas temperaturas y a la afinidad química entre la herramienta de corte y el material de la pieza. La fricción entre la pieza a transformar y la herramienta de corte produce una energía que aumenta la temperatura tanto de la pieza como de la herramienta. Este incremento de la temperatura disminuye su resistencia y favorece el desgaste de la herramienta de corte . Todas las herramientas de corte dejan de ser útiles cuando su filo sufre de desgaste y el tiempo que tarda en producirse depende de muchos factores como son los materiales utilizados o las condiciones y parámetros de corte por lo que es difícil determinar el tiempo útil de una herramienta. Por todo ello es tan importante poder encontrar una buena relación entre la productividad y el tiempo de vida útil de la herramienta.

Este desgaste se puede deber a varios fenómenos como: la **abrasión**, pérdida de material

debido al contacto entre las superficies ; **difusión**, parte del material de la pieza se mezcla con la herramienta debido a altas temperaturas; **oxidación**, se produce generalmente cuando la pieza y la herramienta se relacionan durante el proceso de arranque debido a la presión y a las elevadas temperaturas; **fatiga**, los materiales se vuelven frágiles y acaban rompiéndose esto se puede deber a materiales inadecuados, altas temperaturas o mala refrigeración de la herramienta; y **adhesión**, adición de parte de la viruta en la herramienta.

El desgaste de la herramienta no es homogéneo y se pueden diferenciar distintos tipos de desgaste: **desgaste de flanco**, en la superficie de incidencia debido a la abrasión; **desgaste de cráter**, en la superficie de desprendimiento debido a la afinidad química; entallamiento en el filo principal; filo recrecido por adhesión; deformación plástica (causa deformaciones); o fisuras térmicas (grietas debidas al calor).

Este desgaste se puede notar entre otros motivos por: un incremento en la potencia requerida para poder arrancar la viruta durante el mecanizado (debido a un incremento del esfuerzo); por un mayor incremento del calor (producido por un aumento de la fricción que puede deberse a una herramienta deformada), por ruidos agudos (debido a un incremento en las vibraciones), por peores acabados de las piezas a realizar o por cambios, astillamiento o rotura del filo de corte de la herramienta. En el caso de trabajar con aceros la formación de rebabas (parte de material que sobresale en los bordes o en la superficie) durante el arranque.

4.3 Máquinas CNC

El control numérico surge con el objetivo de automatizar el proceso de fabricación y de homogeneizar los resultados. Son un método capaz de controlar con precisión los movimientos de una máquina herramienta a través de instrucciones codificadas a un lenguaje entendible por la unidad de control de la máquina. Estas máquinas trabajan con coordenadas para controlar las posiciones. Utilizando otra definición un control numérico es cualquier dispositivo capaz de dirigir el posicionamiento en diferentes planos de un dispositivo mecánico móvil.

En estas máquinas las instrucciones se convierten en corriente eléctrica que los motores transforman en los movimientos de la máquina. Las máquinas de control numérico permiten garantizar la exactitud de las piezas que fabrican mediante dispositivos de medición

y registros.

Las primeras máquinas por control numérico (CN) utilizaban cintas magnéticas o tarjetas perforadas para programar las instrucciones. Con el tiempo y los avances en tecnología, la programación de estas máquinas se realiza por ordenador dando paso a las máquinas por control numérico por computador que se utilizan actualmente. Esto facilita su programación y mejora su flexibilidad a la hora de modificar el proceso de fabricación.

Las máquinas CNC se componen del sistema CNC y la máquina-herramienta. Un sistema con control numérico por computador o CNC es un sistema de fabricación capaz de realizar automáticamente una serie de operaciones siguiendo operaciones numéricas que le establece un ordenador. Es decir, una CNC es una máquina que trabaja automáticamente mediante un programa de control y que actualmente puede trabajar con todo tipo de máquinas-herramientas incluyendo fresadoras. Con el CNC, la máquina es controlada mediante comandos operacionales que le permiten funcionar con velocidad, eficiencia, precisión y la dotan de capacidad de repetición.

En las máquinas CNC la posición y velocidad de los motores que accionan los ejes se controlan desde un ordenador, que es capaz de mover la máquina por los distintos ejes al mismo tiempo permitiendo realizar una gran variedad de trayectorias. Dicho de otro modo el ordenador controla los movimientos de la mesa de trabajo, del carro de la máquina y del husillo. En el caso de la fresadora se controlan los movimientos en los ejes X,Y y Z. El control numérico por computadora es capaz de combinar el movimiento simultáneo de varios ejes lo que se conoce como interpolación de los ejes para lo cual requiere resolver un problema cinemático inverso (a partir de la trayectoria obtener el movimiento de cada eje) y controlar en cada instante la posición de cada uno de los ejes.

Las máquinas CNC contienen: un programa de instrucciones generado por software de CAD/CAM o manualmente, que puede estar en varios lenguajes dependiendo del controlador; y una unidad de control de la máquina (MCU) que se encarga de leer e interpretar las instrucciones, calcular las interpolaciones, controlar la cinemática, automatizar datos tecnológicos y analizar la información recibida a través de sus sensores. Los dispositivos CNC permiten dirigir la posición en diferentes planos de un aparato mecánico.

Desde su inicio se han producido avances en la programación de estas máquinas siendo ahora más fáciles y más rápidas de programar.

Las CNC controlan la máquina y analizan la información recibida algo que las antiguas

máquinas de control numérico (CN) que seguían una serie de instrucciones obtenidas de una cinta magnética o perforada no permitían.

Las CNC son muy comunes y se usan en muchos de los procesos de fabricación debido a una serie de ventajas:

- Reducen tiempos permitiendo programar los pasos de fabricación y poderlos repetir.
- Aumentan la variedad de operaciones.
- Permiten mayor flexibilidad del mecanizado.
- Reducen el tiempo de programación, anteriormente realizado con cintas magnéticas o tarjetas perforadas.
- Aumentan el control sobre las máquinas y su precisión, mejorando así el rendimiento.
- Reducen el coste de las herramientas.
- Optimizan el uso de materia prima reduciendo las pérdidas.
- Aumentan la seguridad.

En resumen permiten obtener una buena precisión, reducir el tiempo que la máquina está inactiva, hacer modificaciones sin necesidad de parar la máquina, obtener piezas iguales y mecanizar grandes series lo que produce un incremento de la productividad.

Algunos de los retos de la industria son la exigencia de precisión con rapidez y con diseños complejos que hacen necesario minimizar los errores. Para estos retos, las ventajas mencionadas hacen que estas máquinas sean muy adecuadas. Sin embargo, a pesar de estas ventajas, presenta una serie de inconvenientes como un incremento en el mantenimiento eléctrico, una elevada inversión inicial, un mayor coste por hora de operación y son máquinas más voluminosas que requieren de un mayor espacio, además requieren personal cualificado .

Para que su funcionamiento sea correcto se deben considerar una serie de factores que influyen en los resultados, como son, el refrigerante utilizado, la geometría de la pieza, el material que se utilice en la fabricación o la máquina-herramienta que se utilice y sus características (como su rigidez mecánica y térmica o su estabilidad, la cual, afecta a su capacidad de precisión). De estos factores dependen los parámetros con los que se realice

el mecanizado como son la velocidad de giro, el avance o la velocidad y profundidad del corte.

Además de realizar tareas relacionadas con la lógica de control, estas máquinas realizan otras tareas como la recolección de datos estadísticos.

Las máquinas CNC se pueden aplicar a distintos métodos de fabricación como son el fresado, torneado, taladro entre otros.

Las máquinas-herramientas que poseen esta estructura (figura 4.2), poseen una unidad central encargada de leer e interpretar las instrucciones, calcular los movimientos y traducirlos y enviárselos a la máquina, además poseen lazos de control, sensores, contactores, electroválvulas y de un PLC que automatiza funciones que no tienen que ver con el movimiento de la máquina como son alarmas, cambios de herramienta o aperturas y cierres de puertas entre otros. El PLC puede estar incorporado a la unidad central.

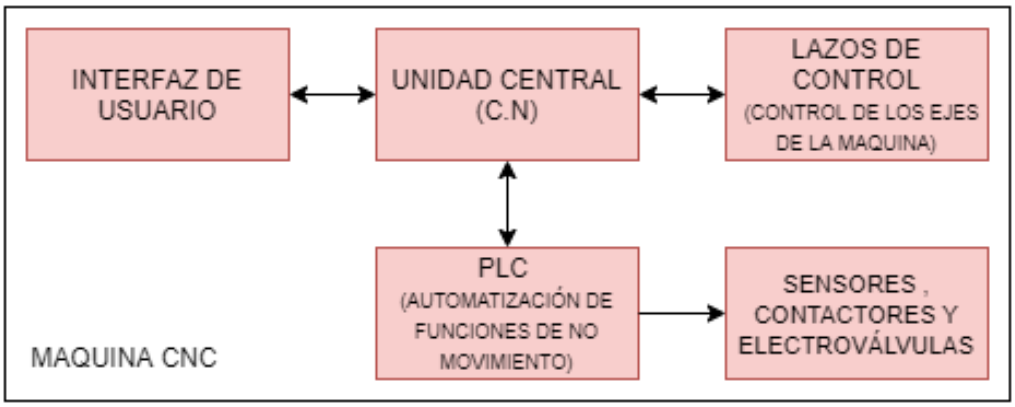


Figura 4.2: Componentes de una máquina CNC

Los componentes principales de la unidad central del control numérico son (figura 4.3):

- **Unidad de entrada y salida de datos** que se encarga de cargar el programa de mecanizado en el lenguaje adecuado al equipo de control numérico.
- **Unidad de Memoria interna e interpretación de órdenes** los programas se almacenan en memoria no volátil y se encarga de leer de forma secuencial las órdenes de ejecución para el trabajo de mecanizado.
- **Unidad de Cálculo** crea las órdenes que se encargaran de manejar la máquina-herramienta, el control lee el número de órdenes para completar el ciclo de trabajo, las órdenes se interpretan y se configuran las trayectorias de los ejes x,y,z.

- **Unidad de enlace con la máquina y servomecanismos** que permiten controlar los motores de la máquina-herramienta

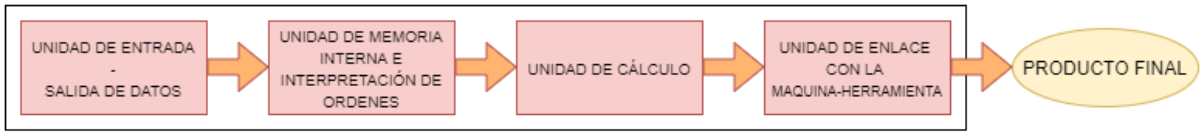


Figura 4.3: Componentes de la unidad de control de un sistema CNC

Las máquinas CNC cuentan con varios lazos de control, uno por cada eje pueden ser cerrados (con retroalimentación) o abiertos (sin retroalimentación) y se encargan de realizar las instrucciones que reciben de la unidad central y controlar su realización.

Estas máquinas tienen tres tipos de controles para situar y manejar la herramienta de corte y el material de trabajo pudiendo ser: punto a punto, controla la posición de la herramienta; paraxial, controla la posición y la trayectoria en los ejes X e Y; y trayectoria continua, donde se controlan los movimientos.

En la fabricación por ordenador se utilizan sistemas CAM. Se trata de un software que permite generar de forma automática programas CNC, es decir, pueden controlar una máquina CNC y son capaces de obtener el programa de mecanizado en dos pasos, primero obteniendo la posición relativa de la herramienta respecto a la pieza y en un segundo paso volver a procesar los datos para adaptarlo a las características concreta de la máquina CNC que se utilice. Estos sistemas se utilizan para la elaboración de piezas complejas.

Las órdenes que se pasan a la máquina siguen una estructura de bloque, donde cada bloque consta de un número de bloque (o programa), el código de la orden que indica la función de maquinado que se realiza (consta de un registro de palabras), los parámetros como las coordenadas x,y,z y otros comentarios.

En una **fresadora CNC**, desde un ordenador se puede controlar el movimiento de la mesa, el carro y el husillo. En el caso de la fresadora CNC se deben tener una serie de consideraciones a la hora de programarla, entre los que se incluye el tiempo de ciclo corto, los costes de la pieza y la producción así como tener en cuenta el modo de fresado empleado en la fabricación, la forma y material de la fresa empleada, los tipos de corte que se realizan y la capacidad de la máquina.

Para integrar estas máquinas con los conceptos de la industria 4.0 se deben reacondicionar los sistemas actuales, ya que un parque industrial tecnológicamente obsoleto puede reducir

su competitividad. Trabajos como el de [37] presentan nuevos paradigmas que permiten combinar las máquinas CNC con los conceptos del IIoT para actualizar los sistemas de mecanizado y adaptarse a la nueva industria 4.0.

En [37] proponen un método para el reacondicionamiento de máquinas CNC teniendo en cuenta requisitos del cliente, requisitos funcionales, parámetros de diseño, modelo de datos y la arquitectura del sistema. Para ello realiza una adaptación siguiendo 3 conceptos el diseño: axiomático, basado en las necesidades del cliente; el modelado de datos, para mostrar la información del sistema; y la arquitectura del sistema, de acuerdo a los principios de la industria 4.0.

Se debe conseguir un diseño axiomático que defina requisitos a partir de las necesidades del cliente y que permita un diseño del sistema personalizado. Para conseguirlo proponen clasificar las características y equipos del sistema en familias de componentes (características de las máquinas, dispositivos auxiliares, proceso, seguimiento de piezas y recursos humanos). Una vez clasificados se identifican los recursos disponibles y los recursos del cliente, definiendo restricciones de diseño que estén de acuerdo con la industria 4.0 y que permitan reducir la complejidad del diseño. Para terminar de acuerdo a ellos, se obtienen los correspondientes requisitos funcionales y patrones de diseño junto con las variables del proceso (que serán utilizadas en el modelado de datos y en la arquitectura del sistema).

El modelado de los datos debe permitir la integración del sistema de acuerdo a las necesidades del cliente y ser más personalizable. En el modelado se utiliza la clasificación de familias de componentes realizada para representar digitalmente los dispositivos del sistema, junto con las relaciones entre ellos y lograr una correlación lógica entre dispositivos.

De esta forma esta metodología es técnicamente factible y con bajo coste y permite actualizar los sistemas de mecanizado para adaptarlos a la industria 4.0.

Capítulo 5

Estado del Arte

Los datos de monitorización son recopilaciones de las mediciones de los sensores de las máquinas durante su tiempo de funcionamiento, se tratan de datos temporales. Para poder detectar patrones o anomalías se requiere considerar la dimensión temporal de estos datos.

Los artículos [70, 36, 1, 40, 25, 57, 45, 10, 8, 55] encontrados que utilizan este tipo de datos se emplean principalmente para la detección de roturas y anomalías, para la predicción de RUL, la optimización de la trayectoria y para detección de intrusos en redes industriales.

Entre los estudios que se centraron en el uso de los datos recopilados para la mejora de la precisión encontramos [45], donde se revisaron varios trabajos. Comentan que los estudios recientes se enfocan en la minimización del tiempo de mecanizado, la reducción de coste, del recorrido de la herramienta y en mejorar la eficiencia del mecanizado. Entre los métodos que se utilizan para lograr estos objetivos se encuentra el uso de algoritmos genéticos para planificar la trayectoria y algoritmos de enjambre, como la optimización de la colonia de hormigas y la optimización de partículas de enjambre, para la optimización de los parámetros de corte. Finalmente recalca la importancia de reducir el tiempo de mecanizado, siendo el objetivo principal en la mayoría de estudios, para tal fin recomienda el uso de métodos híbridos.

En los artículos donde se emplean para la detección de anomalías y predicción de RUL nos encontramos [36] donde destacaron una serie de dificultades para poder detectar la rotura como la dificultad para extraer características efectivas que puedan reflejar el desgaste de la herramienta y la elevada precisión requerida, resaltando la necesidad de un enfoque robusto con gran capacidad de aprendizaje. Tanto en su trabajo como en

otros se demostraba con bastante éxito la capacidad de la corriente para la detección de comportamientos inusuales debido a su relación con las fuerzas de corte involucradas en el mecanizado.

Además de la corriente, otras de las características que se han usado anteriormente para la detección de la rotura incluyen las señales acústicas, las fuerzas de corte y las señales de vibración recogidas mediante acelerómetros.

En [36] se utilizó una red neuronal convolucional para predecir la rotura de la herramienta, logrando una precisión del 93 %. Obtiene así un éxito mayor que el que obtiene mediante una red neuronal BP (backpropagation), que logró un rendimiento del 80 % mediante los datos de corriente del husillo expresados en el dominio del tiempo. Además de la precisión obtenida, se detectó un comportamiento similar en todas las herramientas, con un incremento de la corriente en el momento previo a la rotura y un cambio brusco en la rotura.

En [40] utilizaron la transformada discreta de wavelet junto con técnicas estadísticas (valores medios aritméticos de las asimetrías y un umbral de configuración) para detectar valores inusuales que se puedan deber a un fallo. Se examinaron 4 características: la fuerza de corte, vibración, emisión acústica y corriente del motor, de las cuales descartó la vibración, debido a su sensibilidad al ruido y las emisiones acústicas, por un elevado número de falsos positivos . Sin embargo existen varios artículos donde se obtienen buenos resultados, utilizando como variable la aceleración.

Aunque se han repasado brevemente algunos trabajos donde se aplican redes dinámicas bayesianas o redes neuronales, este documento se centra en la aplicación de métodos de clustering en datos temporales provenientes de un dominio industrial, concretamente el de una fresadora CNC.

En algunos métodos se usa el clustering como un paso intermedio para la predicción de RUL como en el artículo [57]. En este artículo se utiliza el algoritmo K-Means para identificar a qué etapa de desgaste de la herramienta (estado de salud) corresponde la medición; una vez identificada se utiliza para entrenar el modelo de desgaste. Este artículo utiliza cada uno de estos clusters para estimar los parámetros de una mezcla de modelos ocultos de Markov gaussianas correspondiente a cada una de las etapas, para finalmente estimar RUL.

En [10] compararon los algoritmos de clustering K-Means, jerárquico, DBSCAN y SOM

en aplicaciones industriales. Observaron que SOM era el que mejores resultados obtiene, pero que su rendimiento empeora con el aumento del número de clusters. Para grandes datos hallaron que el algoritmo que mejor funciona es K-Means, sin embargo, K-Means, jerárquico y SOM son sensibles al ruido.

En [70] destacan la utilidad de aplicar técnicas de aprendizaje no supervisado para la detección de anomalías, debido a la diversidad de piezas, materiales y otras características de la fabricación. Comprueban la aplicabilidad de las mediciones recopiladas de la máquina para detectar dichas anomalías, así como la importancia de prevenir los falsos negativos. En este artículo se utilizan datos con múltiples variables que contienen, entre otros, datos de las posiciones de cada uno de los ejes de la herramienta, velocidades, el estado de la máquina y un contador de piezas fabricadas. Las dos últimas variables permiten filtrar solo mediciones relativas al funcionamiento habitual de la máquina, y el contador divide esas mediciones en componentes individuales que denomina firmas.

El método que proponen consiste en obtener las firmas y aplicar a cada una un análisis de componentes principales (PCA) para posteriormente usar una representación basada en características y poder compararlas. El uso de una representación basada en características se debe a que los datos en bruto son demasiado volátiles para compararlas de manera eficaz. La representación que utiliza se compone de 7 características entropía, autocorrelación, cambio de nivel, cambio de varianza, curvatura (coeficiente del polinomio de segundo grado al que se aproxime la firma) ,picos (varianza de residuos) y puntos planos. Para terminar a esta representación se vuelven a aplicar PCA y finalmente se usa el algoritmo DBSCAN para detectar las anomalías.

La representación en características de series tiempo también se aplicó en [25], donde presentaron el método TS3C para agrupar series de tiempo teniendo en cuenta las secuencias de la serie de tiempo, para lo que utiliza dos etapas.

En una primera etapa, segmenta cada serie de tiempo usando un algoritmo de ventana creciente y representa cada segmento obtenido con un vector de características formado por los coeficientes de la representación lineal del segmento junto con diferentes parámetros (varianza, asimetría y coeficiente de correlación). Una vez obtenidos los vectores de cada segmento de la serie, estos se agrupan en k grupos mediante un algoritmo jerárquico y enlace Ward.

En la segunda etapa, se utilizan los resultados de esta primera agrupación para representar las series de tiempo y agruparlas. Para ello representa cada serie mediante los centroides

de los k clusters obtenidos junto con el elemento con mayor varianza de dicho cluster, el número de segmentos y la diferencia del error cuadrático medio entre el segmento más similar y el menos similar al centroide. Finalmente, se aplica clustering jerárquico aglomerativo a esta representación para agrupar las series de tiempo.

El uso de una representación basada en características para el clustering de series de tiempo también se utilizó en [64] logrando reducir considerablemente con la dimensión de los datos. El método que proponen aplica un método de búsqueda hacia delante para seleccionar las características más adecuadas para representar el conjunto de series.Para ello considera varias variables como la periodicidad, la correlación, la estructura autoregresiva no lineal, la asimetría o la kurtosis.

En [55] proponen un método de detección de anomalías en series de tiempo mediante segmentación de la serie a través de un método de puntos extremos importantes y el posterior agrupamiento de dichos segmentos mediante el algoritmo I-leader que realiza un clustering incremental. Para esta segmentación, se considera que un punto x es un mínimo/máximo importante dado un valor positivo R, si existe un valor a la izquierda y otro a la derecha de x de forma que $dist(x_{izq}, x) \geq R$ y $dist(x, x_{der}) \geq R$ en caso de mínimo y $dist(x_{izq}, x) \leq R$ y $dist(x, x_{der}) \leq R$ en caso de máximo. Los segmentos obtenidos se deben transformar a segmentos de igual longitud antes de aplicar el algoritmo .

Hay que mencionar que el clustering de series de tiempo completas puede no ser siempre eficaz para los datos de monitorización. Aunque en artículos como [70] trabajan con datos que cuentan con campos que permiten filtrar las horas de trabajo y dividir la señal en intervalos correspondientes a períodos de fabricación similares, no siempre se puede. Los intervalos de inactividad pueden variar y no siempre se disponen de campos o de información sobre los datos para poder identificarlos. Además las máquinas de fabricación como las fresadoras CNC permiten realizar distintas piezas con distintos materiales y que tendrán distintos datos de monitorización. Esto puede requerir la aplicación de métodos de clustering de subsecuencias (algunos de estos métodos se explicaran en el capítulo 7).

Relacionado con el clustering de subsecuencias se encuentra el descubrimiento de motivos y aunque no se ha utilizado para detectar el detectar anomalías, en [8] se probó la aplicación de los perfiles de matriz (estructura de datos que facilita el descubrimiento de motivos y discordias) para la detección de intrusiones en redes industriales logrando un bajo número de falsos positivos y un buen rendimiento comparándolo con métodos estadísticos y de aprendizaje profundo.

Parte II

Estudio Teórico

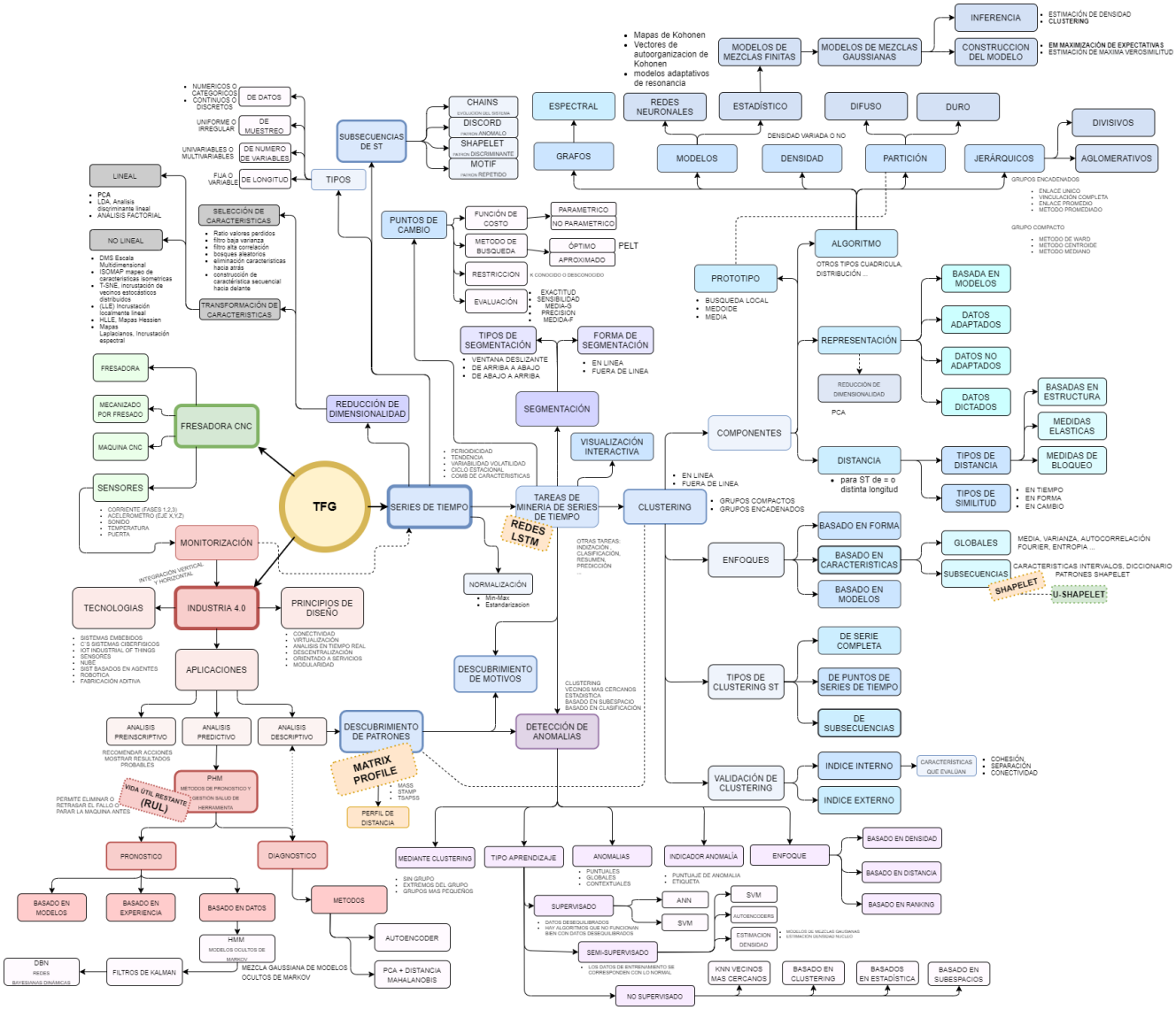


Figura 5.1: Mapa guía del estudio

Contiene las conceptos que se han explorado para la realización del trabajo, en este apartado se explicarán los conceptos de la parte derecha del diagrama (azul y morado) incluyendo los referentes al clustering, descubrimiento de motivos y detección de anomalías.

Capítulo 6

Aprendizaje no supervisado y clustering

El **Aprendizaje automático** es el campo dentro de la informática que se encarga del estudio, diseño y desarrollo de algoritmos que sean capaces de aprender a partir de unos datos para después tomar decisiones basadas en ellos [29].

Estos algoritmos son capaces de hallar patrones o tendencias a partir de unos datos que les permitan predecir o clasificar. En una definición dada por Mitchell en 1997, los definió como programas capaces de aprender de una experiencia E con respecto a alguna tarea T y alguna medida de rendimiento P, cuando su rendimiento en T, medido por P, mejora con la experiencia E.

Existen varios tipos de aprendizaje automático:

- **Aprendizaje supervisado** donde se conoce la clase a la que pertenecen los datos, es decir, incluyen los resultados que se pretende obtener con el algoritmo y que tiene como objetivo aprender un modelo y predecir resultados. Tienen retroalimentación directa y entre las tareas del aprendizaje supervisado se encuentra la clasificación o la regresión.
- **Aprendizaje no supervisado** no se conoce a que clase pertenecen los datos y su objetivo es descubrir patrones ocultos en ellos. No tiene retroalimentación directo y una de las tareas del aprendizaje no supervisado es el clustering
- **Aprendizaje Semi supervisado** no se conoce a que clase pertenecen todos los datos, pero sí se conoce cierta información de alguno de ellos.
- **Aprendizaje por refuerzo** que tiene como objetivo construir un sistema capaz de

mejorar su rendimiento gracias a su entorno. En este aprendizaje un agente explora un entorno y determinará acciones que irá aprendiendo a través de un sistema de penalizaciones y recompensas.

- **Aprendizaje autosupervisado** se trata de un tipo de aprendizaje supervisado donde la clase no se conocía y ha sido determinada de forma automática mediante otros datos o de otra forma.

Uno de los métodos de aprendizaje no supervisado más utilizados es el **clustering** o agrupamiento que consiste en clasificar datos sin conocimiento previo de las clases.

6.1 Clustering

El clustering o agrupamiento es una de las técnicas más utilizadas para el descubrimiento de patrones.

Definición 1

*Se denomina **clustering** al proceso de partición no supervisada de un conjunto de datos $D=\{F_1, F_2, ..., F_n\}$ dentro de k grupos $C=\{C_1, C_2, ..., C_k\}$ de acuerdo a una medida de similitud que maximice la similitud entre los objetos que se encuentren en el mismo grupo y minimice la similitud con los datos del resto de grupos.*

Los objetos que se encuentran dentro de un mismo grupo deben compartir características o estar relacionados, tener pequeñas diferencias entre ellos o al menos estar relacionado con otros objetos del grupo.

Se considera que el conjunto de datos es agrupable cuando hay regiones continuas con una densidad relativamente alta, rodeadas por otras regiones continuas cuya densidad es menor.

En el clustering de datos numéricos se distinguen dos tipos de grupo (figura 6.1):

- **Grupos Compactos**: todos los objetos del grupo son similares entre sí y se puede representar el grupo mediante su centro.

- **Grupos Encadenados**: cada objeto del grupo es más similar a otro de los miembros del grupo que a cualquier otro objeto del resto de grupos pudiendo conectar dos objetos del grupo mediante una ruta.

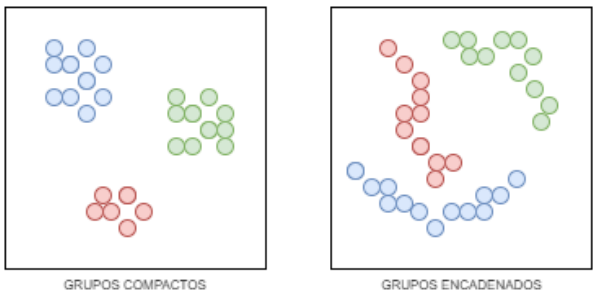


Figura 6.1: Grupos Compactos y encadenados

En el modelado del problema se debe determinar la definición de grupo así como los criterios de separación.

Los métodos de clustering se componen de varios elementos (figura 6.2):

- **Representación o patrón de los datos**: Es el conjunto de características de los datos que se pasan al algoritmo. Puede requerir que previamente se realice una reducción de la dimensionalidad mediante selección (elegir que características de los datos son más efectivas para realizar la agrupación) o extracción de características (transformar los datos en nuevas características que faciliten y mejoren el proceso de agrupación) . El método de representación afecta a la eficiencia y precisión del algoritmo.

- **Medida de similitud**: Métrica capaz de medir la semejanza entre pares de representaciones de los datos. Esta medida debe ser clara y tener un significado práctico.

- **Algoritmo de clustering**: Método que nos permita dividir las representaciones de los datos en grupos.

- **Medidas de Evaluación**: Medidas para analizar la validez del agrupamiento.

Además de estos componentes, algunos métodos requieren de un proceso de abstracción de datos que una vez realizada la agrupación, nos permita obtener una representación compacta y más simple del conjunto de datos.

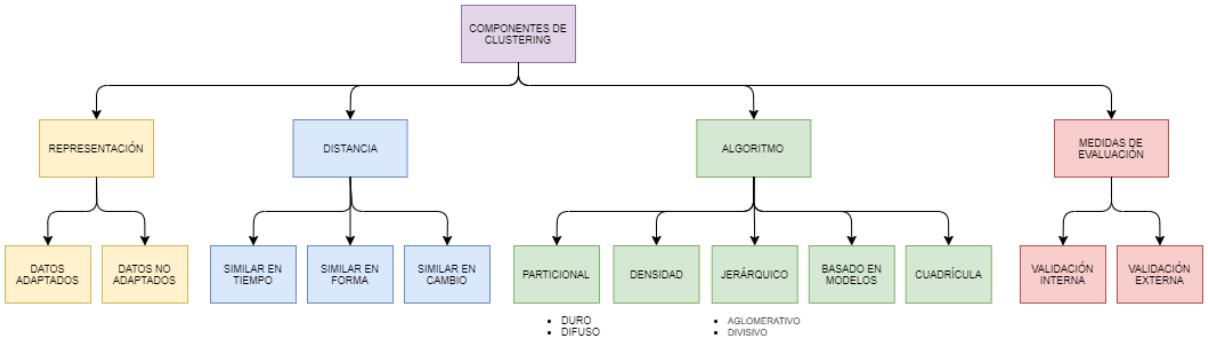


Figura 6.2: Componentes del clustering

A partir de estos componentes se puede realizar un proceso estándar de agrupación (figura 6.3), que se compondría de la obtención de una representación de los datos, del diseño y ejecución del algoritmo de agrupación mediante la medida de similitud apropiada, la evaluación y validación de los resultados obtenidos y una visualización y explicación de los mismos.

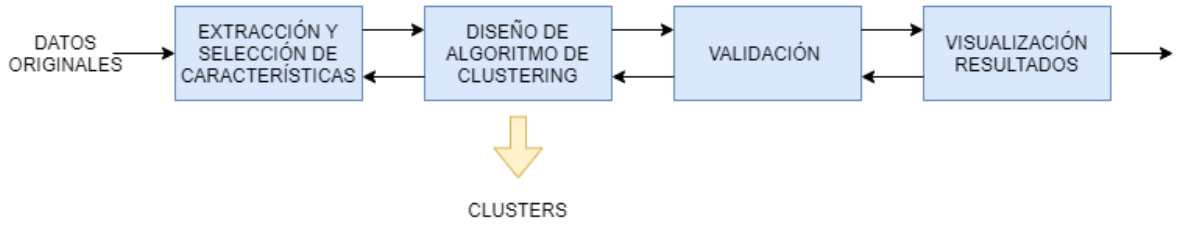


Figura 6.3: Proceso de clustering

Dependiendo del problema, los métodos de clustering que se utilicen deberán tener en cuenta una serie de consideraciones entre las que se encuentran:

- Escalabilidad, ya que no todos los algoritmos funcionan bien para un gran volumen de datos.
- Capacidad de manejar diversos tipos de datos, ya sean categóricos, numéricos o secuenciales entre otros.
- Capacidad de descubrir agrupaciones con formas arbitrarias, debido a que muchos algoritmos hacen suposiciones sobre las formas de las agrupaciones, como por ejemplo, suponer que son esféricas, de forma que luego no funcionen bien para otro tipo de formas.

- El manejo de ruido en los datos.

Además, muchos algoritmos requieren de información sobre el dominio de los datos para establecer los parámetros de los algoritmos de clustering, como puede ser conocer el número de clusters.

Existe una gran diversidad de técnicas tanto para representar los datos, como para medir la similitud entre pares de datos y para formar las agrupaciones de los elementos, así como de evaluar los resultados. Todas estas técnicas no siempre son compatibles entre ellas o funcionan igual de bien.

Además, algunos algoritmos pueden presentar varias configuraciones de grupos, dependiendo de algún otro criterio como el orden en que se analicen los datos.

Para comparar los métodos de agrupamiento se suelen utilizar los criterios para agrupar los datos, la separación entre clusters, la medida de similitud o el espacio en el que trabajan.

La elección de estos componentes (método de representación, algoritmo, medida de similitud y medida de evaluación) dependerá del problema, no existiendo un método de clustering que funcione igual de bien para cualquier situación.

6.2 Tipos de Algoritmos

Los algoritmos de clustering se pueden clasificar en diferentes tipos, los más conocidos son basados en partición, densidad, cuadrícula, jerárquico y basado en modelos. Además se pueden combinar varios algoritmos de distinto tipo para realizar clustering en varios pasos. Cada uno de estos tipos de algoritmos tiene una serie de ventajas y desventajas que les hacen más o menos adecuados dependiendo del problema. La elección del algoritmo dependerá de los datos a agrupar.

6.2.1 Clustering basado en partición o en representantes

Los métodos de partición dividen los datos en k grupos donde cada grupo contiene al menos un elemento. Los clústeres se crean de una sola vez y no existen relaciones jerárquicas entre los grupos obtenidos.

Para crear estos grupos se utiliza un conjunto de elementos representantes, también llamados prototipos de cada uno de los grupos. Estos representantes pueden pertenecer al grupo o crearse a partir de los elementos que lo componen.

Sin embargo, la elección de estos prototipos para conseguir una partición óptima de los elementos es desconocida. Por ello, los algoritmos basados en partición siguen un enfoque iterativo en dos pasos.A partir de los prototipos elegidos inicialmente se asignan los elementos al cluster del prototipo más cercano (Paso de asignación) y tras ello se recalculan los prototipos (Paso de optimización). Estos pasos se repiten, hasta que se cumpla algún requisito preestablecido, como puede ser un error o un límite en el número de iteraciones.

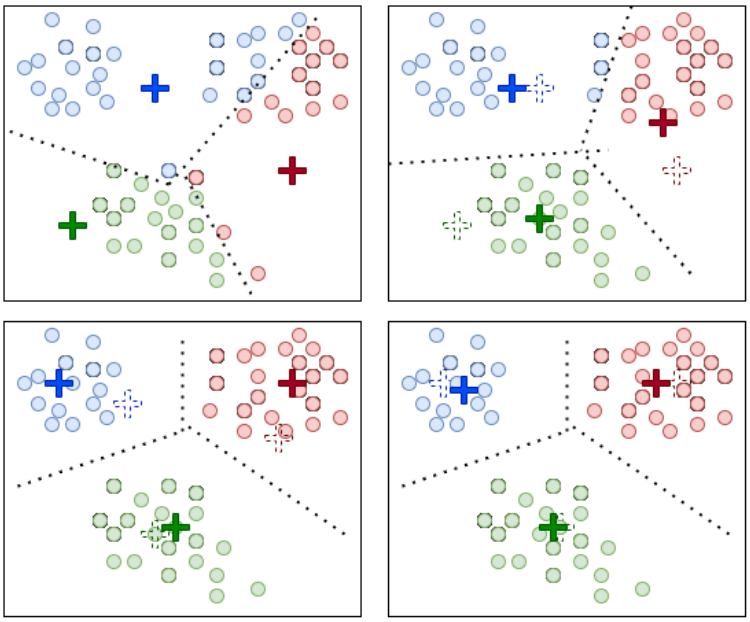


Figura 6.4: Ejemplo de clustering basado en partición

La efectividad del método empleado no solo depende del prototipo que se defina, sino también del método de actualización que se emplee para recalcular los prototipos tras cada iteración del algoritmo.

Estos algoritmos se dividen en agrupación dura, cuando cada elemento pertenece a un grupo y solo a uno y en agrupación difusa, cuando a cada elemento se le asigna un porcentaje de probabilidad de pertenecer a cada uno de los clusters.

Sea U una matriz NxK que represente los elementos en cada uno de los grupos, donde N es el número de elementos que se agrupan y K el número de grupos, las características del clustering duro y difuso se resumen en la siguiente tabla 6.1:

Clustering Duro	Clustering Difuso
$U_{i,j} \epsilon \{0,1\}$ $\begin{matrix} 1 \leq j \leq K \\ 1 \leq i \leq N \end{matrix}$ Cada elemento pertenece o no a un grupo	$U_{i,j} \epsilon [0,1]$ $\begin{matrix} 1 \leq j \leq K \\ 1 \leq i \leq N \end{matrix}$ Cada elemento tiene una probabilidad entre 0 y 1 de pertenecer a un grupo
$\sum_{j=1}^{K} U_{i,j} = 1 \quad 1 \leq j \leq K$ La probabilidad total de que un objeto pertenezca a uno de los grupos es 1	
$\sum_{j=1}^{N} U_{i,j} > 0, \quad 1 \leq j \leq K$ Cada grupo tiene al menos un elemento	

Tabla 6.1: Clustering duro y difuso

Tienen una complejidad baja, son rápidos y suelen dar una buena eficiencia, sin embargo no son adecuados para datos no convexos y requieren conocer el número de particiones. Además su eficiencia queda determinada por el prototipo utilizado.

Los algoritmos más conocidos en esta categoría son k-means, donde se utiliza la media del grupo como prototipo, k-medoids, donde se utiliza el medoide del grupo y sus enfoques difusos como fuzzy c-means.

6.2.2 Clustering basado en densidad

Estos algoritmos agrupan los datos en función de su conectividad y densidad; las regiones con alta densidad pertenecen al mismo grupo. Es decir, un elemento puede seguir expandiendo un grupo con sus elementos cercanos cuando su vecindad, que es el número de elementos cercanos a él, supere dicho umbral.

Tienen alta eficiencia y son capaces de agrupar datos con diferentes formas, pero sus resultados empeoran cuando la densidad del espacio de datos no es uniforme y depende de los parámetros de entrada.

Existen dos enfoques de clustering basado en densidad; conectividad basada en densidad que emplean algoritmos como DBSCAN y OPTICS y un segundo tipo que se basa en la función de densidad, que se aplica en algoritmos como DENCLUE que utiliza además la función de influencia.

6.2.3 Clustering jerárquico

Estos algoritmos establecen una relación jerárquica del conjunto de elementos, permitiendo obtener varias particiones de los grupos (a diferencia del clustering basado en partición donde se obtiene solo una), se dividen en dos tipos:

- **Aglomerativo** o enfoque ascendente se inicializa con un objeto en un grupo independiente y en cada iteración se fusionan los grupos más cercanos hasta que se cumpla un criterio de finalización.
- **Divisivo** o enfoque descendente se inicializa con un solo grupo que contiene a todos los objetos y en cada iteración se van dividiendo los grupos hasta alcanzar un criterio.

Para realizar la fusión o la división estos métodos pueden basarse en varios criterios como distancia o en densidad. La mayoría de métodos utilizan la distancia.

En función de la distancia hay varias formas de determinar los grupos que se deben fusionar (figura 6.5):

- **Enlace Simple** cuando la distancia entre 2 grupos se determina calculando la distancia entre cada objeto del primer grupo con todos los del segundo grupo y seleccionando la distancia mínima (esto permite obtener clusters con una forma más alargada), se define como:

$$DSL(C_i, C_j) = min\{x \epsilon C_i, y \epsilon C_j \; dist(x, y)\}$$

- **Enlace Completo** cuando la distancia entre 2 grupos es la distancia más larga entre cada par de los elementos que componen los grupos (obteniendo clusters con forma más esférica), se define como:

$$DSL(C_i, C_j) = max\{x \epsilon C_i, y \epsilon C_j \; dist(x, y)\}$$

- **Enlace Promedio** en este caso la distancia entre los grupos es la distancia media entre cada par de objetos de ambos grupos.
- **Distancia al centroide** se determina el centro de cada grupo (centroide) y se calcula la distancia entre dos grupos como la distancia entre sus centroides.

- **Enlace Ward** se fusionan los dos grupos que suponen un aumento mínimo en la varianza. Esto se calcula comparando la varianza de los grupos antes de fusionarlos y después de fusionarlos para hallar el par de grupos que suponen el incremento mínimo. Para determinar qué grupos fusionar se puede usar la fórmula de Lance-Williams.

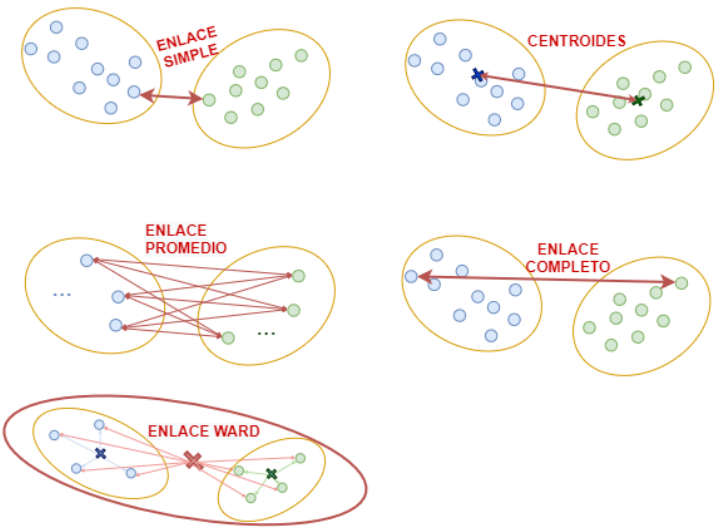


Figura 6.5: Enlaces Jerárquico

La fórmula de Lance-Williams permite actualizar la matriz de disimilitud D tras la fusión de dos grupos en cada iteración del algoritmo. Dado el nuevo grupo $C_{(i,j)}$ formado por la fusión de los grupos i y j la disimilitud con un grupo k, se calcula como:

$$D(C_{(i,j))},C_k) = \alpha_i D(C_i, C_k) + \alpha_j D(C_j, C_k) + \beta D(C_i, C_j) + \gamma \left| D(C_i, C_k) - D(C_j, C_k) \right| \quad (6.1)$$

Según el tipo de enlace empleado los parámetros de la fórmula son:

Métodos	α_i	α_j	β	γ
simple	$1/2$	$1/2$	0	-1/2
completo	$1/2$	$1/2$	0	$1/2$
promedio	$\frac{\lvert C_i\rvert}{\lvert C_i\rvert+\lvert C_j\rvert}$	$\frac{\lvert C_j\rvert}{\lvert C_i\rvert+\lvert C_j\rvert}$	0	0
centroide	$\frac{\lvert C_i\rvert}{\lvert C_i\rvert+\lvert C_j\rvert}$	$\frac{\lvert C_j\rvert}{\lvert C_i\rvert+\lvert C_j\rvert}$	$\frac{\lvert C_i\rvert\cdot\lvert C_j\rvert}{(\lvert C_i\rvert+\lvert C_j\rvert)^2}$	0
ward	$\frac{\lvert C_i\rvert+\lvert C_k\rvert}{\lvert C_i\rvert+\lvert C_j\rvert+\lvert C_k\rvert}$	$\frac{\lvert C_j\rvert+\lvert C_k\rvert}{\lvert C_i\rvert+\lvert C_j\rvert+\lvert C_k\rvert}$	$\frac{\lvert C_k\rvert}{\lvert C_i\rvert+\lvert C_j\rvert+\lvert C_k\rvert}$	0

El enlace completo por lo general produce clusters más compactos que otros enlaces como el enlace simple.

El principal problema de estos algoritmos es que una vez se realiza la fusión o división de un grupo no se puede volver hacia atrás, lo que afecta negativamente a la calidad del agrupamiento y hace que se suelan emplear en enfoques de agrupamiento híbridos.

Aunque la complejidad de estos algoritmos es alta, son algoritmos deterministas que no requieren conocer el número de agrupaciones ni requieren utilizar un prototipo y poseen una alta capacidad de visualización permitiendo representar diferentes agrupaciones y sus relaciones mediante dendrogramas.

Estos dendrogramas (figura 6.6), permiten visualizar la jerarquía de las agrupaciones. Sin embargo, no son adecuados a partir de un número moderado de objetos, ya que el árbol pierde capacidad de visualización al aumentar el número de ramas.

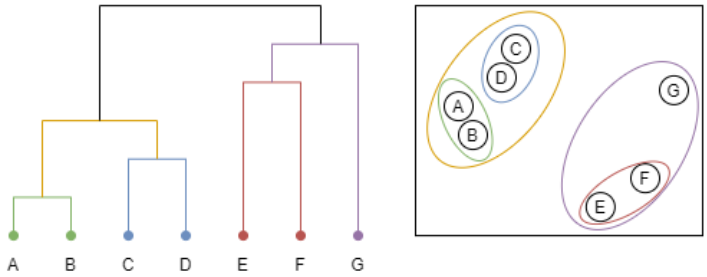


Figura 6.6: Dendograma clustering jerárquico

Además de los dendrogramas, se pueden utilizar los mapas de árbol para visualizar los resultados. Estos diagramas permiten visualizar conjuntos de datos de mayor tamaño. Esta visualización consiste en un área rectangular que se divide recursivamente en rectángulos que representan los niveles de agrupación jerárquica, siendo el tamaño de cada rectángulo proporcional al número de elementos del grupo.

Dentro de este tipo algunos de los algoritmos más conocidos son Birch y Cure.

6.2.4 Clustering basado en cuadrícula

Este tipo de algoritmos dividen o cuantifican el espacio de los elementos de la agrupación en celdas y realiza un agrupamiento de las mismas. Dicho de otro modo estos algoritmos se centran en el espacio de datos en vez de en los datos para realizar la agrupación.

Suelen tener baja complejidad y son altamente escalables, pudiendo aprovechar un procesamiento en paralelo. Sin embargo, son sensibles al número de celdas en que se divide el

espacio; a menor número de celdas mayor velocidad de cálculo pero menor precisión del agrupamiento.

Constan de una serie de pasos básicos:

1. Dividir el espacio en un número finito de celdas
2. Calcular la densidad de cada celda
3. Clasificar celdas por densidades
4. Identificar centros de agrupación
5. Recorrer celdas contiguas

Estos algoritmos no requieren conocer de antemano el número de grupos, sin embargo requieren definir el número de cuadrículas y el umbral de densidad.

Si el número de cuadros es demasiado pequeño se puede dar el caso, de que se sitúen elementos de diferentes grupos en una misma cuadrícula. Sin embargo, si el número es demasiado elevado no solo aumenta la complejidad computacional, sino que también puede darse el caso de cuadrículas vacías dentro de los grupos.

En el caso del umbral de densidad, a menor valor del umbral se producirán menos grupos y más grandes y se detectará menos ruido pero si es demasiado alto puede haber grupos o elementos de los mismos que se identifiquen como ruido.

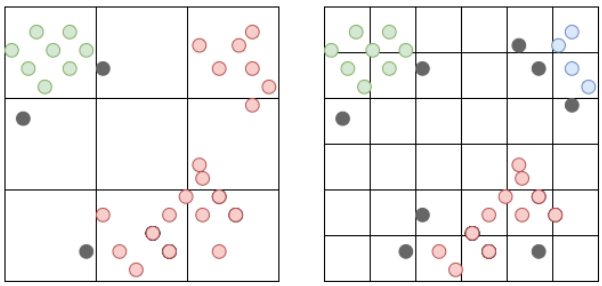


Figura 6.7: Impacto del número de celdas clustering cuadrícula

Los algoritmos STING y CLIQUE se encuentran dentro de este tipo.

6.2.5 Clustering basado en modelos

Los algoritmos de clustering basados en modelos intentan recuperar un modelo original de los datos, es decir, suponen un modelo para cada uno de los grupos e intentan ajustar los elementos a uno de los modelos. Suelen utilizar dos enfoques, los basados en aprendizaje probabilístico y basados en aprendizaje de redes neuronales. Este tipo de algoritmos requieren establecer varios parámetros y tienen un procesamiento bastante lento.

Entre los algoritmos que se basan en aprendizaje probabilístico se encuentra EM y COBWEB y en los que se basan en redes neuronales se encuentra ART y SOM.

Modelos Probabilísticos

En el clustering probabilístico, se presupone que los datos se generan a partir de una distribución mezcla. Una distribución está formada por k componentes que son a su vez distribuciones. Sea C la variable aleatoria del componente, la distribución de mixtura se define como:

$$P(x) = \sum_{i=1}^{k} P(C = i) \cdot P(x \mid C = i) \tag{6.2}$$

El objetivo del clustering probabilístico es obtener el modelo de mixtura al que pertenecen los datos.

Los modelos probabilísticos permiten representar subpoblaciones dentro de una población. La distribución de componentes más común para datos continuos es la gaussiana multivariante, dando lugar a los **modelos de mezclas gaussianas (GMM)**. Estos modelos permiten realizar un clustering difuso.

$$P(x) = \sum_{i=1}^{k} \pi_i \cdot N(x \mid \mu_i, \sigma_i) \tag{6.3}$$

Las mezclas gaussianas se definen por $\pi_i = P(C = i)$ que indica el peso del componente junto con su media μ_i y desviación típica σ_i.

Si se conoce k ,el método más común para determinar el modelo es el *algoritmo EM*. Para

obtener el modelo primero inicializa los parámetros gaussianos de cada componente de forma aleatoria y después alterna hasta lograr la convergencia los siguientes pasos:

- **Paso-E (Expectación)**: Calcula las probabilidades de cada uno de los elementos a pertenecer a cada uno de los cluster, es decir, la probabilidad de cada elemento de haber sido generado por cada uno de los componentes (al suponer distribución gaussiana cuanto más cerca se encuentre el elemento del centro del cluster es más probable que pertenezca a él).

$$\gamma_{i,k} = \frac{\pi_k \cdot N(X_i|\mu_k, \sigma_k)}{\sum_{j=1}^{K} \pi_j \cdot N(X_i|\mu_j, \sigma_j)} \tag{6.4}$$

  Para cada elemento i y cada componente k se debe calcular $\gamma_{i,k}$ que es la probabilidad de que X_i sea generado por el componente C_k.

- **Paso-M (Maximización)**: Recalcula los parámetros de los componentes usando las probabilidades de que los datos pertenezcan a cada uno de los componentes, es decir, maximizar la probabilidad de que los datos se encuentren dentro del grupo.

$$\pi_k = \frac{\sum_{i=1}^{N} \gamma_{i,k}}{N} \tag{6.5}$$

$$\mu_k = \frac{\sum_{i=1}^{N} \gamma_{i,k} \cdot x_i}{\sum_{i=1}^{N} \gamma_{i,k}} \tag{6.6}$$

$$\sigma_k^2 = \frac{\sum_{i=1}^{N} \gamma_{i,k} \cdot (x_i - \mu_k)^2}{\sum_{i=1}^{N} \gamma_{i,k}} \tag{6.7}$$

  Se usan los $\gamma_{i,k}$ del paso de expectación para calcular los parámetros de cada componente k.

Modelos Basados en redes neuronales

En este tipo se encuentran algoritmos como SOM (mapas autoorganizados) que consisten en una red neuronal de una capa donde los clusters se obtienen con la asignación de los objetos a agrupar a las neuronas de salida.

Se trata de aprendizaje no supervisado competitivo y requiere como parámetros el número de grupos y la cuadrícula de neuronas. En una red SOM la capa de entrada y la capa de salida están totalmente conectadas.

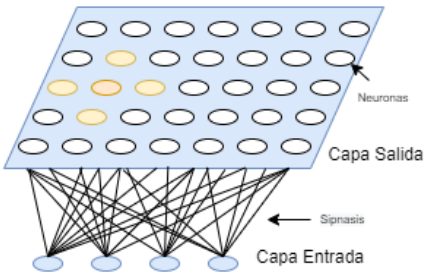


Figura 6.8: Mapa Autoorganizado

En los SOM, los datos se asignan a sus centroides más cercano y cuando se actualizan los centroides también se actualizan los objetos que se encuentran cerca del centroide. Presenta así una proyección del espacio de entrada a un mapa de neuronas bidimensional. Cuenta como ventaja que es fácil de visualizar, una forma de visualizarlo es a través de la proyección de Sammon.

El funcionamiento de una red SOM consiste en que cada conexión tiene un peso $W_{i,j}$ (peso del enlace entre la neurona i y la entrada j) y cada neurona queda definida por sus pesos ($W_{i,1}, W_{i,2}, ..., W_{i,n}$ siendo n el número de entradas). De esta forma las neuronas forman parte del espacio vectorial y son inicializadas con valores aleatorios cercanos a cero.

Luego se calcula para cada fila del conjunto de datos D, previamente normalizado la neurona más cercana, utilizando la fórmula $distancia_j = \sqrt{\sum_{i=1}^{D}(x_i - w_{ji})^2}$. La neurona más cercana se denomina BMU (Mejor unidad coincidente). SOM después, actualiza los pesos para que la fila se aproxime más al BMU correspondiente y traza un radio de influencia en torno al BMU y actualiza los pesos de las celdas que se encuentren dentro de él (teniendo en cuenta la distancia y modificando más los más cercanos), esto se repite con el resto de filas. Tras recorrer todas las filas (habiendo completado así una época) se repite el proceso acortando el radio de influencia del BMU. Finalmente se obtiene un mapa de neuronas deformado que se adapta a los datos de entrada conservando así sus interrelaciones y su estructura.

Además de SOM también se ha realizado clustering basado en redes neuronales usando vectores de cuantificación de aprendizaje de Kohonen (LVQ) y con modelos adaptativos de resonancia (ART).

6.2.6 Otros algoritmos de clustering

Existen más enfoques que se han utilizado para el clustering entre los que se encuentran el clustering basado en grafos donde los nodos representan la relación entre los puntos. Dentro de este enfoque se encuentran algoritmos como CLINK, el clustering espectral donde se construye un gráfico de similitud para luego realizar una incrustación espectral (aplicando autovectores del gráfico laplaciano) y aplicar un algoritmo de clustering tradicional o clustering basados en algoritmos de inteligencia de enjambre entre otros.

6.3 Medidas de Evaluación

Una parte importante de los métodos de clustering es la validación de los resultados obtenidos. Para ello existen principalmente dos tipos de validación:

- **Índice Externo**, se pueden emplear cuando se conoce la verdad (a qué grupo pertenecen los datos) y en los que se compara la solución obtenida con la real. Algunos índices externos son la pureza del grupo, el índice rand o la entropía entre otros.
- **Índice Interno**, no emplean la verdad fundamental para evaluar el resultado del agrupamiento y se basan en evaluar una alta similitud entre los datos del mismo grupo y baja similitud entre diferentes grupos. Entre estos índices se encuentra entre otros el índice de silueta e índice de Dunn.

6.3.1 Índices Internos

En general en los problemas de clustering se desconocen las etiquetas en los datos. En este caso no se pueden utilizar índices externos y en su lugar se utilizan **índices internos** para medir la bondad de la estructura de agrupamiento .

Fisher y Van Ness definieron unos criterios de admisibilidad para comparar los algoritmos de clustering basados en 3 aspectos; el modo en que se forman los grupos, la estructura de los datos y la sensibilidad a los parámetros del algoritmo de cluster empleado.

El objetivo es maximizar la similitud dentro del grupo (cohesión) y minimizar la similitud entre los distintos grupos (separación). La separación, se puede medir calculando la dis-

tancia entre centros o la distancia mínima entre pares de objetos de distinto grupo. Por tanto, las medidas de validación se basan en medir la cohesión, la separación o ambos.

Entre las métricas de evaluación se encuentran:

Índice de Silueta

Mide si está bien agrupado un dato. Para ello calcula la distancia promedio entre grupos. Los valores de este índice se pueden representar mediante el diagrama de silueta, en el que se muestra que tan cerca está cada dato a los datos de los grupos vecinos. Para calcularlo se siguen los siguientes pasos:

1. Calcula a_i como la similitud media entre un dato p_i y el resto de datos del mismo grupo (cohesión).
2. Calcula b_i para cada grupo al que no pertenece el dato p_i se calcula la similitud media entre ese dato y los datos del grupo al que no pertenece y selecciona la distancia más pequeña, es decir, b_i es la diferencia entre p_i y su grupo vecino más cercano (separación).
3. Calcula el valor del índice silueta:

$$S_i = \frac{(b_i - a_i)}{\text{máx}(a_i, b_i)} \tag{6.8}$$

Un dato está bien agrupado si S_i es similar a 1, si es negativo es que p_i está clasificado en en un grupo incorrecto y si es similar a 0 se debe a que el dato se encuentra entre dos grupos.

En un gráfico de silueta se representan los valores del índice de silueta de todos los datos ordenados por grupo y de forma incremental.A través del gráfico se puede observar si los datos tienen un S_i elevado (similar a 1) y están bien agrupados o ‘por el contrario es similar a 0.

Índice de Dunn

Es la relación entre la separación mínima entre grupos y la cohesión del grupo. Cuanto mayor es el índice Dunn mejor, agrupados se encuentran los datos. Para calcularlo se siguen los siguientes pasos:

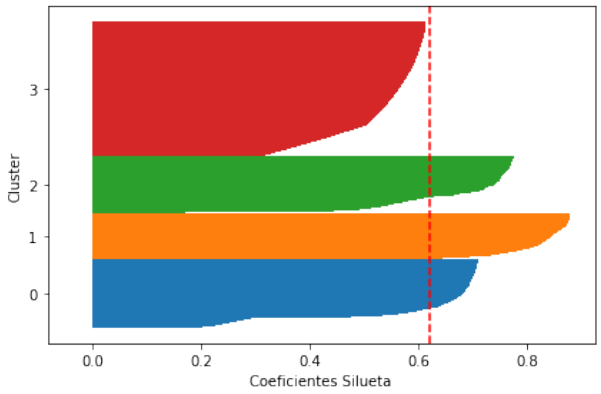


Figura 6.9: Gráfico de silueta

1. Calcula la distancia entre cada dato del grupo y los datos del resto de grupos.

2. Seleccionar la distancia mínima de las calculadas (separación mínima).

3. Para cada grupo calcula la distancia entre los datos del mismo grupo.

4. Selecciona la distancia máxima de separación entre dos datos del mismo grupo (cohesión).

5. Obtiene el índice de Dunn como:

$$D = \frac{sepMin}{cohesion} \tag{6.9}$$

Davies-bouldin

Se debe minimizar para conseguir que los grupos sean más compactos y están más separados, su principal inconveniente es que no detecta bien las formas.

Para hallarla, se calcula la medida $D_{i,j}$ de cada par de clusters, que consiste en la suma de la dispersión (media de la distancia entre cada punto del cluster al centroide) entre el cluster i y el cluster j, dividida entre la distancia entre los centroides de ambos grupos.

Una vez calculado los $D_{i,j}$ de cada par de cluster, el índice de Davies-Bouldin se obtiene como el máximo valor de $D_{i,j}$ entre el número de cluster.

El índice de Davies-Bouldin finalmente se obtiene como el máximo de los $D_{i,j}$ entre el

número de grupos:

$$DB = \frac{1}{K} \cdot \left(\sum_{i=1}^{i=K} \cdot max_{j,j \neq i}(D_{i,j}) \right)$$
$$D_{i,j} = \frac{\frac{1}{n_i} \cdot \sum_{x \epsilon C_i} \cdot d(x,c_i) + \frac{1}{n_j} \cdot \sum_{x \epsilon C_j} \cdot d(x,c_j)}{d(c_i,c_j)} \tag{6.10}$$

Índice S Dbw

Se calcula como la suma de la densidad entre los clusters que se usa para medir la separación entre grupos y la dispersión promedio de los cluster que se usa para medir la cohesión. Este índice comprueba que la densidad de al menos uno de los centros de los cluster sea mayor que la densidad en el punto medio de ambos.

Para calcular la densidad entre dos cluster se calcula la densidad en el valor medio entre el cluster i y el cluster j y se divide entre el máximo de densidad de los centroides de ambos grupos. Dens_bw se calcula como la suma de la densidad entre cada par de cluster del conjunto entre el número de cluster.

La varianza intra-cluster se calcula como la suma de la desviación típica de los datos de cada grupo a su centroide dividido entre la desviación total de todos los datos al centro del conjunto de datos y entre el número de cluster.

$$Dens\_bw = \frac{1}{K(K-1))} \sum_{i=1}^{K} \sum_{j=1\, i \neq j}^{K} \frac{densidad(u_{ij})}{max(densidad(c_i), densidad(c_j))}$$
$$Scat = \frac{1}{K} \sum_{i=i}^{K} \|\sigma(c_i)\| \,/\, \|\sigma(D)\| \tag{6.11}$$
$$S\_Dbw = Scat + Dens\_bw$$

Índice de Calinski-Harabasz

Se obtiene como el cociente entre la varianza dentro del cluster y la varianza fuera del cluster. Consiste en calcular la varianza entre la media de cada cluster respecto a la media de todo el conjunto de datos y dividirlo entre la suma de las varianzas de cada uno de los grupos.

$$CH = \frac{(\sum_{k=0}^{K} |C_k| \cdot \|\mu_k - \mu\|^2) \cdot (N-K)}{(\sum_{k=1}^{K} \sum_{i=0}^{|C_k|} \|x_i - \mu_k\|^2) \cdot (K-1)} \tag{6.12}$$

Dónde $|C_k|$ es el tamaño del grupo k.

Xie Beni

Divide la cohesión de los grupos entre la separación de los mismos expresados como la razón de la distancia media dentro de cada grupo (suma de la distancias de cada dato con el centro de su respectivo cluster, distancia intra-cluster) entre la separación mínima entre los centros de los cluster.

$$XB = \frac{\sum_{i=1}^{K} \sum_{j=1}^{|C_i|} \cdot d^2(x_j - c_i)}{N \cdot min_{j \neq i}[d^2(c_i - c_j)]} \tag{6.13}$$

Ningún algoritmo de agrupación funciona bien para todas las medidas de evaluación interna, igual que no todas las medidas funcionan bien para todo tipo de datos. Algunas medidas hacen suposiciones en los datos que pueden no ser correctas como solo considerar grupos con formas esféricas. En [38] se comparó el rendimiento de las medidas de evaluación en función de varias características que pueden presentar los datos para obtener el número idóneo de grupos (para realizar la comparación usaron el algoritmo K-Means salvo en los datos sesgados donde el experimento se realizó con Chameleon) llegando a una serie de conclusiones:

- Monotonicidad: hace referencia a cómo se comportan los índices ante el aumento del número de grupos, los índices que solo comparan una característica, la separación o la cohesión aumentan o disminuyen de forma constante ante el aumento de datos mientras otros índices alcanzan un máximo o un mínimo al encontrar el número correcto de grupos.
- Ruido: los índices que utilizan distancias mínimas y máximas para calcular la cohesión y la separación son más sensibles al ruido.
- Densidad: en general la mayoría de índices funcionan bien para diferentes datos con distintas densidades.
- Impacto de subgrupos: Un subgrupo es un grupo que está encerrado en otro grupo donde hay más de un subgrupo. Los índices que miden la separación obtienen máximos cuando los subgrupos se consideran un solo grupo lo que conduce a resultados incorrectos.
- Distribuciones sesgadas: Cuando existen grupos muy grandes y grupos muy pequeños en general la mayoría de índices trabajan bien con datos sesgados, sin embargo el índice de Calinski-harabasz no funciona bien con este tipo de datos.

El estudio reveló que de los índices que comparó solo S Dbw funcionaba bien para todas estas características. Para formas arbitrarias muchas de estas medidas no funcionan bien cuando miden la separación y la cohesión del grupo a través del centro del grupo o a través de pares de puntos.

6.3.2 Índices Externos

Estas medidas se pueden calcular mediante la **matriz de contingencia** donde las columnas de la matriz representan los clusters obtenidos y las filas se utilizan para las etiquetas de clase de los objetos, de esta forma las celdas de la matriz n_ij representan el número de objetos del cluster j que pertenecen a la clase i:

	Cluster 1	...	Cluster k	$\sum$
clase 1	n_{11}	...	n_{1k}	n_{clase_1}
...	...	...	...	
clase c	n_{c1}	...	n_{ck}	n_{clase_c}
$\sum$	$n_{cluster_1}$		$n_{cluster_k}$	n

Tabla 6.2: Matriz de contingencia

Pureza

Se utiliza para medir la homogeneidad de las etiquetas en los clusters obtenidos,es decir , si la mayoría de objetos del grupo pertenece a la misma clase. Para calcularla primero se calcula la pureza de cada grupo mediante la fórmula: $P_j = \frac{1}{n_j} \cdot \text{máx}_i(n_j^i)$, es decir, la pureza de un grupo j es el número máximo de objetos del cluster que pertenecen a la misma clase i. Una vez calculada la pureza de cada clusters se obtiene la pureza de la agrupación mediante:

$$Pureza = \sum_{j=1}^{k} \frac{n_j}{n} \cdot P_j \tag{6.14}$$

Donde k es el número de clusters, n_j el número de objetos que se han agrupado en el cluster j y n el número objetos totales.

Entropía (H)

Es similar a la pureza y se utiliza para medir la homogeneidad de las etiquetas en los clusters obtenidos. Ambos métodos son frecuentes de validación en K-Means. Similar a la pureza para calcular la entropía primero se calcula la entropía asociada a cada cluster j con $H_j = \sum_{i=1}^{c} \frac{n_j^i}{n_j} \cdot \log \frac{n_j^i}{n_j}$ para después calcular la entropía global con la fórmula:

$$\text{Entropía} = \sum_{j=1}^{k} \frac{n_j}{n} \cdot H_j \tag{6.15}$$

Valor-F

Esta medida combina la exhaustividad y la precisión para evaluar la agrupación. La exhaustividad, se define como la relación entre los objetos de la clase i en el cluster j y el total de objetos de la clase i $Recall(i,j) = \frac{n_j^i}{n^i}$, mientras que la precisión se define como la relación entre los objetos de la clase i en el cluster j y el total de objetos en el cluster j $Precision(i,j) = \frac{n_j^i}{n_j}$. Cuanto más altos son los valores-F de los clusters obtenidos mejor es la agrupación. El valor F se calcula como:

$$Valor - F = \sum_{j=1}^{k} \frac{n_j}{n} \max_i \left[\frac{2 \cdot Recall(i,j) \cdot \text{Precisión}(i,j)}{Recall(i,j) + \text{Precisión}(i,j)} \right] \tag{6.16}$$

Siendo j el cluster e i las clases de los objetos.

Información mutua Normalizada

La información mutua es una medida utilizada en teoría de la información que mide la independencia mutua entre dos variables aleatorias.

$$MI = \sum_{j=1}^{k} \sum_{i=1}^{c} \frac{n_j^i}{n} \cdot \log \frac{n_j^i}{n_i \cdot n_j} \tag{6.17}$$

De esta media surge la variante información mutua normalizada donde la MI se divide entre la entropía de los clusters ($H_k = \sum_{j=1}^{k}) \frac{n_j}{n} \log \frac{n_j}{n}$) y la entropía de clases ($H_c =$

$\sum_{i=1}^{c}) \frac{n_i}{n} \log \frac{n_i}{n})$.

$$NMI = \frac{MI \cdot 2}{H_k + H_c} \tag{6.18}$$

A diferencia de otras medidas como la pureza donde su valor mejora al aumentar el número de clusters la información mutua normalizada (NMI) compensa este incremento. Además gracias a esto se puede aplicar para comparar el rendimiento de un algoritmo con diferente número de agrupaciones

Además de estos índices externos se encuentran entre otros el índice de Rand, medida-V, el coeficiente Jaccard y el estadístico de Hubert Γ .

6.4 Clustering en aprendizaje semi-supervisado

Existen múltiples formas de agrupar los datos. A veces se dispone de información adicional del dominio de aplicación que puede orientar el proceso de clustering o que indican alguna restricción que se debe respetar en el resultado. En estos casos, hablamos de un problema de clustering semi-supervisado. Generalmente se distinguen dos tipos de semi-supervisión.

- Supervisión puntual: se conocen las etiquetas de algunos de los datos.
- Supervisión por pares: se conocen restricciones entre algunos datos, pares de objetos que deben estar en el mismo grupo o que tienen que estar en grupos distintos.

En supervisión puntual se distingue la supervisión flexible donde los datos con distintas etiquetas se pueden agrupar en el mismo grupo o supervisión estricta en caso contrario.

Se puede obtener una agrupación semi-supervisada por siembra, utilizando el algoritmo K-Medias y usando como semillas iniciales datos de diferentes etiquetas.

En el caso de supervisión por pares no siempre existe solución, pudiendo darse casos donde las restricciones sean incompatibles. En caso de que las restricciones sean solo de que determinados pares de objetos se encuentren en el mismo grupos, el problema se puede convertir en uno de supervisión puntual.

Para manejar estas restricciones se puede utilizar una variante de K-Medias, conocida como COP-KMedias. El procedimiento de este algoritmo es igual que k-medias, pero

variando el paso de asignación, donde un objeto se asigna al grupo más cercano que no incumpla ninguna restricción. COP-KMedias exige que se cumplan todas las restricciones. Existen otras variaciones para permitir que se puedan incumplir alguna de ellas como PCKmedias donde para lograrlo se modifica la función objetivo del algoritmo tradicional.

Aunque la mayoría de problemas de clustering semi-supervisado utilizan algoritmos basados en partición, se han aplicado la semi-supervisión a otros algoritmos como los jerárquicos. Los algoritmos jerárquicos permiten obtener diferentes niveles de agrupación y en algún nivel todos los datos se agrupan en un mismo grupo. En supervisión por pares, esto significa que en algún nivel se cumplen todas las restricciones de enlace obligatorio y se incumplirán las de enlace incompatible. Para usar semi-supervisión en este tipo de algoritmos, una forma de aplicarla consiste en obtener varias jerarquías en vez de una sola permitiendo separar los objetos con restricciones de enlace incompatible o otro método consiste en usar restricciones de orden, que indiquen que dos objetos no se pueden agrupar juntos antes de que se agrupe un tercer objeto en alguno de los dos grupos.

6.5 Clustering de Series de Tiempo

Los datos de monitorización de la fresadora son una recopilación de valores referentes a sus parámetros de trabajo que se han recopilado durante el tiempo que ha estado encendida, es decir, tienen una correlación temporal. Formalmente se denomina **serie temporal ST** a un conjunto de datos secuenciales $\{y_{t1}, y_{t2}, ...y_{tn}\}$ que se corresponden con un conjunto de N observaciones tomadas en distintos instantes de tiempo $[t1, t2, ..., tn]$.

Estos datos pueden ser de valores reales o discretos, provenir de un muestreo regular o irregular, ser multivariables o univariables y tener una longitud fija o variable. En las fresadoras analizadas se recogen datos con longitud variable que han sido recopilados, una mediante un muestreo irregular donde se analizará una única variable y la otra con muestreo regular y múltiples variables.

Figura 6.10: Características ST

En el análisis de series de tiempo se busca encontrar tendencias y patrones que se repiten

así como anomalías.

Los atributos de la serie temporal se pueden dividir en dos tipos:

- Atributos de comportamiento: son los valores que guardan información de un contexto particular (corriente, acelerómetro, sonido, temperatura ...).
- Atributos contextuales: son los que se refieren a las dependencias implícitas de los datos y hacen referencia a la marca temporal. Los atributos contextuales influyen en los valores de los atributos de comportamiento.

Además de la dependencia temporal de los datos y de su carácter dinámico en el tiempo, estas series suelen tener una alta dimensión que dificulta su manejo y ralentiza el proceso de agrupamiento y son potencialmente ruidosas, presentando valores atípicos, desplazamientos en el tiempo y longitudes variables. Todas estas variantes complican el agrupamiento de series temporales.

La **minería de series de tiempo** es el proceso de extraer conocimiento útil y comprensible de datos de series temporales. Entre las tareas de la minería de series de tiempo se encuentra la indexación para ser capaces de encontrar una serie temporal similar mediante una medida de similitud entre un conjunto de series de tiempo, la detección de anomalías, la segmentación y el clustering.

Todas estas tareas requieren de una representación de alto nivel de los datos que permita trabajar con ellos.

El **clustering de series temporales** consiste en agrupar series de tiempo en función de una medida de similitud. Se utiliza para descubrir patrones interesantes que permiten descubrir motivos de la serie (patrones que se repiten con frecuencia) o detectar patrones anómalos. La agrupación se puede utilizar también como parte de otras tareas de minería de datos como puede ser resumir series de tiempo .

Muchos algoritmos de clustering tradicional no trabajan bien con series de tiempo debido a la dependencia temporal de los datos y la complejidad de su estructura; esto hace que se requiera adaptar la tarea de clustering a las características de las series temporal.

El clustering de series de tiempo hace referencia a un tipo especial de clustering donde los datos a agrupar son dinámicos, cambian con el tiempo. Este agrupamiento requiere encontrar una distancia de similitud adecuada.

Definición 2

*Se denomina **clustering de series temporales** al proceso de partición no supervisada de varias series temporales ST={y_1, y_2, ..., y_n} dentro de k grupos C={C_1, C_2, ..., C_k} en función de cierta medida de similitud. Donde C_i se denomina cluster y se cumple que:*

$$ST = \bigcup_{i=1}^{i=k} C_i \qquad y \qquad C_i \cap C_j = \varnothing \forall i \neq j$$

En general estos problemas quedan determinados por los datos de la serie de tiempo que se agrupan, qué estrategia se sigue para realizarlo y los componentes que se utilizan para resolverlo.

Existen distintas formas de aplicar el clustering a las series de tiempo. Dependiendo de qué datos de la serie se agrupen, se distinguen los siguientes:

- **Clustering de series de tiempo completa** donde se agrupan series de tiempo individuales aplicando una distancia de similitud para series temporales y un algoritmo de clustering convencional.
- **Clustering de subsecuencias** se agrupan secuencias extraídas de una serie de tiempo mediante una ventana deslizante.
- **Clustering de puntos de tiempo** se agrupan puntos individuales de una serie temporal teniendo en cuenta tanto la proximidad temporal de los puntos como la proximidad o similitud de sus valores. En este tipo no se agrupan todos los puntos sino que algunos se descartan considerándolos ruidos.

Mientras que el clustering de subsecuencias y el de puntos de tiempo se aplica a una serie, el clustering de series de tiempo completas agrupa varias.

En general el clustering de series de tiempo se compone de un método de representación o reducción de dimensionalidad, un algoritmo de clustering, una medida de distancia y los prototipos (en caso de que el algoritmo de clustering lo requiera).

6.5.1 Enfoques de clustering de series temporales

Existen varias formas de agrupar las series de tiempo:

- **Enfoque Basado en Forma**: Se busca agrupar series de tiempo que tengan una figura similar aunque esté desplazada, estiradas o contraídas en el tiempo. Se utiliza en datos de series de tiempo sin procesar y se aplica un algoritmo de clustering convencional con una medida de similitud adecuada para series de tiempo.

- **Enfoque Basado en Características**: Consiste en agrupar las series de tiempo transformando cada una de ellas en un vector de características de dimensión inferior y generalmente de tamaño fijo que permita aplicar un algoritmo de clustering convencional, normalmente con la distancia euclidiana.

- **Enfoque Basado en Modelos**: Se transforma cada serie de tiempo en parámetros de un modelo y estos parámetros se agrupan a través de una distancia y un algoritmo adecuados al modelo

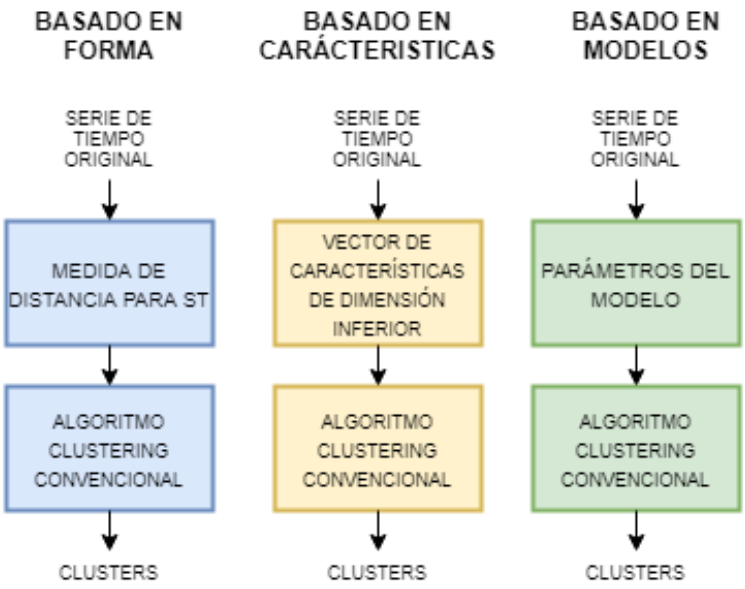


Figura 6.11: Enfoques clustering de series temporales

6.6 Representación de series temporales

En la mayoría de problemas es necesario transformar la serie de tiempo a un espacio de dimensión inferior, no sólo para reducir memoria sino también para acelerar el cálculo de distancia y con ello el tiempo de agrupamiento. Además, una buena representación

permite destacar características de los datos que permitan una mejor agrupación. La representación elegida afecta a la eficiencia y a la precisión del algoritmo. En [63] mencionan la utilidad de representaciones espectrales cuando los datos son altamente periódicos.

Definición 3

*Una **representación de la serie temporal ST'** consiste en la transformación de la serie temporal ST=$\{y_1, y_2, ..., y_n\}$ en otro vector de dimensión inferior ST'=$\{y'_1, y'_2, ..., y'_p\}$ donde p<n. Si dos series ST_a y ST_b son similares entonces sus representaciones ST'_a y ST'_b también deben ser similares entre ellas.*

Algunos algoritmos requieren que los datos cumplan con algunas condiciones como un muestreo regular, lo que requiere una transformación para poder aplicarlos.

Hay distintas formas de representar la serie de tiempo:

- **Datos adaptados** son representaciones que buscan reducir el error de reconstrucción global utilizando segmentos de diferentes tamaños.
- **Datos no adaptados** son representaciones que utilizan segmentos del mismo tamaño lo que facilita la comparación de las series representadas.
- **Basado en modelos** el enfoque de clustering basado en modelos supone que las series proceden de un modelo, en este tipo de clustering las series se representan como parámetros del modelo que subyace.

Algunas de las representaciones más comunes se indican a continuación.

6.6.1 PAA

La aproximación agregada por partes consiste en dividir la serie de tiempo en intervalos de la misma longitud y representar cada uno de los intervalos por su valor medio.

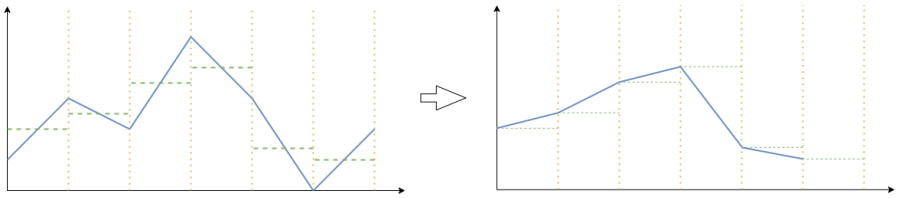


Figura 6.12: Muestreo PAA

6.6.2 APCA

La aproximación constante por partes adaptativa (APCA) es una variante de PAA que permite usar intervalos de distinta longitud. De esta forma permite usar intervalos más largos para partes de la serie de tiempo con poca variación y usar intervalos más pequeños cuando la serie tiene más detalle

6.6.3 SAX

La aproximación simbólica agregada permite transformar una serie de tiempo de valores reales en una secuencia de símbolos discretos pertenecientes a un alfabeto relativamente pequeño. Esta representación primero obtiene la representación PAA de la serie y después, para discretizar los datos, se calculan los puntos de interrupción que permiten dividir el espacio en n regiones equiprobables, siendo n el tamaño del alfabeto (los datos se deben normalizar previamente). Al ser los símbolos equiprobables, las cadenas resultantes tienen todas la misma probabilidad, lo que evita un sesgo estadístico.

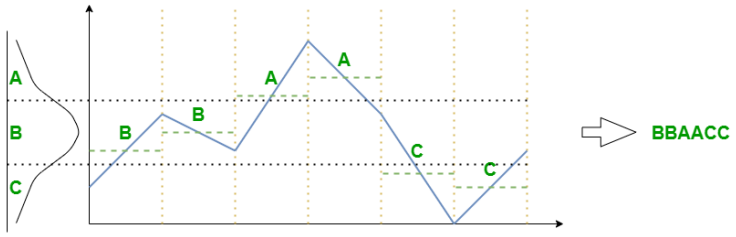


Figura 6.13: SAX

6.6.4 DFT Transformada discreta de Fourier

Cuando la información relevante se repite en intervalos regulares o medianamente regulares se pueden aplicar transformaciones que permitan pasar del dominio del tiempo a

frecuencia. La transformada de Fourier es una representación no adaptativa que permite transformar una serie temporal periódica del dominio del tiempo al de la frecuencia y se basa en que una serie de tiempo se puede representar por una superposición finita de ondas sinusoidales. La transformada discreta de Fourier permite, a partir de una secuencia de datos, obtener la transformada de Fourier mediante la siguiente ecuación:

$$F\{x[n]\} = X(e^{jw}) = \sum_{n=-\infty}^{\infty} x[n] \cdot e^{-jwn} \tag{6.19}$$

Con esta transformación se puede reducir el tamaño de la representación, pasando de la longitud n de la serie original de entrada a n/2 ondas seno.

6.6.5 DWT Transformada discreta de wavelet

Los wavelets permiten representar una serie temporal como la suma o resta de varios componentes. Los primeros componentes contienen una aproximación general de la función y el resto aproximan zonas destacables que requieren de mayor detalle. A diferencia de DFT algunos componentes de wavelets representan solo una parte de los datos en vez de representar el conjunto global.

Hay varios tipos de wavelets. Uno de los más utilizados es el wavelet de Haar que se puede calcular a través de la transformada discreta de wavelet de Haar (DWT).

6.6.6 Otras Representaciones

Además de las mencionadas existen muchas otras representaciones que se pueden aplicar a series temporales incluyendo PLA (aproximación lineal a trozos) que consiste en representar la serie de tiempo mediante segmentos lineales o la descomposición de valores singulares (SVD) representa la serie como una combinación de formas básicas.

6.6.7 Resumen

Aquí se presenta una pequeña tabla con las características principales de las representaciones comentadas.

	PAA	APCA	SAX	DFT	DWT
Datos	no adaptativo	adaptables	adaptables	no adaptativo	no adaptativo
Características	muy rápido.	eficiente pero más complejo.	discretización, depende del tamaño del alfabeto.	mejores resultados en datos altamente periódicos .	mejores resultados que DFT.

6.7 Medidas de Similitud de series temporales

Seleccionar una medida de similitud o disimilitud para series de tiempo es crucial para la agrupación. Dos series de tiempo pueden ser parecidas o distintas en función de las características de la serie que estemos comparando. La elección de una medida de similitud u otra depende además, de la representación de datos elegida existiendo incompatibilidades entre algunas de ellas.

Definición 4

Dada S_i=$\{x_i1$, x_i2, ..., $x_in\}$ y otra serie S_j=$\{x_j1$, x_j2, ..., $x_jn\}$, sí la distancia entre todos los puntos de series de tiempo se ha definido entonces la ***Distancia entre las series temporales****, S_i y S_j se calcula como:*

$$dist(S_i, S_j) = \sum_{t=1}^{T} dist(S_{it}, S_{jt})$$

Podemos clasificar estas medidas de distintas formas, en función de si son específicas para un tipo de representación o son independientes de ella, en el tipo de enfoque de agrupación de series de tiempo en el que se utiliza o el tipo de similitud que mide.

Se pueden medir distintos tipos de similitud, podemos clasificar el tipo de similitud en:

- **Similitud en el tiempo** se busca la similitud de dos series en los mismos instantes de tiempo, para este objetivo la distancia más empleada es la distancia euclidiana. Su cálculo es costoso en series de tiempo sin procesar y se suele utilizar en series transformadas. Las medidas que calculan este tipo de similitud se llaman medidas de bloqueo.

- **Similitud en la forma** se busca similitud de dos series pero no tiene que ser en los mismos instantes de tiempo pudiendo la serie ST_1 ser similar en el instante t_i a otra serie ST_2 en el instante t_j. En este tipo de similitud dos series pueden ser similares aunque estén comprimidas o expandidas en el eje x. Para calcular esta similitud se utilizan métodos elásticos, una de las que más se utiliza es DTW.

- **Similitud en el cambio** busca comportamientos similares que tengan una estructura de correlación similar. Se utilizan enfoques basados en modelos para series de tiempo prolongadas donde se transforma la serie en un modelo y se calcula la distancia entre los parámetros del modelo y enfoques basados en compresión.

De estos tres enfoques la mayoría de casos requieren similitud en forma, en [18] indican que las medidas de similitud en la forma deben cumplir una serie de propiedades:

- Deben permitir reconocer objetos que sean perceptiblemente similares y consistentes con la intuición humana.

- Enfatizar características destacadas a nivel local y global.

- Debe ser robusta ante distorsiones.

- Debe ser invariable a un diversas transformaciones en las series entre las que se incluyen:

  - Desplazamiento de amplitud ($S' = \{x'_1, x'_2, ..., x'_n\}$ donde $x'_i = x_i + k$ y $k \in \mathbb{R}$).
  - Amplificación Uniforme ($S' = \{x'_1, x'_2, ..., x'_n\}$ donde $x'_i = k \cdot x_i$ y $k \in \mathbb{R}$).
  - Escalado en tiempo uniforme ($S' = \{x'_1, x'_2, ..., x'_n\}$ donde $x'_i = x_{k \cdot i}$ y $k \in \mathbb{R}$).
  - Amplificación Dinámica (la serie original multiplicada por una función de amplitud que solo es 0 si la original en ese instante también es 0).
  - Escalado en tiempo dinámico (el índice de tiempo de la serie temporal se multiplica por una función positiva y estrictamente dinámica).
  - Adición de ruido (adicción de ruido blanco independiente y uniformemente distribuido).
  - Valores atípicos (adicción de valores anómalos en posiciones aleatorias).

De las transformaciones mencionadas en [18] se distinguen varios tipos de robustez de la medida de similitud: robustez por escalado ante un cambio de amplitud, robustez ante deformación del eje temporal, robustez ante el ruido y robustez ante valores atípicos.

En [4] concluyen que los enfoques más efectivos se basan en programación dinámica pero que son altamente costosos así como las dificultades para resolver incompatibilidades entre los métodos de representación y la medida de similitud.

Hay que tener en cuenta que para el clustering no se requiere que las medidas de similitud que se utilizan sean estrictamente una métrica. Las medidas de similitud que no son métricas incumplen alguno de los axiomas siguientes.

Definición 5

Una métrica sobre un conjunto X es una función d: $\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ *que cumple*

1. $d(X, Y) \geq 0$ *(No negativa).*
2. $d(X, Y) = 0 \Leftrightarrow X = Y$ *(definida positiva).*
3. $d(X, Y) = d(Y, X)$ *(simétrica)*
4. $d(X, Z) \leq d(X, Y) + d(Y, Z)$

Las medidas de similitud más utilizadas para el clustering de series temporales son la distancia euclidiana y la distorsión de tiempo dinámica DTW (dynamic time warping).

En [63] compararon el rendimiento de varias distancias, en esta comparación la distancia euclidiana obtuvo un peor rendimiento que DTW, LCSS, EDR , ERP y swale. Teniendo el resto de distancias rendimientos similares.

Además hay que considerar que el tamaño del conjunto de datos puede afectar a la precisión y velocidad del cálculo de similitud, al aumentar el número de datos a agrupar las distancias elásticas mencionadas convergen a la misma precisión que la distancia euclidiana.

6.7.1 Distancia DTW

La distorsión de tiempo dinámico permite medir la similitud de dos series temporales que pueden estar desplazadas en el tiempo, para ello alinea las series antes de calcular la similitud a diferencia de la distancia euclidiana.

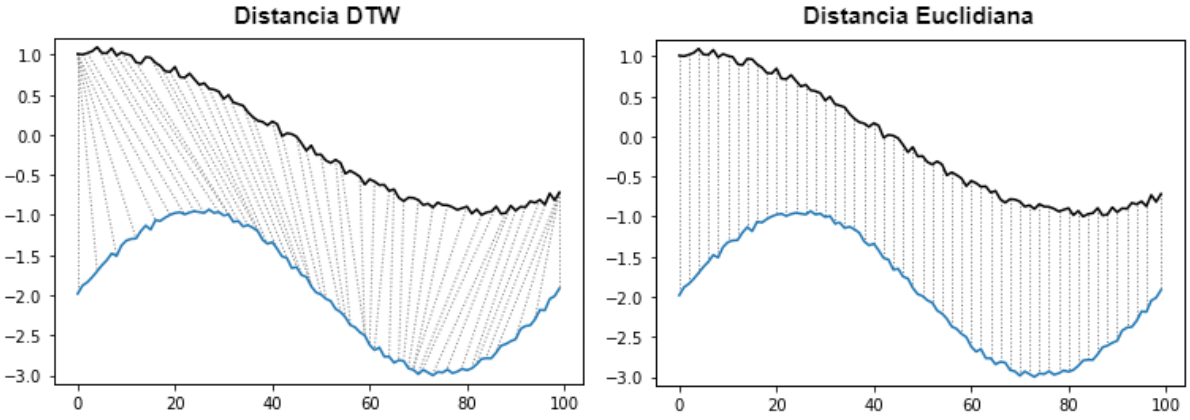


Figura 6.14: Diferencia entre DTW y distancia euclidiana

Sea un espacio vectorial $\mathbb{F}$ y sean dos series de tiempo X=(x_1, x_2,...,x_N) e Y=(y_1, y_2,...,y_M) tal que $\forall n \in [1, N]$, $\forall m \in [1, M]$ $x_n, y_m \in \mathbb{F}$ se define la distancia local o coste como $c : \mathbb{F} \times \mathbb{F} \rightarrow \mathbb{R}_{\geq}0$ y a la matriz formada por la distancia de cada par de elementos llamada matriz de costes como $C \in \mathbb{R}^{N \times M}$. Los caminos de alineación p (warping path) de coste mínimo permiten alinear las series, un camino de alineación se define como $p = (p_1, p_2, ..., p_L)$, con $p_i = (n_i, m_i) \in [1, N] \times [1, M], \forall i \in [1, L]$.

El camino de alineación p debe cumplir:

1. **Condición de Contorno** los primeros y últimos elementos deben estar alineados $p_1 = (1, 1)$ y $p_L = (N, M)$. La ruta comienza en la esquina inferior izquierda y termina en la esquina superior derecha, con esta condición se evita que la alineación no tome solo una subsecuencia de alguna de las series de tiempo.

2. **Condición de monotonía** para que los elementos se sucedan alineados $n_1 \leq n_2 \leq ... \leq n_L$ y $m_1 \leq m_2 \leq ... \leq m_L$, es decir la ruta no puede retroceder en el tiempo.

3. **Condición de salto** el alineamiento es continuo y debe contener todos los elementos de X e Y $p_{l+1} - p_l \in \{(0, 1), (1, 0), (1, 1)\} \forall l \in [1, L - 1]$, no se pueden saltar índices para trazar la ruta.

Siendo entonces el coste total del camino de alineamiento $c_p(X, Y)$ la suma de los costes locales $c_p(X, Y) = \sum_{l=1}^{L} c(x_{nl}, y_{ml})$

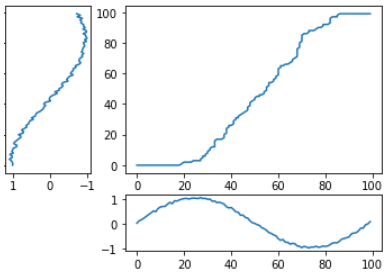


Figura 6.15: Camino DTW

La distancia DTW se define como:

Definición 6

*La **Distancia DTW** es el coste total del camino de alineamiento que tiene el coste mínimo de todos los posible siendo*

$$DTW(X, Y) = c_{p^*} = min\{(X, Y) \mid p \quad es \quad camino \quad de \quad alineamiento\} \tag{6.20}$$

Definiendo DTW de forma recursiva y sea n la el tamaño de la serie X y m el de la serie Y tenemos :

$$DTW(X, Y) = \begin{cases} 0 & \text{Si } n = m = 0 \\ \infty & \text{Si } \begin{matrix} n \neq m \text{ y} \\ n \cdot m = 0 \end{matrix} \\ dist(x_0, y_0) + min \left\{ \begin{matrix} DTW(X_{[1:]}, Y_{[1:]}), DTW(X_{[1:]}, Y), \\ DTW(X, Y_{[1:]}) \end{matrix} \right\} & Resto \end{cases} \tag{6.21}$$

DTW permite medir series de diferente longitud muestreadas regularmente en el tiempo.

Uno de los problemas de DTW es que puede producir resultados poco intuitivos. El uso de restricciones globales y locales permite acelerar ligeramente los cálculos así como prevenir deformaciones patológicas.

Las restricciones globales de ventana como la banda de Sakoe-Chiba y el paralelogramo de Itakura garantizan que no se salten características y se asignen demasiados puntos a características similares. En [48] menciona la popularidad de la banda de Sakoe-Chiba con una restricción del 10 % como restricción global. Aunque el tamaño de la ventana de deformación afecta a la precisión, las restricciones más grandes no siempre mejoran la precisión.

El tamaño de la ventana de restricción no depende solo de la forma de los datos, también depende de la longitud de la serie.

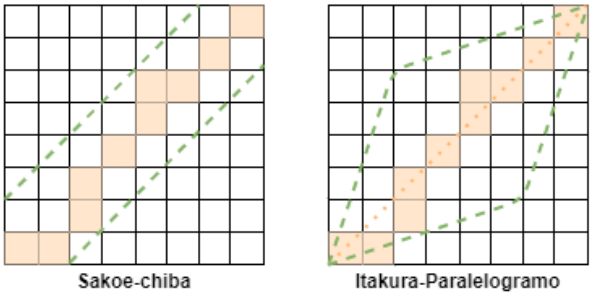


Figura 6.16: Restricciones Globales

Como restricciones locales nos encontramos la restricción de pendiente (para evitar que la ruta sea muy empinada) con la que se evitan que subsecuencias muy cortas se correspondan con subsecuencias muy largas de la otra serie.

Hay varias variaciones de la distancia DTW entre las que se encuentran DDTW que es una combinación ponderada de distancia DTW de las series de tiempo y de sus derivadas de primer orden.

$$DDTW(X,Y) = (1-\alpha) \cdot DTW(X,Y) + \alpha \cdot D_{DTW}(X,Y)) \tag{6.22}$$

6.7.2 Distancia de edición

La distancia de edición (EDIT distance, ED) mide el número de inserciones, eliminaciones e intercambios harían falta para que dos series que pueden ser de distinta longitud sean

iguales. La distancia de edición se calcula como:

$$ED(X,Y) = \begin{cases} m & \text{Si } n = 0 \\ n & \text{Si } m = 0 \\ ED(X_{[1:]}, Y_{[1:]}) & \text{Si } x_0 = y_0 \\ min \begin{Bmatrix} ED(X_{[1:]}, Y_{[1:]}) + 1, ED(X_{[1:]}, Y) + 1, \\ ED(X, Y_{[1:]}) + 1 \end{Bmatrix} & Resto \end{cases} \tag{6.23}$$

La distancia de edición se creó originalmente para la comparación de cadenas y se han aplicado modificaciones a dicha medida para hacerla compatible con series de tiempo. A partir de la distancia de edición surgen las distancias EDR (distancia de edición de secuencia real) donde se aplican penalizaciones en función del tamaño de los espacios entre las series y ERP (distancia de edición con penalización real) donde aplica un punto de referencia constante. Las distancias EDR y ERP se definen como:

$$EDR(X,Y) = \begin{cases} m & \text{Si } n = 0 \\ n & \text{Si } m = 0 \\ min \begin{Bmatrix} EDR(X_{[1:]}, Y_{[1:]}) + subcoste(x_0, y_0), \\ EDR(X_{[1:]}, Y) + 1, EDR(X, Y_{[1:]}) + 1 \end{Bmatrix} & Resto \end{cases} \tag{6.24}$$

Siendo el subcoste 0 sí $|x_0 - y_0| \leq \delta$ y 1 en caso contrario. La otra variante utiliza un valor constante llamado g (el valor más fácil es g=0). La distancia ERP se calcula como:

$$ERP(X,Y) = \begin{cases} \sum_{i=1}^{n} |x_i - g| & \text{Si } m = 0 \\ \sum_{i=1}^{m} |y_i - g| & \text{Si } n = 0 \\ min \begin{Bmatrix} ERP(X_{[1:]}, Y_{[1:]}) + |x_0 - y_0|, \\ ERP(X_{[1:]}, Y) + |x_0 - g|, \\ ERP(X, Y_{[1:]}) + |y_0 - g| \end{Bmatrix} & Resto \end{cases} \tag{6.25}$$

6.7.3 LCSS

La subsecuencia común mas larga es una variante de la distancia de edición que permite comparar dos series sin reorganizar los elementos pero permitiendo que algunos no coincidan (evita que se tengan que comparar todos los elementos).

LCSS se calcula como:

$$LCSS(X,Y) = \begin{cases} 0 & \text{Si } X \cdot Y = 0 \\ 1 + LCSS(X_{[1:]}, Y_{[1:]}) & \text{Si } x_0 = y_0 \\ max \left\{ LCSS(X, Y_{[1:]}), LCSS(X_{[1:]}, Y) \right\} & \text{Resto} \end{cases} \tag{6.26}$$

Para aplicarla en series temporales se cambia el concepto de que dos elementos sean iguales para considerarse coincidentes por un valor umbral ε, de esta forma se dice que los puntos x_i y y_i coinciden cuando $x_i \cdot (1-\varepsilon) < y_i < x_i \cdot (1+\varepsilon)$, el valor ε se encuentra entre 0 y 1. Además se utiliza δ para controlar la separación máxima en el eje de tiempo (eje X) para que dos elementos puedan ser coincidentes.

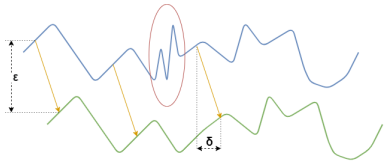


Figura 6.17: LCSS

LCSS para series temporales, dada la serie X de tamaño n y la serie Y de tamaño m, se calcula como:

$$LCSS(X,Y) = \begin{cases} 0 & \text{Si } X \cdot Y = 0 \\ 1 + LCSS(X_{[0:-1]}, Y_{[0:-1]}) & \text{Si } \begin{matrix} |x_m - y_n| < \varepsilon \\ |m-n| < \delta \end{matrix} \text{ y} \\ max\left\{LCSS(X, Y_{[0:-1]}), LCSS(X_{[0:-1]}, Y)\right\} & \text{Resto} \end{cases} \tag{6.27}$$

La distancia LCSS finalmente se calcula como $dist_{LCSS}(X,Y) = 1 - \frac{LCSS(X,Y)}{min\{n,m\}}$.

6.7.4 TWED

La distancia de edición deformada en el tiempo (Time Warp Edit Distance) combina características de la distancia DTW y de la distancia EDR, utilizando como en EDR una penalización λ y un parámetro umbral como DTW denotado como ν que debe ser mayor o igual a 0.

$$TWED(X,Y) = \begin{cases} 0 & \text{Si } n = m = 0 \\ \infty & \text{Si } n \neq m \text{ y } n \cdot m = 0 \\ min\begin{Bmatrix} TWED(X_{[2:]}, Y_{[2:]}) + coste_{coincidencia}(x_1, y_1), \\ TWED(X_{[2:]}, Y_{[1:]}) + coste_{eliminacion}(x_1, x_0), \\ TWED(X_{[1:]}, Y_{[2:]}) + coste_{eliminacion}(y_1, y_0) \end{Bmatrix} & Resto \end{cases} \tag{6.28}$$

$coste_{coincidencia}(x_1, y_1) = |x_1 - y_1| + |x_0 - y_0| + 2 \cdot \nu \cdot (|t_{x_1} - t_{y_1}|)$
$coste_{eliminacion}(a_1, a_0) = |a_1 - a_0| + \nu + \lambda$, Siendo $a \in \{x, y\}$

6.7.5 Resumen

En esta tabla se indican las principales características de las distancias examinadas, indicando complejidad, si son elásticas y si se tratan de métricas.

	Euclidiana	DTW	ERP	LCSS	TWED
Tipo	no elástica	elástica	elástica	elástica	elástica
Métrica	sí	no	sí	no	sí
Características	muy rápida.	más precisa y permite series de distinta longitud.	robusta al ruido .	permite que algunos elementos no coincidan.	permite controlar la elasticidad del método .
Complejidad	$O(n)$	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$

6.8 Prototipos de Series temporales

Algunos algoritmos como los basados en partición requieren de un prototipo o representante de cada grupo. La elección de este representante cuando el grupo está formado por series temporales no es trivial y existen varias maneras de obtenerlo.

El enfoque más común y más sencillo para obtener el prototipo es elegir el medoide del grupo (la serie de tiempo que minimice la distancia de similitud al resto de series del grupo). A parte de este método se han propuesto otros, como calcular una serie promediada donde el valor en cada instante de tiempo es la media de las series del grupo en el mismo instante sin embargo este enfoque sólo funciona cuando las series son equidistantes y de igual longitud.

En [26] proponen otros métodos para calcular el prototipo utilizando la distancia DTW; proponen usar el método de prototipo óptimo, el método promediado y por búsqueda local (que combina el método promediado y las rutas de deformación al promedio obtenido).

Además de elegir el prototipo de la serie temporal, igual que en el resto de algoritmos basados en prototipos se debe definir de forma adecuada el método de actualización.

6.9 Clustering de subsecuencias de series temporales

El clustering de subsecuencias de series temporales, hace referencia a un tipo específico de problema en el que la agrupación se realiza a subsecuencias extraídas de una única serie temporal.

Definición 7

*Una **subsecuencia de una serie temporal** de longitud m de la serie ST=$\{y_1, y_2, ..., y_n\}$ es $ST_{i,n} = \{y_i, y_{i+1}, ..., y_{m+i+1}\}$, donde $1 \leq i \leq m\text{-}n+1$ y $n<m$.*

El enfoque clásico de este problema se realiza mediante una ventana deslizante que extrae todas las subsecuencias posibles de la serie temporal para luego agruparlas, pero este enfoque se ha demostrado que puede conducir a resultados no válidos.

Relacionado con esta tarea encontramos la segmentación que puede ayudarnos a dividir la serie en subsecuencias y el descubrimiento de motivos que aunque no se trata de clustering permite detectar patrones que se repiten a lo largo de la serie.

6.9.1 Segmentación de series temporales

Dentro de las tareas de la minería de series de tiempo se encuentra la segmentación.

Los **puntos de cambio** son datos de una serie temporal que representan una transición entre diferentes estados de un proceso que genera los datos de la serie.

Los métodos de detección de puntos de cambio permiten identificar cambios en los modelos subyacentes que conforman la serie de tiempo. Esta detección se puede realizar tanto en línea como fuera de línea (cuando se realiza fuera de línea a veces se denomina detección de eventos).

Los algoritmos en línea realizan el procesamiento según se van recibiendo los datos y solo van evaluando el cambio más reciente. En cambio, en los algoritmos fuera de línea todos los datos se reciben y procesan al mismo tiempo y evalúan todos los cambios.

En general los métodos de detección de puntos de cambio se pueden dividir según si el

número de cambios es conocido o desconocido y se caracterizan por tres componente: una **función de costo** que mide la homogeneidad del segmento (la función puede ser paramétrica o no), un **método de búsqueda** óptimo o aproximado y una **restricción en el número de puntos cambios** (el número de puntos de cambio puede ser conocido o no). Además existen otros métodos de detección de puntos de cambio que no se ajustan al anterior criterio como los que utilizan aprendizaje bayesiano.

A través de la detección de estos puntos se puede segmentar la serie de tiempo que se puede definir de la forma siguiente:

Definición 8

*La **segmentación de series de tiempo** consiste en la división de la serie temporal en segmentos internamente homogéneos que permiten descubrir la estructura de la serie temporal.*

Figura 6.18: Segmentación de una serie temporal

Además de permitir dividir la serie, la segmentación se puede usar para reducir la dimensionalidad, representando cada uno de los segmentos por parámetros de un modelo o por un valor constante.

Hay que tener en cuenta que para decidir qué método de segmentación elegir, no solo se debe considerar la precisión (que las aproximaciones tengan el mínimo error posible), sino que también se debe considerar su eficiencia (que el tiempo de ejecución y el coste computacional del algoritmo sea aceptable).

El enfoque típico de los algoritmos de segmentación son los nombrados algoritmos basados en aproximación a un modelo, en los cuales se supone que cada segmento ha sido creado por una función (el más simple es usar aproximaciones lineales).

Segmentación de series temporales por aproximación lineal

Este enfoque está inspirado en la representación PLR y consiste en obtener una representación lineal por partes de la serie temporal ya sea por **interpolación lineal** o utilizando **aproximación de mínimos cuadrados** para obtener una recta de regresión.

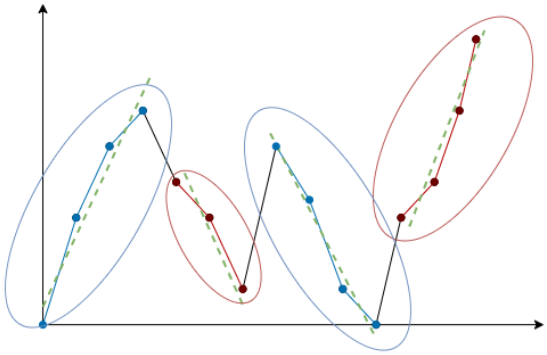


Figura 6.19: Segmentación por aproximación lineal

Estos algoritmos pueden tener diferentes objetivos como:

- Conseguir que ninguno de los segmentos supere un determinado error.
- Conseguir que el error total del conjunto de segmentos lo supere.
- Obtener la partición que minimiza el error dado un número determinado de segmentos.

Los métodos de segmentación de series temporales que siguen esta estrategia se pueden clasificar en 3 categorías:

- **Ventana Deslizante** Dado el punto de inicio de la serie de tiempo, se desliza una ventana hacia la derecha a través de la serie, esta ventana crece mientras el error del segmento no supere el límite de error, cuando se supera el segmento termina y el final del segmento se utiliza como punto de referencia para el inicio del siguiente segmento. Cuando se termina esta segmentación el resultado es una representación PLA de la serie original.
- **De Arriba a Abajo** La serie de tiempo se divide de forma recursiva hasta que se supera un criterio de error. Se toma la serie de tiempo completa y a través de

un método de búsqueda se identifica el punto de ruptura óptimo para dividir la serie de tiempo en dos de forma que se consiga la mayor diferencia posible entre los segmentos resultantes. Estos segmentos a su vez se aproximan y se comprueba el error obtenido si no se cumple con el criterio objetivo se repite el procedimiento con el segmento de mayor error.

- **De Abajo a Arriba** Se fusionan segmentos cortos recursivamente hasta cumplir con el criterio de error. Primero se divide la serie en segmentos muy pequeños y se comparan los segmentos continuos para comprobar que fusión produce el menor incremento del error, una vez determinado los dos segmentos se fusionan y se repite el proceso hasta que se cumpla el criterio objetivo.

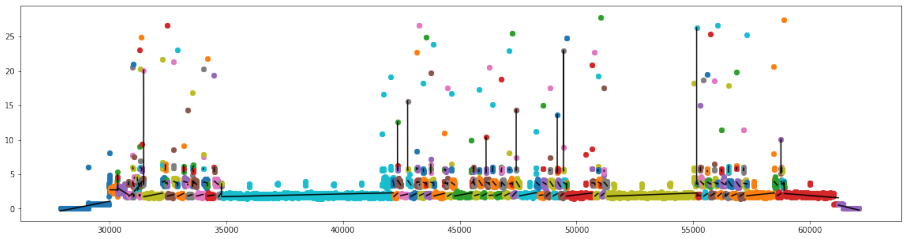


Figura 6.20: Ejemplo de segmentación obtenida por ventana deslizante, los segmentos obtenidos se representan con diferentes colores junto con las rectas a las que se aproxima cada uno de ellos.

Mientras que el algoritmo de ventana deslizante se puede aplicar tanto en línea como fuera de línea los métodos de arriba a abajo y de abajo a arriba solo se pueden aplicar fuera de línea. Sin embargo, la complejidad del algoritmo de ventana deslizante presenta una complejidad mayor ($O(K \cdot n^2)$) que los otros dos métodos ($O(L \cdot n$ siendo L la longitud media de los k segmentos).

La principal desventaja de estos métodos es que son inflexibles y una vez obtenido un segmento no se puede volver atrás; es decir, no se modifica aunque pueda existir un segmento mejor. Por ejemplo, el caso de arriba y abajo el mejor punto de ruptura para dividir la serie en dos puede no ser óptimo para dividir la serie en 3 o más partes.

En [31] además de la clasificación de los tres métodos de segmentación clásicos, presentaron Swab como un algoritmo híbrido que mezcla conceptos del algoritmo de ventana deslizante y el de abajo arriba, que se puede aplicar en línea. La idea de este algoritmo consiste en utilizar un buffer con un tamaño suficiente a varios segmentos y aplicar sobre él el algoritmo de abajo hacia arriba. Una vez aplicado se saca del buffer el segmento de más

a la izquierda y se aplica el algoritmo de ventana deslizante para introducir más datos en el buffer.

6.10 Clustering Significativo de subsecuencias de series temporales

Mediante un tamaño de ventana fija w y una serie de tiempo n es posible extraer $n - w + 1$ subsecuencias desplazando una posición cada vez mediante un algoritmo de ventana deslizante.

Utilizar una ventana deslizante para extraer todas las subsecuencias posibles para después aplicar el clustering, puede dar resultados erróneos. Como se comprobó en [30], esto puede producir agrupamientos donde los resultados obtenidos no dependen de los datos de entrada.

Estos resultados se deben a que en el clustering de series de tiempo, el centroide de los clusters es otra serie de tiempo que tiene como valores los promedios de las series que conforman el cluster y que debería ser similar a ellas. Sin embargo, al aplicar una ventana deslizante e intentar agrupar todas las subsecuencias se obtienen clusters cuyos centroides convergen a ondas sinusoidales y no a patrones reales de la serie temporal.

Esto se debe a que al aplicar la ventana deslizante, los puntos de la serie aparecen en diferentes posiciones de las subsecuencias extraídas (por ejemplo si usáramos una ventana con desplazamiento de 1, obtendríamos subsecuencias de la siguiente forma: $SST_1 = \{x_1, x_2, x_3.., x_w\}, SST_2 = \{x_2, x_3, .., x_{w+2}\}, SST_3 = \{x_3, .., x_{w+3}\}...$) y las subsecuencias extraídas de forma consecutiva suelen ser similares entre ellas de forma que las más similares suelen ser versiones parciales de las mismas y agruparse juntas. Esto produce que al hacer el promedio para extraer el centroide, se sumen los mismos puntos de la serie en diferentes posiciones, dando como resultado centroides que no se parecen a los datos.

A pesar de estos resultados, a través de otras formas de obtención de las subsecuencias se pueden lograr agrupamientos adecuados con centroides que sí representen a los datos.

Algunos trabajos como [42] y [49] proponen no utilizar todos los datos disponibles para lograr resultados correctos . En [42] sugieren que en la mayoría de series temporales

existen estados de transición entre los estados de comportamiento discretos que se deberían ignorar para hacer el clustering.

En ambos trabajos proponen algoritmos que siguen una estrategia de búsqueda codiciosa empleando tres tipos de operaciones para realizar la agrupación.

- Crear: crear un nuevo grupo a partir de las dos subsecuencias más similares que no pertenecen a ningún grupo.
- Actualizar: Añadir una subsecuencia al grupo más similar.
- Fusionar: juntar dos grupos en uno solo.

La idea que sigue, consiste en que en cada iteración se busca la mejor operación que realizar de las que se encuentren disponibles (al inicio de la búsqueda sólo está la operación crear , después las operaciones crear y actualizar ...). La búsqueda se detiene antes de agrupar todas las subsecuencias.

Además en ambos artículos se trabaja con subsecuencias de diferentes longitudes.

Otros trabajos ([59, 69]) proponen agrupar shapelet no supervisados (u-shapelets) de forma que los grupos obtenidos sean de formas representativas de la serie de tiempo. Al igual que los otros trabajos en la agrupación de u-shapelets también se ignoran parte de los datos y se agrupan subsecuencias de diferente longitud.

6.11 Motivos, Shapelets, Discordias y Perfil de matriz

Cuando no se requiere agrupar todos los datos una alternativa al clustering puede ser el descubrimiento de **motivos**. Se entiende por motivo de una serie temporal a una subsecuencia o patrón que se repite varias veces a lo largo de la serie.

Se dice que dos subsecuencias C y M son **coincidentes** cuando la distancia entre ambas es menor que un valor positivo real R que se denomina rango.

Dada C empezada en el instante p y M en el instante q si p y q son iguales o no existe una subsecuencia M' que comience en un instante q' entre p y q, que cumpla $Dist(C, M') > R$ se denomina **coincidencia trivial**.

Definición 9

*El **K- motivo más significativo** es la subsecuencia C_k de la serie temporal ST que tiene el mayor número de coincidencias no triviales y cumple que Dist(C_k,C_i) >2R para $1 \leq i \leq k$.*

La definición de coincidencia trivial hace que subsecuencias que conforman cada uno de los motivos sean excluyentes entre ellas, es decir, que no sean mayoritariamente la misma subsecuencia. Si se utiliza la distancia euclidiana como Dist los motivos serán regiones circulares.

El opuesto a los motivos serían las **discordias**. Las discordias son las subsecuencias más largas que son más diferentes a todo el resto de subsecuencias de la serie.

Una forma de hallar tanto discordias como motivos es utilizar el **Perfil de Matriz** o matrix profile. Esta estructura proporciona un método exacto y sin parámetros que solo requiere del tamaño de subsecuencia. Además se trata de un método rápido e incrementalmente mantenible.

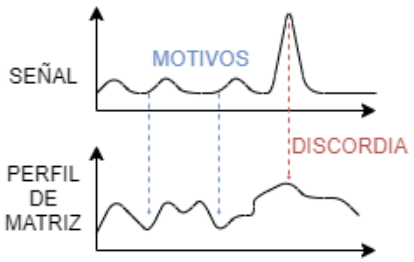


Figura 6.21: Interpretación Perfil de Matriz

El perfil de matriz es una estructura de datos que comenta una serie temporal.Se basa en el concepto de unión de similitud que consiste en comparar subsecuencias de la serie temporal con el resto de subsecuencias de la serie. El perfil de matriz tiene dos componentes:

- **Perfil de distancia** (Distance Profile): es un vector de distancias euclidianas que contiene la distancia mínima normalizada entre una subsecuencia con el resto de subsecuencia de la serie (esta distancia también se denomina distancia de subsecuencia).
- **Índice de Perfil** (Profile index): es un vector que contiene el índice del primer vecino más cercano (la subsecuencia más similar), estos índices no son simétricos.

Para calcularla se utiliza una ventana deslizante de tamaño fijo m y se debe establecer una zona de exclusión para ignorar las coincidencias triviales.

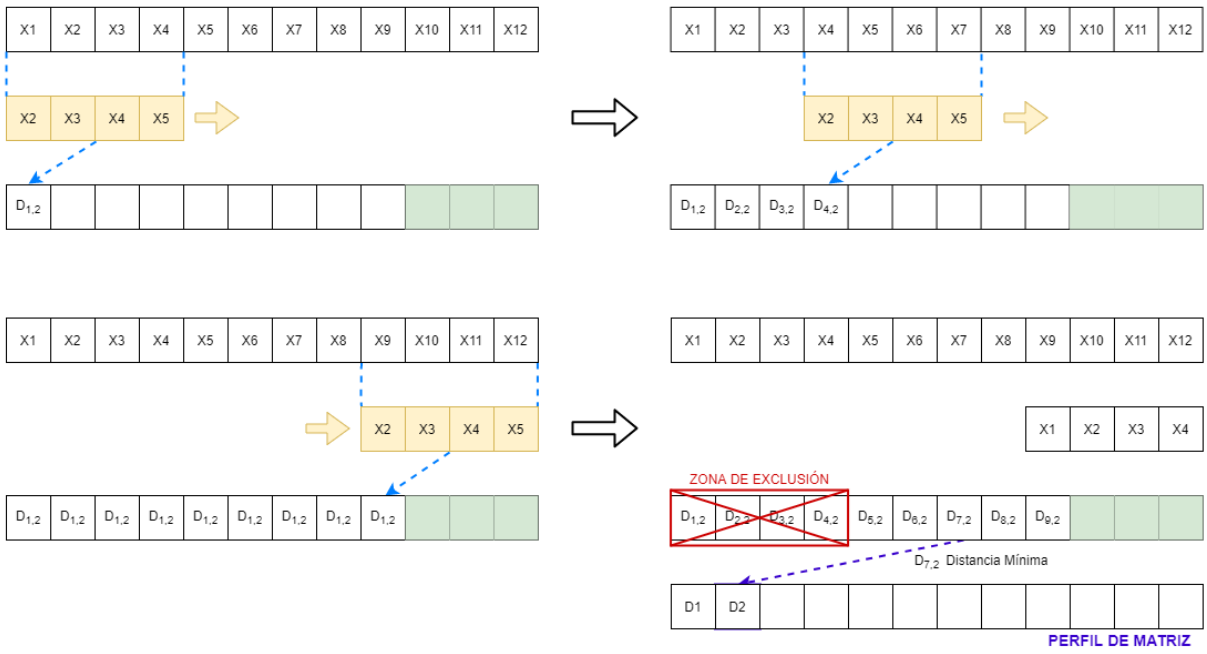


Figura 6.22: Perfil de Matriz

Una vez calculado el perfil de matriz se debe evaluar para hallar los motivos y discordias y visualizar los resultados. Al visualizar el perfil de matriz se puede ver los motivos que se corresponden con los valores mínimos y discordias que se corresponden a los máximos.

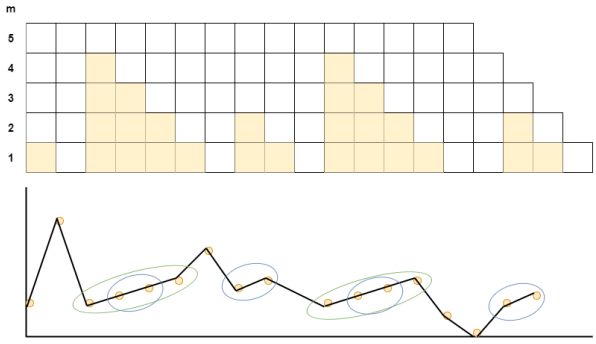


Figura 6.23: PMP

Para determinar un tamaño de ventana adecuado se puede usar el perfil de matriz panorámica PMP que consiste en una matriz cuyas filas son perfiles de matriz para diferentes tamaños de ventana y visualizarla.

Aparte de las discordias y los motivos, recientemente se extendió el concepto de motivos para definir las cadenas de series temporales [72], que son un tipo especial de motivos que

cambian o evolucionan hacia alguna dirección. Estas cadenas pueden ayudar, por ejemplo, a identificar el desgaste de la herramienta u otros procesos graduales en el tiempo.

Una **cadena** de la serie de tiempo es un conjunto ordenado de subsecuencias de una serie $CST = ST_{c1}, ST_{c2}, ..., ST_{ck}$ que cumple que el vecino más cercano a la derecha (subsecuencia posterior más parecida) de ST_{ci} es $ST_{c(i+1)}$ y que de igual forma el vecino más cercano a la izquierda (subsecuencia anterior más parecida) de $ST_{c(i+1)}$ es ST_{ci}, siendo $1 \leq i \leq k-1$. Si ST_{ck} es la última secuencia de la serie de tiempo o se cumple que el vecino más cercano a la izquierda del vecino más cercano a la derecha de ST_{ck} es distinto a ST_{ck} se dice que la cadena está anclada.

Los **shapelets**, por otro lado, son subsecuencias que son patrones significativos y que se podrían considerar de alguna forma máximos representantes de una clase (lo que le da un alto poder predictivo). Los shapelets permiten resultados interpretables y pueden dar lugar a resultados más precisos.

Si el conjunto de series de tiempo ST contiene dos clases A y B la entropía de ST se define como:

$$I(ST) = -p(A) \cdot \log(p(A)) - p(B) \cdot \log(p(B)) \tag{6.29}$$

Siguiendo una estrategia de división ed que permita dividir ST en dos conjuntos se define como ganancia:

$$Ganancia(ed) = I(ST) - I'(ST) = I(ST) - f(ST_1) \cdot I(ST_1) + f(ST_2) \cdot I(ST_2) \tag{6.30}$$

Siendo $f(ST_i)$ el porcentaje de datos en ST_i, $f(ST_i) = \frac{tam(ST_i)}{tam(ST)}$.

Una de las estrategias de división es la distancia al shapelet estableciendo un valor umbral d^{th}. Según esta estrategia ST_i pertenece a un subconjunto si la distancia entre ST_i y S es menor que d^{th} y en caso contrario al otro subconjunto. El valor que maximiza la ganancia se denomina punto de división óptimo $d_{OSP}(ST, S)$.

$$Ganancia(ST, d_{OSP(ST,S)}) \geq Ganancia(ST, d'_{th}) \tag{6.31}$$

Finalmente se puede definir un shapelet como:

Definición 10

Dado el conjunto ST que contiene datos de las clases A y B. Un ***Shapelet(D)*** *es una subsecuencia que con su correspondiente punto de división óptimo obtiene la mayor ganancia:*

$$Ganancia(Shapelet(ST), d_{OSP(Shapelet(ST))}) \geq Ganancia(S, d'_{th}) \tag{6.32}$$

Siendo S otra subsecuencia.

En general son un problema supervisado y se ha aplicado sobretodo para clasificación (permitiendo a través de los shapelets construir un árbol de clasificación). Varios trabajos han abordado el descubrimiento de shapelets no supervisados [69] [59] . Al no conocer las clases en que se dividen los datos, la idea que siguen los **u-shapelets** es encontrar la máxima diferencia entre los dos conjuntos que separa para ello utiliza el concepto de gap. Los algoritmos siguen un método de búsqueda codiciosa para maximizar la diferencia que se calcula como $gap = \mu_b - \sigma_b - (\mu_a + \sigma_a)$.

Los shapelets que se van encontrando tienen un poder discriminativo decreciente siendo el primer u-shapelet encontrado el que mejor divide la serie, en [59] toman como criterio de parada el cambio en el índice de rand, además para considerar un candidato a u-shapelet fijan un rango en la relación entre los tamaños de los conjuntos que divide el shapelet de forma que evita que un grupo se componga de la mayoría de elementos .

Capítulo 7

Detección de Anomalías

Una tarea de minería de datos relacionada con el clustering es la detección de anomalías. Mientras que en el clustering se busca encontrar patrones comunes, en la detección de anomalías se persigue lo contrario, es decir, busca aquellos datos que se alejan del comportamiento habitual.

En el caso de una fresadora, la detección de anomalías pueden informar del desgaste de una herramienta, de un fallo del mecanizado, de rotura de la herramienta ...

Una **anomalía** se define como un elemento, evento u observación que no se ajusta a un patrón esperado u a otros elementos en un conjunto de datos.

La detección de anomalías consiste en utilizar técnicas para identificar patrones que se desvían del comportamiento esperado [13].

Las anomalías son diferentes del ruido. El ruido no aporta información de interés sobre los datos mientras que las anomalías sí, ya que suelen informar de eventos inusuales. Para evitar que el ruido interfiera en la detección de anomalías, se puede eliminar o adaptar (haciendo que este no afecte al modelo y a los resultados).

Incluso sin ruido, a menudo es complicado identificar un comportamiento anómalo de uno que no lo es. Las anomalías generalmente no son producidas por el mismo mecanismo que el resto de datos. Una forma habitual de justificar que datos se corresponden con anomalías, es hacer suposiciones sobre cómo se han producido el resto de datos y justificar por qué las anomalías no cumplen esas suposiciones [29].

La detección de anomalías está relacionada con la **detección de novedades**. Sin embar-

go, las novedades identificadas no suelen ser patrones atípicos y una vez que se detectan se suelen incorporar al comportamiento normal.

Cabe mencionar que aunque en la mayoría de problemas y los métodos de detección suponen que las anomalías son menos frecuentes que los datos a los que se les supone un comportamiento normal, esto no es siempre así.

Existen distintos tipos de anomalías. Estos tipos no son excluyentes pudiendo un dato ser anómalo puntual, contextual y/o global al mismo tiempo. Los distintos tipos de anomalías son:

- **Anomalías Puntuales o Globales**: Son los datos que se encuentran significativamente más separados del resto del conjunto de datos. Son el tipo de anomalía más simple y la mayoría de los métodos de detección de anomalías tienen como objetivo encontrar este tipo de anomalías. La dificultad para detectar estas anomalías, se encuentra en hallar una medida adecuada de la desviación. Existen varios tipos de métodos de detección de anomalías en función de la medida utilizada.

- **Anomalías Contextuales o Condicionales**: Son los datos que se encuentran significativamente separados del resto de datos con respecto a un contexto específico, es decir, son anomalías condicionadas. El contexto se define mediante atributos contextuales y los atributos de comportamiento se utilizan para determinar si el dato es anómalo en esa condición. Esto proporciona una mayor flexibilidad. En este caso, para identificar estas anomalías hay que considerar además la medida de desviación a los atributos contextuales. Para detectar esta anomalías se puede reducir el problema a uno de anomalías puntuales o modelar la estructura de los datos y utilizar el modelo para detectarlas. También se denominan anomalías locales.

- **Anomalías Colectivas**: Es un subconjunto de los datos que se encuentra significativamente separado del resto de datos y que en conjunto representan una anomalía aunque pueda haber puntos que individualmente no lo sean. Para identificar estas anomalías se requiere de más información como la relación entre los datos o las mediciones de similitud entre todos ellos.

Para detectar anomalías contextuales y colectivas, se pueden utilizar algoritmos de detección de anomalías puntuales introduciendo el contexto como una nueva característica. En datos industriales, las anomalías suelen ser contextuales o colectivas.

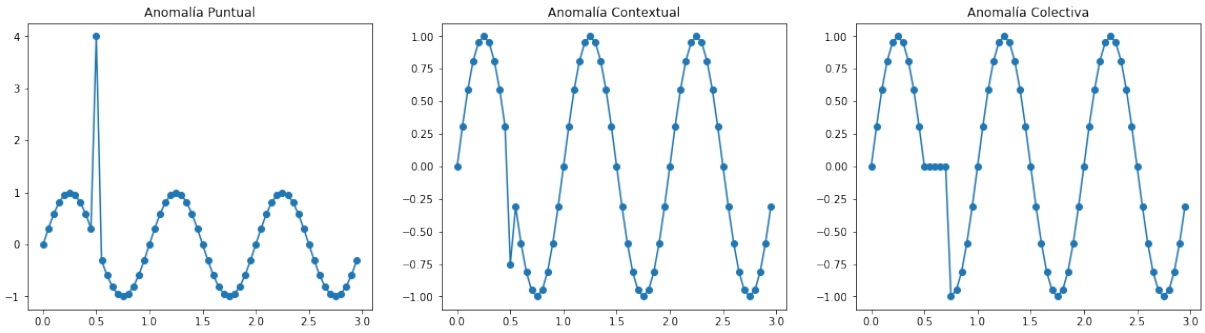


Figura 7.1: Tipos de Anomalías

En las anomalías colectivas un dato no es anómalo en sí mismo, sino que es anómalo cuando coincide con otros datos. Esto implica que debe haber una relación entre los datos (como puede ser una dependencia temporal). La detección de estas anomalías requiere conocer la estructura de las regiones anómalas.

Se pueden distinguir varios problemas de detección de anomalías colectivas como: detección de secuencias anómalas en un conjunto de secuencias, donde un método de resolución de este problema es transformar los datos para convertirlo en un problema de anomalías puntuales; de subsecuencias anómalas en una serie larga, está relacionado con las discordias comentadas en el capítulo anterior; o que la subsecuencia sea anómala porque la frecuencia de aparición de ese patrón es distinta a la que se esperaba.

Las anomalías contextuales solo son anómalas bajo determinadas condiciones. Un enfoque para detectarlas, consiste en aplicar un método de detección de anomalías puntuales bajo esas condiciones o contexto. Igual que en el caso de las anomalías colectivas, un método que se puede aplicar es convertir el problema en uno de detección de anomalías puntuales.

Para detectar anomalías se pueden aplicar técnicas supervisadas, semi-supervisadas y no supervisadas. Sin embargo, en estos problemas no se suele disponer de datos etiquetados o solo se disponen de etiquetas para una parte de los datos (que generalmente se corresponden con etiquetas normales). Por ello en la mayoría de problemas no se pueden utilizar técnicas supervisadas.

Los resultados de la detección de anomalías se pueden expresar mediante **etiquetas** donde se clasifican los datos en anómalos, o no, por **puntuaciones** donde a cada dato se le asigna un grado de anomalía o un grado de normalidad. Para aprendizaje supervisado el resultado se suele expresar con etiquetas mientras que en técnicas semi-supervisadas y no supervisadas se suele utilizar puntuaciones. Las etiquetas aunque ofrecen menos

información, son más concisas y muchos algoritmos presentan este tipo de salida. Mediante un valor de umbral se pueden transformar las puntuaciones a etiquetas.

El enfoque básico para su detección consiste en identificar el comportamiento normal y ver qué elementos escapan de él. Sin embargo es difícil identificar todos los comportamientos normales y sus límites. Además el comportamiento habitual no es estático y cambia con el tiempo. Esto además se agrava en el caso de la fabricación, donde el comportamiento habitual puede variar por una gran variedad de factores como pueden ser el material o la pieza que se esté elaborando.

En el caso de detección supervisada se pueden distinguir dos clases de problemas, los de clase única y los de múltiples clases. Los de clase única suponen que los datos sólo tienen una clase y se basan en establecer un límite para determinar si un dato pertenece a la categoría normal. A diferencia de los de clase única, los de múltiples clases suponen que los datos provienen de distintas clases normales y se basan en aprender un clasificador que pueda distinguir entre ellas; si no se clasifica para ninguna (o en caso de utilizar puntuación no supere el límite establecido), el dato se considera anómalo.

Algunos métodos usados para la detección supervisada incluyen redes neuronales, redes bayesianas, máquinas de vectores de soporte (SVM) y métodos basados en reglas. Una ventaja de estos métodos es su rapidez en la fase de prueba. La necesidad de las etiquetas y los resultados comúnmente devueltos en forma de etiquetas, hacen que sean menos utilizados que los métodos semi-supervisados y los métodos no supervisados.

Para aprendizaje semi-supervisado algunos de los métodos que se han aplicado son máquinas de vectores de soporte de una clase (entrenando la máquina con datos normales), autoencoders y estimaciones de densidad.

7.1 Aprendizaje no supervisado para la detección de anomalías

Para los problemas no supervisadas se han utilizado una gran variedad de métodos que se pueden clasificar en 4 categorías:

7.1.1 Basados en métodos de vecinos más cercanos KNN

El algoritmo de k-vecinos más cercanos (KNN) es un método de aprendizaje perezoso que memoriza el conjunto de datos en lugar de aprender una función de discriminación. El algoritmo se basa en la idea del clasificador KNN que se puede resumir en determinar un número k de vecinos más cercanos obtenidos a partir de una medida de distancia y asignarle la clase a la que pertenezcan la mayoría de estos vecinos.

Para detección de anomalías, se basan en la idea de que los datos normales se encuentran en vecindarios densos y que los datos que se encuentran más lejos de sus vecinos son anómalos. El elemento más decisivo en estos métodos es la elección de la medida de distancia. Según la forma en que se calculen los datos anómalos se dividen en:

- Técnicas que utilizan la distancia: utilizan la distancia al k vecino más cercano o el número de vecinos que se encuentran a una distancia igual o menor que un valor d cómo puntuación.
- Técnicas que utilizan la densidad relativa: en este caso se calcula la densidad del dato y si la densidad de su vecindario es baja se considera anómala. Estas técnicas no funcionan bien cuando existen regiones con diferentes densidades.

Dentro de este tipo se encuentran los algoritmos LOF y COF. El **factor de valor atípico local (LOF)** es un algoritmo que permite identificar anomalías comparando densidades locales, que consiste en:

1. Se identifican los k vecinos más cercanos de cada dato.
2. Se calcula la densidad local de cada dato mediante la densidad de accesibilidad local (LRD):
$$LRD_k(x) = \left(\frac{\sum_{v \in K} d_k(x, v)}{len(K)} \right)^{-1} \tag{7.1}$$
3. Finalmente se calcula la puntuación LOF de x con la siguiente formula:
$$LOF(x) = \frac{\sum_{v \in K} \frac{LRD_k(v)}{LRD_k(x)}}{len(K)} \tag{7.2}$$
Si el valor de LOF es alto indica que la densidad local es baja.

En [24] señalan que este método produce bastantes falsos negativos cuando las anomalías no son locales.

El **factor atípico basado en conectividad (COF)** es otro algoritmo típico basado en k-vecinos más cercanos, es similar a LOF pero se diferencia de este método en su forma de calcular la densidad local.

COF utiliza un enfoque de camino más corto llamado distancia de encadenamiento para calcular la densidad local. Con este enfoque, la densidad se calcula de modo incremental hasta alcanzar los k vecinos. El vecindario se va incrementando con el elemento que tenga distancia mínima con cualquiera de los objetos que se encuentren en ese momento en el vecindario. Este cambio evita la suposición que hace LOF al usar la distancia euclidiana para el cálculo de vecinos de que los datos siguen una distribución esférica. El resto del algoritmo COF es igual que LOF.

Las técnicas basadas en vecinos más cercanos tienen una complejidad alta.

7.1.2 Basadas en métodos estadísticos

Suponen que en un modelo estocástico los datos normales ocurren en regiones que tienen una alta probabilidad, al contrario de las anomalías que ocurren en regiones de baja probabilidad.

En estos métodos se construye un modelo estadístico para explicar el comportamiento normal y después se determina si un elemento es anómalo o no mediante inferencia. Los elementos que tengan una baja probabilidad de haber sido generados por el modelo construido se consideran anómalas.

Se pueden aplicar técnicas paramétricas y no paramétricas.

Las técnicas paramétricas suponen que las instancias normales se generan de una distribución paramétrica de los datos y con una función de densidad. Se asume que se conoce el modelo de distribución y se estiman los parámetros del modelo supuesto a partir de los datos.

Las técnicas paramétricas se pueden clasificar según el modelo que se supone que genera los datos. Entre estas técnicas se encuentran los basados en modelo gaussiano, basados en regresión y basados en distribución de mezclas (suponen que los datos los genera una

combinación de distribuciones paramétricas).

Las técnicas no paramétricas suponen que el modelo es desconocido y utilizan los datos para determinar la estructura del modelo.

El método no paramétrico más simple, es el basado en histograma. Este método consiste en construir el histograma y comprobar si elemento pertenece a alguna de las barras (resultados como etiquetas) o comprobar la altura de la barra a la que pertenece (resultado como puntuación). El ancho de las barras es clave para la detección de anomalías.

Los métodos estadísticos sólo son adecuados cuando la dimensión de los datos es baja.

7.1.3 Clustering para detección de anomalías

A través del clustering podemos descubrir patrones repetidos o detectar sucesos o datos inusuales. Existen varias formas de utilizar métodos de clustering para poder identificar datos anómalos.

- **Las anomalías no pertenecen a ningún cluster**: Consiste en agrupar los datos normales en clusters y los datos que no pertenezcan a ningún cluster identificarlos como anomalías. Para esto se requiere emplear algún algoritmo de clustering que no obligue a cada uno de los elementos a pertenecer a un grupo como puede ser DBSCAN. Sin embargo, hay que tener en cuenta que los algoritmos de clustering están optimizados para encontrar grupos y no para detectar anomalías.

- **Las anomalías son los datos menos similares al resto de su grupo**: En este enfoque los datos que se consideran normales son los que se encuentran cercanos a los centroides del cluster correspondiente y consideran anomalías los que se encuentren lejanos a su centroide. Este enfoque consta de dos pasos, en un primer paso agrupar los datos y en el segundo paso calcular la distancia de cada dato a su centroide. Este enfoque usa la distancia al centroide como puntuación de anomalía. A diferencia del primer enfoque y debido al primer paso se puede utilizar algoritmos que obliguen a que cada dato pertenezca a un cluster. Sin embargo, este método no es aplicable para anomalías colectivas donde existan agrupaciones exclusivas de anomalías.

- **Las anomalías son los grupos más pequeños y con baja densidad**, en un tercer enfoque los grupos grandes y con alta densidad pertenecen a los datos nor-

males mientras que los grupos pequeños y con baja densidad se establecen como anomalías.

El rendimiento de estos métodos depende de la capacidad del algoritmo de poder identificar el comportamiento normal y los métodos de clustering no suelen estar optimizados para esta función, además algunas técnicas no se pueden aplicar cuando las anomalías forman un grupo independiente.

Igual que en el caso de clustering de series de tiempo para la detección de anomalías la elección de la medida de distancia es crucial para el rendimiento del algoritmo.

Como ventaja estas técnicas ofrecen una fase de prueba rápida a pesar de ser computacionalmente complejas.

Algunos algoritmos de clustering para la detección de valores atípicos son **CBLOF** y **LDCOF**.

El **factor de valor atípico local basado en clustering (CBLOF)** consiste en aplicar clustering y luego estimar la densidad de cada cluster. CBLOF clasifica los clusters obtenidos según su tamaño en dos grupos. Para calcular el puntaje de anomalía calcula la distancia de cada elemento a su centroide multiplicado por los elementos del cluster, en caso de que se encuentre en el grupo de clusters de menor tamaño el puntaje se calcula como la distancia al cluster grande más cercano.

Existe una versión de este algoritmo conocido como uCBLOF que difiere en la utilización de la multiplicación por el número de elementos del cluster, debido a que esta multiplicación no estima la densidad local del cluster.

$$CBLOF(x) = \begin{cases} |C_i| \cdot min(d(x, C_j)) & \text{Si } x \in C_i, C_i \in C_{menor} \text{ y } C_j \in C_{mayor} \\ |C_i| \cdot d(x, C_i) & \text{Si } x \in C_i \text{ y } C_i \in C_{mayor} \end{cases} \tag{7.3}$$

$$uCBLOF(x) = \begin{cases} min(d(x, C_j)) & \text{Si } x \in C_i, C_i \in C_{menor} \text{ y } C_j \in C_{mayor} \\ d(x, C_i) & \text{Si } x \in C_i \text{ y } C_i \in C_{mayor} \end{cases} \tag{7.4}$$

En factor de valor atípico basado en clustering de densidad local (LDCOF), tras aplicar el clustering estima la densidad de cada cluster suponiendo una distribución esférica. Lo

que hace es dividir la distancia de un elemento a su centroide por la distancia promedio.

$$LDCOF(x) = \begin{cases} min(d(x, C_j)) \div \frac{\sum_{k \in C_j} d(k, C_j)}{|C_j|} & \text{Si } x \in C_i, C_i \in C_{menor} \text{ y } C_j \in C_{mayor} \\ d(x, C_i) \frac{\sum_{k \in C_i} d(k, C_i)}{|C_i|} & \text{Si } x \in C_i \text{ y } C_i \in C_{mayor} \end{cases} \tag{7.5}$$

7.1.4 Otros enfoques

Además de los enfoques mencionados se han aplicado otros para la detección de anomalías entre los que se incluyen los basados en técnicas de subespacio, en teoría de la información y en técnicas espectrales.

7.2 Detección de anomalías en series temporales

Igual que pasaba en el clustering de series temporales, en las anomalías de series de tiempo nos podemos encontrar puntos atípicos, subsecuencias atípicas y series de tiempo completas que son anómalas.

En el caso de las anomalías puntuales se diferencian dos categorías según tengan en cuenta o no la dimensión temporal de los datos. La forma más común de calcularlas teniendo en cuenta la dimensión temporal, es usar métodos basados en modelos (ya sean modelos de estimación o de predicción). En estos métodos se obtiene un modelo a partir de los datos y se compara el valor previsto con el real, si la diferencia supera un valor umbral establecido, se considera que el dato es anómalo.

Otros métodos que se pueden aplicar teniendo en cuenta la dimensión temporal, son los métodos basados en densidad y los métodos basados en histograma. Los métodos basados en densidad comprueban el número de datos cercanos a un elemento para establecer si es anómalo, dentro de estos métodos, un ejemplo consiste en utilizar ventanas de tiempo de forma que cuando un punto tiene al menos un determinado número de vecinos sucesivos se dice que ese punto no puede ser anómalo para ninguna ventana. En los métodos basados en histogramas, se evalúa la evolución del error de reconstrucción de la serie de tiempo usando una representación en forma de histograma. Se eliminan datos que tienen

posibilidades de ser anómalos y si el error de reconstrucción tras eliminar el dato se ha reducido considerablemente el dato es probable que se trate de una anomalía.

Para la detección de subsecuencias anómalas se distinguen los métodos que consideran subsecuencias de longitud fija y los que las consideran de longitud variable.

Hay diferentes maneras de detectar secuencias anómalas como:

- Identificar las subsecuencias más inusuales de una serie de tiempo (las discordias). Sin embargo, que sea la secuencia menos similar al resto no quiere decir necesariamente que se trate de una anomalía.

- Usar métodos basados en disimilitud que consisten en comparar las subsecuencias con una referencia de lo que es normal (puede tratarse de una serie de tiempo externa, que no pertenece a la serie de tiempo). En este tipo se encontrarán los métodos de clustering.

- Aplicar el modelo de predicción para hallar secuencias anómalas, de forma similar a lo que se aplica para anomalías puntuales. Las predicciones pueden ser tanto puntuales como secuencias.

- Métodos basados en frecuencia, considerando que una secuencia es anómala cuando no aparece con la frecuencia esperada.

- Métodos basados en la teoría de la información que permiten subsecuencias anómalas periódicas en series de tiempo discretas y univariadas. Buscan secuencias que son poco frecuentes (ya que cuanto más se repita menos información aporta) pero que se repiten con valores extraños. Para calificarla como atípica multiplica la información de la secuencia con su probabilidad y si se supera un umbral se considera anómala.

El descubrimiento de series completas de tiempo anómalas, se puede aplicar en series de tiempo multivariadas para identificar variables atípicas. En este caso se pueden aplicar métodos de reducción de dimensionalidad o métodos basados en disimilitud usando como referencia la propia serie multivariada.

7.3 Aprendizaje profundo para detección de anomalías

Los métodos comentados tienen problemas para capturar estructuras complejas como puedan ser las series de tiempo, además cada vez se requiere más poder detectarlas a gran escala. Esto ha incrementado el uso de técnicas de aprendizaje profundo.

Dentro de los métodos de aprendizaje profundo para detección de anomalías encontramos dos tipos, los que utilizan las técnicas de aprendizaje profundo para extraer características que luego se pasan a métodos tradicionales (métodos híbridos) y los métodos de redes neuronales de una clase (OC-NN) que utilizan los datos para crear una clase envolvente que se ajuste en torno a los datos normales (para ello para obtener la representación, usan una función objetivo personalizada para la detección de anomalías) .

En los métodos híbridos, generalmente esta extracción se realiza mediante autoencoders y como método tradicional se puede aplicar entre otros las máquinas de vectores de soporte.

En los modelos OC-NN se realiza un aprendizaje de representación combinado para obtener la puntuación de anomalía y son capaces de extraer factores de variación en la distribución de datos, suponen que los datos anómalos no tienen factores comunes de variación que puedan ser capturados en la capa oculta.

La detección de anomalías profundas supervisadas suelen componerse de dos redes, una red de extracción de características seguida de una de clasificación. Cuando el espacio de características es muy complejo y no lineal no consiguen separar bien los datos normales de los que no lo son.

En el caso de las técnicas semi-supervisadas, la idea es establecer un límite discriminatorio que permita identificar el comportamiento normal. Para lograrlo se basan en que los elementos cercanos entre sí son más probables que se encuentren en la misma clase (proximidad y continuidad) o en aprender características sólidas que retengan los atributos discriminatorios a través de la red neuronal. Estas técnicas tienden a tener problemas de sobreajuste, aunque las redes adversarias generativas (GAN) permiten obtener buenos resultados con datos poco etiquetados [12].

El aprendizaje no supervisado profundo para detección de anomalías requiere basarse en que las regiones anómalas y normales se pueden diferenciar en el espacio de características original o latente o que la mayoría de datos son normales comparándolos con el conjunto de datos. La puntuación de anomalía se obtiene de propiedades intrínsecas de los datos

que se capturan en las capas ocultas de la red.

Los autoencoder son los métodos fundamentales de este tipo de problemas. La idea es que al entrenar la red con datos normales, el autoencoder no puede reconstruir los datos anómalos que tendrán por tanto un error de reconstrucción alto. Hay diferentes arquitecturas de autoencoder y su elección dependerá del tipo de datos, por ejemplo las redes de memoria larga a corto plazo (LSTM) que están diseñadas para aprender dependencias a largo plazo, son un método utilizado cuando los datos de entrada son series de tiempo.

Parte III

Estudio Práctico

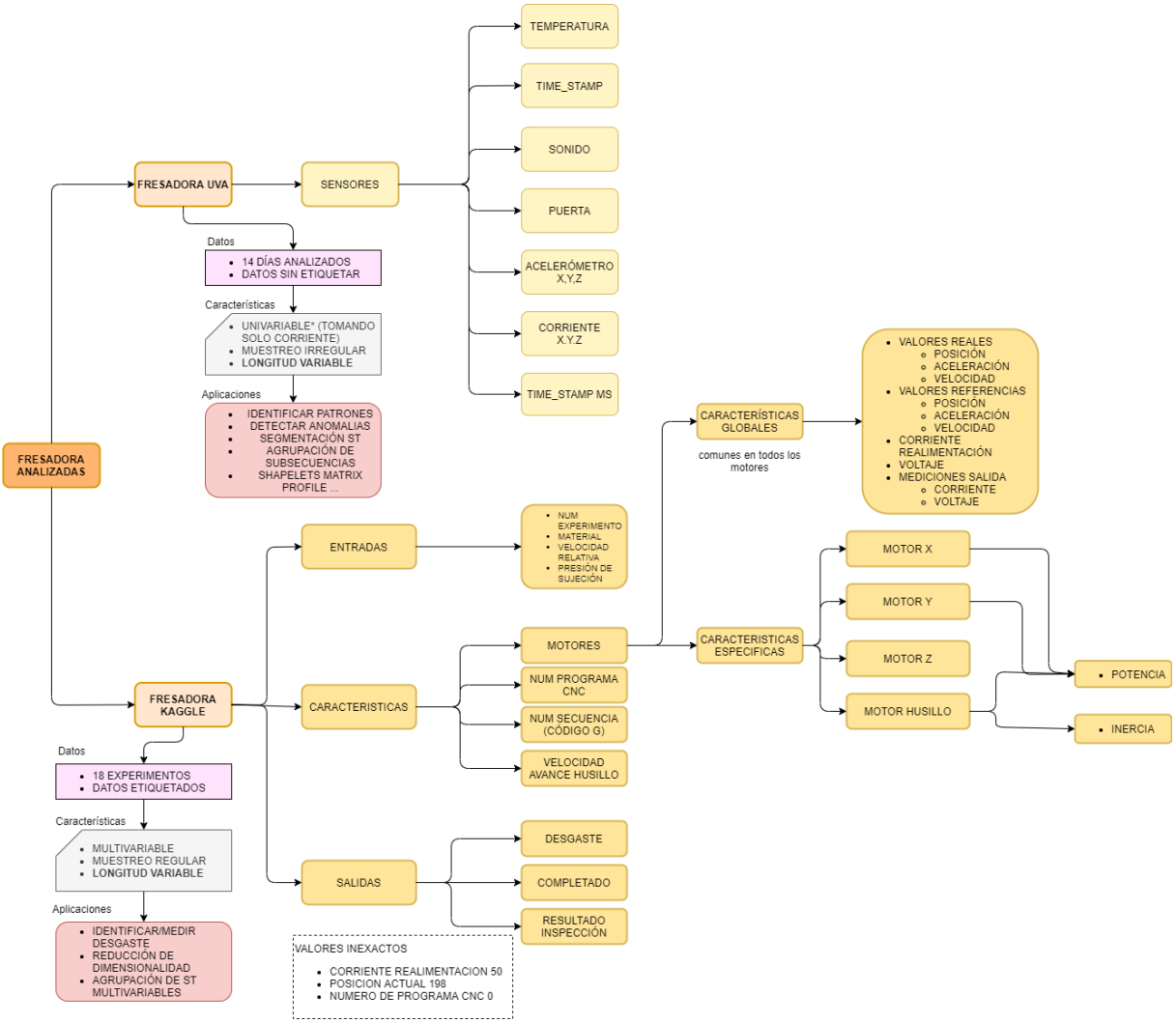


Figura 7.2: Mapa guía datos utilizados

Esquema de las características y de los parámetros de los datos de las fresadoras utilizados para la parte práctica del proyecto.

Capítulo 8

Algoritmos de Clustering Aplicados

En este apartado se explicarán más en detalle algunos los algoritmos de clustering realizados:

8.1 Basados en Partición

8.1.1 K-Medias

Se trata de un método de agrupación basado en partición, que se basa en asignar cada elemento al cluster cuyo valor promedio del grupo sea más cercano, al valor medio de cada grupo se conoce como centroide.Es uno de los algoritmos de agrupación más utilizados.

Este algoritmo requiere de un único parámetro, que es el número de clusters para realizar la agrupación. Sin embargo esto no siempre se conoce a priori y su elección afecta en gran medida a la calidad del agrupamiento,por ello han surgido una serie de heurísticas para poder determinar el número de clusters más adecuado en función de la calidad de los grupos generados para diversos números de grupos, algunas de estas técnicas son el método del codo o los gráficos de silueta.

Es bastante rápido y eficiente y minimiza el error cuadrático de la distancia a los centros de los cluster pero solo es aplicable cuando los clusters son convexos. Este algoritmo es sensible al ruido.

Implementación

Este algoritmo consiste en:

Algoritmo 1: K Medias

Datos: D Conjunto de datos a agrupar, **k** numero de cluster
Resultado: Todos los datos que forman D agrupados en uno de los k cluster

```
1  Inicio;
2  centroides=lista de k elementos aleatorios no repetidos de D;
3  clusters=lista de k cluster ;
4  repetir
5      para cada punto P de D hacer
6          centroideMasCercano=centroides[0];
7          para c en centroides hacer
8              si dist(P,c) <dist(P,centroideMasCercano) entonces
9                  centroideMasCercano=c;
10             fin
11         fin
12     fin
13     Asignar P al cluster al que pertenece centroideMasCercano;
14     para clust en clusters hacer
15         Calcular Promedio del clust;
16         Actualizar centroide;
17     fin
18 hasta que se cumplan un número de iteraciones, que todos los grupos cumplan
     una determinada tolerancia, que los grupos no cambien en esta iteración u otra
     condición de parada.;
```

Determinación de parámetros

K-Means requiere como parámetro el número de clusters y los centroides iniciales (o un método para calcularlos), cuando no se conoce el número de clusters un método habitual para obtener un número adecuado es usar el **método del codo**.

El método del codo emplea la suma de errores al cuadrado (SSE) para comparar la calidad del agrupamiento para diferentes números de cluster (diferentes k).

Siendo SSE la suma de la distancia de cada elemento al centroide del grupo al que perte-

nece:

$$SSE = \sum_{i=1}^{n}\sum_{j=1}^{k} w^{(i,j)} \cdot dist(x_i, c_j)^2 \tag{8.1}$$

donde $w^{(i,j)}$ es 1 si x_i pertenece al grupo j y 0 en caso contrario.

Este método emplea un diagrama que permite visualizar la relación entre diferentes valores de k y los respectivos valores de SSE obtenidos para esos k. La relación es decreciente, al aumentar el número de grupos, los clusters serán más pequeños y los elementos estarán más cerca de sus centroides. Para hallar el k óptimo se escoge el k donde el diagrama presenta un cambio pronunciado (el codo del diagrama).

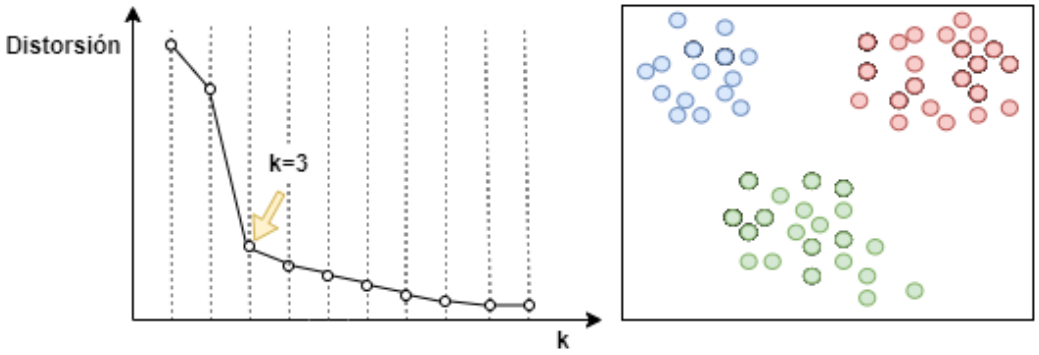


Figura 8.1: Diagrama Codo

Cuando se tiene información de los datos, es posible pasar unos centroides iniciales al algoritmo pero en la mayoría de casos no se tiene información adicional que ayude a determinarlos y escoger los centroides de manera aleatoria puede llevar a una mala agrupación o una agrupación demasiado lenta. Por eso se suele utilizar K-Means ++ para seleccionar los centroides iniciales.

Los pasos para determinar los centroides iniciales a través de k-Means ++ son:

1. Elegir aleatoriamente el primer centroide entre los objetos a agrupar.

2. Hasta llegar a los k centroides:

   2.1 Calcular la distancia cuadrática mínima entre cada uno de los elementos con su centroide más cercano de los previamente seleccionados: $dist_{min}(x, K)^2$ siendo K el conjunto de centroides.

2.2 Seleccionar de forma aleatoria el siguiente centroide c_j usando una distribución de probabilidad ponderada donde cada elemento x tiene de probabilidad. $\frac{dist_{min}(x,K)^2}{\sum_i dist_{min}(x_i,K)^2}$ (distancia mínima a uno de los centroides entre la suma de la distancia mínima a un centroide del resto de elementos del conjunto).

Visualización

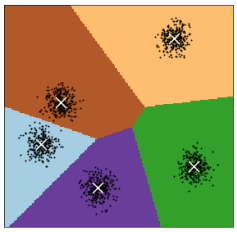


Figura 8.2: Diagrama de Voronoi

A partir de los centroides obtenidos se puede obtener una visualización del espacio de partición a través de los **diagramas de Voronoi**.

Estos diagramas consisten en unir los centroides y trazar las mediatrices de los segmentos resultantes. Las mediatrices y sus intersecciones dividen el espacio en polígonos.

K means con sklearn

La librería sklearn contiene una implementación del algoritmo K-Means. Los parámetros más relevantes para aplicar el algoritmo son:

Parámetro	Descripción
n_clusters	Número de grupos en los que se dividen los datos.
init	Método para determinar los centroides iniciales por defecto es "K-Means++", también permite seleccionarlos aleatoriamente asignando random.ªl parámetro y la posibilidad de pasarle una función así como pasarle al método los centroides iniciales pasándole un array de dimensiones (k,m) siendo n el número de características de los datos a agrupar.
n_init	Repeticiones del algoritmo K-Means cada vez con diferentes centroides para obtener la mejor agrupación posible.
max_iter	Número máximo de iteraciones del algoritmo para unos centroides iniciales dados.

Una vez aplicado el clustering el método K-Means devuelve los siguientes atributos:

Atributo	Descripción
cluster_centers_	centroides de cada uno de los grupos obtenidos, consiste en un array de dimensiones (k,m).
labels_	etiquetas de cada elemento del conjunto que se ha agrupado, los elementos etiquetados con la etiqueta i tienen como centroide el elemento que se encuentra en la posición i de cluster_centers_ .
inertia_	inercia del grupo es la suma de errores cuadráticos (SSE) , es decir, la suma de las distancias de cada elemento a su centroide más cercano.
n_iter_	número de iteraciones realizadas.

TimeSeriesKMeans con tslearn

La implementación del algoritmo K-Means de sklearn no permite modificar la medida de distancia empleada permitiendo usar solo la distancia euclidiana, a través de tslearn se puede aplicar K-Means con la distancia DTW.

Ambos métodos comparten los mismos atributos y parámetros explicados en el apartado anterior y utiliza algunos parámetros adicionales. Entre los parámetros adicionales se encuentra metric, que permite cambiar la distancia euclidiana usada por defecto por DTW o softdtw (variante de DTW donde el operador min con el que se calcula DTW se sustituye por una función soft-min que es diferenciable a diferencia de DTW).

8.1.2 Fuzzy C-means

El algoritmo fuzzy C-means (FCM) es un algoritmo de clustering difuso similar a K-Means donde las muestras no se asignan a un grupo, sino que se asigna una probabilidad de pertenencia a cada uno de los grupos.

Este algoritmo utiliza como función objetivo:

$$J_m = \sum_{i=1}^{n}\sum_{j=1}^{k} w^{m\cdot(i,j)} \cdot dist(x_i, c_j)^2 \tag{8.2}$$

donde $w^{m\cdot(i,j)} = \left[\sum_{p=1}^{k}\left[\frac{dist(x_i,c_j)}{dist(x_i,c_p)}\right]^{\frac{2}{m-1}}\right]^{-1}$ es la probabilidad de pertenencia al grupo y m

es un coeficiente de difusión (a mayor valor de m, más difusos son los grupos siendo su valor habitual 2).

Los centroides se calculan como:

$$c_j = \frac{\sum_{i=1}^{n} w^{m(i,j)} \cdot x_i}{\sum_{i=1}^{n} w^{m(i,j)}} \tag{8.3}$$

8.2 Basados en Densidad

8.2.1 DBSCAN

DBSCAN es uno de los algoritmos de clustering más utilizado debido a que requiere poco conocimiento del dominio de los datos, permite grupos con formas diversas y se puede utilizar para grandes volúmenes de datos. Este algoritmo tiene como objetivo identificar regiones con alta densidad que están separadas por zonas de baja densidad. Para ello utiliza el concepto de vecindario .

Definiciones

Definición 11

El vecindario de un punto p conocido como N(p) se define como:

$$N(p) = \{q \in D \mid dist(p, q) \leq \varepsilon\} \tag{8.4}$$

De esta definición se sacan dos conceptos:

- **Densidad de P**: consiste en el número de puntos que se encuentran a una distancia menor de ε del punto P, en dos dimensiones se puede representar como el número de puntos que se encuentran dentro de una circunferencia centrada en P con radio ε.
- **Región Densa**: conjunto de puntos donde cada uno de ellos tienen como mínimo a una distancia menor de ε un número determinado número de puntos conocido como minPts.

Con ellos podemos definir p como **Punto Central** sí | N(p) | $\geq$ minPts y a p' como **Punto Borde** cuando | N(p') | $\leq$ minPts pero p'se encuentra dentro del vecindario de un punto central p. Los puntos que no se encuentran en estos dos supuestos se consideran **ruido**.

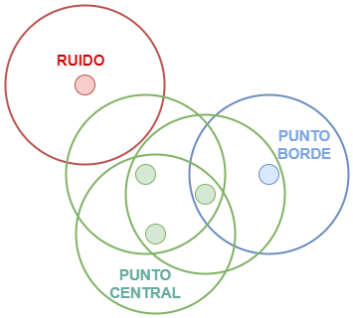


Figura 8.3: Punto Central, Punto Borde y ruido

Se considera que un punto A es **directamente accesibles por densidad** a otro punto B si B es un punto central y A se encuentra en el vecindario de B. Esta propiedad solo es simétrica si tanto A como B son puntos centrales.

Si tenemos una serie de puntos donde $b_1, .., b_n$ donde b_i es directamente accesible por densidad a b_{i+1}, entonces se dice que b_1 es **accesible por densidad** a b_n.

Si un punto A se encuentra en el vecindario de B y un punto C se encuentra en el vecindario de B, entonces se dice que A está **conectada por densidad** con C.

DBSCAN agrupa los datos a través de estos conceptos y de los parámetros ε (umbral de densidad) y minPts. El algoritmo consiste en:

1. Se escoge un punto al azar, se calcula el vecindario a partir de ε, si el vecindario tiene más puntos que minPts se crea un cluster; en caso contrario se etiqueta como ruido. Si se etiqueta como ruido, más adelante ese punto puede pasar a formar parte de otro cluster por los conceptos de accesibilidad y conexión de densidad. Este punto se marca como visitado.

2. Si el vecindario es mayor que minPts y por tanto un es un punto central, se calcula el cluster donde entrarán los puntos de su vecindario que sean puntos centrales así como también los vecindarios de estos puntos que cumplan también esta condición. Todos estos puntos se marcan como visitados.

3. Si quedan puntos no visitados se repite el proceso que puede dar lugar a un nuevo cluster o a que se le etiquete como ruido.

El principal inconveniente de este método es que no permite grupos de densidad variable, además presenta otros inconvenientes como la dificultad para determinar los parámetros

de minPts y ε cuando no se tiene mucho conocimiento del dominio y que al igual que otros algoritmos de agrupamiento sufre de los efectos negativos de la maldición de dimensionalidad al aumentar el número de características para un número fijo de datos de entrenamiento.

Implementación

Los métodos principales que utiliza este algoritmo se detallan a continuación.

Algoritmo 2: DBSCAN

Datos: **D** conjunto de datos a agrupar, **eps** distancia máxima entre dos puntos vecinos, **minPts** mínimo número de puntos para formar un cluster
Resultado: Datos agrupados en clusters

```
1  Inicio;
2  C=0;
3  para cada punto P de D hacer
4      si P no está en visitados entonces
5          Marcar P como visitado;
6          ptosVecinos=calcularRegion(P, eps);
7          si tamaño(ptsVecinos) <minPts entonces
8              marcar P como ruido ;
9          en otro caso
10             C= proximo cluster;
11             expandirCluster(P,ptosVecinos,C,eps,minPts) ;
12         fin
13     fin
14 fin
```

Algoritmo 3: calcularRegion

Datos: **P** punto central del cluster, **ptsVecinos** puntos que se encuentran en la vecindad de P, **C** cluster al que pertenecen **eps** distancia máxima entre dos puntos vecinos, **minPts** mínimo número de puntos para formar un cluster
Resultado: Puntos vecinos de P incluido P

```
1  Inicio;
2  Añadir P al cluster C;
3  para cada punto P' de ptsVecinos hacer
4      si P' no está en visitados entonces
5          Marcar P como visitado;
6          ptosVecinos'=calcularRegion(P', eps);
7          si tamaño(ptsVecinos') >= minPts entonces
8              ptsVecinos+=ptosVecinos';
9          fin
10     fin
11     si P' no pertenece a otro cluster entonces
12         Añadir P' al cluster C
13     fin
14 fin
```

El algoritmo calcularRegion(P', eps) devuelve los puntos del vecindario de P a la distancia eps, incluido el propio P

Determinación de parámetros

Para hallar el valor de ε se pueden utilizar los vecinos más cercanos. La idea es obtener los k vecinos más cercanos de cada uno de los elementos y después juntar en una lista todos los k vecinos más cercanos, ordenar la lista en orden creciente y visualizar el resultado.

DBSCAN con Sklearn

Sklearn contiene una implementación de DBSCAN que presenta una complejidad superior al algoritmo DBSCAN original debido a un alto calculo de vecinos. Este algoritmo presenta una complejidad O(nd) siendo d la media de vecinos de los objetos. Algunos de los parámetros más relevantes son:

Parámetro	Descripción
eps	Indica ε, es decir la distancia máxima para que un elemento se considere dentro del vecindario de otro.
min_samples	Número mínimo de elementos que se deben encontrar dentro de la vecindad de un elemento para que sea considerado punto central.
metric	Métrica de distancia utilizada, por defecto es la distancia euclidiana pero se puede usar otra función de distancia o indicar que la distancia ha sido calculada previamente pasando como parámetro "precomputed".

Una vez entrenado el algoritmo DBSCAN presenta como atributos la distancia al núcleo de los objetos y sus etiquetas.

Al permitir en metric llamar a una función o pasarle al algoritmo una matriz de distancia (en caso de que el parámetro metric valga "precomputed") permite usar diferentes diferentes medidas de similitud incluida DTW.

Esto además, permite aplicar DBSCAN a series de tiempo de diferente longitud sin tener que transformarlas. Para ello se puede usar una medida de similitud que permita series de distinta longitud y pasarle al algoritmo la matriz de distancia en vez de los datos de la serie temporal.

8.2.2 OPTICS

OPTICS es un algoritmo de agrupación basado en densidad que en vez de agrupar un conjunto de datos explícitamente devuelve un orden incremental que representa su estructura de agrupamiento basado en densidad. Su principal ventaja es que no requiere un valor de ε (umbral de densidad) y que permite extraer grupos de diferente densidad.

Definiciones

Cada objeto del clustering queda determinado por su **distancia al núcleo** y su **distancia de alcance**.

Definición 12

*La **Distancia al Núcleo** es el valor mínimo de radio requerido para clasificar un punto dado como un punto central dado un ε máximo. Si el punto dado no es central para el ε máximo, su distancia al núcleo es indefinida.*

$$distNucleo_{\varepsilon,MinPts} = \begin{cases} INDEFINIDO & si |N_\varepsilon(p)| < MinPts \\ Min\ dist\ para\ N_\varepsilon(p)| \geq MinPts \end{cases} \tag{8.5}$$

Definición 13

*La **Distancia de Accesibilidad o de Alcance** entre p y q, es el máximo entre la distancia al núcleo de p y la distancia entre p y q. Solo está definida sí p es un punto central.*

$$distAlcance_{\varepsilon,MinPts} = \begin{cases} INDEFINIDO & si |N_\varepsilon(p)| < MinPts \\ Max(distNucleo_{\varepsilon,MinPts}(p), dist(p,q)) \end{cases} \tag{8.6}$$

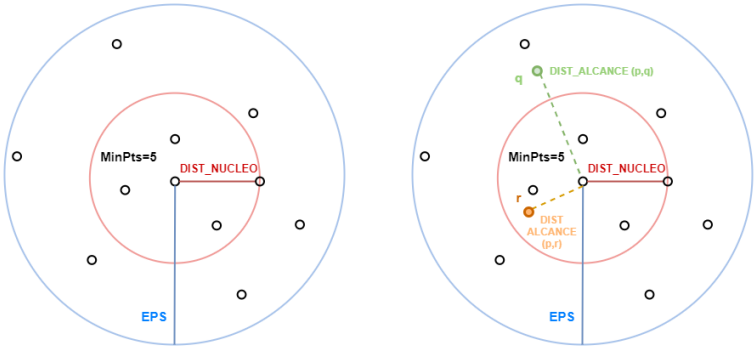


Figura 8.4: Distancia al núcleo y distancia de alcance

Definición 14

Un punto $p \in \{1, ..., n-1\}$ *se denomina* $\xi -$ *empinado ascendente si es* $\xi\,\%$ *más bajo que su sucesor.*

$$PuntoAscendente_{\xi}(p) \Leftrightarrow r(p) \leq r(p+1) \times (1-\xi) \tag{8.7}$$

De forma análoga se denomina $\xi -$ *empinado descendente si es* $\xi\,\%$ *más alto que su sucesor.*

$$PuntoDescendente_{\xi}(p) \Leftrightarrow r(p) \times (1-\xi) \geq r(p+1) \tag{8.8}$$

Definición 15

Un intervalo $I = [i, f]$ *se denomina área* $\xi -$ *empinada ascendente si cumple:*

- $PuntoAscendente_{\xi}(i)$ *y* $PuntoAscendente_{\xi}(f)$.
- *Cada punto entre* i *y* f *es tan alto o más que su predecesor* $\forall x, i < x \leq e : r(x) \geq r(x-1)$.
- I *no contiene más de* $minPts$ *consecutivos que no son* $\xi -$ *empinados ascendentes* $\forall [i', f'] \subseteq [i, f] : ((\forall x \in [i', f'] : \neg PuntoAscendente_{\xi}(p)) \Rightarrow f' - i' < minPts)$.
- I *es máxima* $\forall J : (I \subseteq J, areaAscendente_{\xi}(J) \Rightarrow I = J)$.

Esta definición es análoga para área $\xi -$ *empinada descendente*

Definición 16

Se denomina ξ***-Cluster*** *a* $C = [i_C, f_C] \subseteq [1, n]$ *si* $\exists D = [i_D, f_D], A = [i_A, f_A]$ *y satisface las siguientes condiciones:*

1. $AreaDescendente_\xi(D) \wedge i_C \in D$*, El inicio del cluster es una área descendente que está contenido en D.*
2. $AreaAscendente_\xi(A) \wedge f_C \in A$*, El final del cluster es una área ascendente que está contenida en A.*
3. *Contiene al menos minPts y dada la función r que contiene la distancia de alcance los objetos deben ser al menos x % más bajos que el primer punto de B y que el siguiente después del fin.*
   (a) $f_C - i_C \geq minPts$
   (b) $\forall x, i_D < x < f_D : (r(x) \leq min(r(i_D), r(f_A)) \times (1 - \xi))$

Dadas estas condiciones se establece i_C *y* f_C *como:*

$$(i_C, f_C) = \begin{cases} (max\{x \in D | r(x) > r(f_A + 1)\}, f_A)\} & \text{sí } r(i_D) \times (1 - \xi) \geq r(f_A + 1) \\ (i_D, min\{x \in A | r(x) > r(i_D)\})\} & \text{sí } r(f_A + 1) \times (1 - \xi) \geq r(i_D) \\ (i_D, f_A) & \text{Resto de casos} \end{cases} \tag{8.9}$$

Con esta definición y con el diagrama de la distancia de alcance se puede observar que el primer objeto del cluster es el último objeto con un valor alto de accesibilidad, mientras que el último objeto es el último con un valor bajo de accesibilidad.

Implementación

Para hallar los clusters se recorre el diagrama de alcance y se utiliza el concepto de valores máximos intermedios (mib). Los valores mib representan el valor máximo entre un cierto

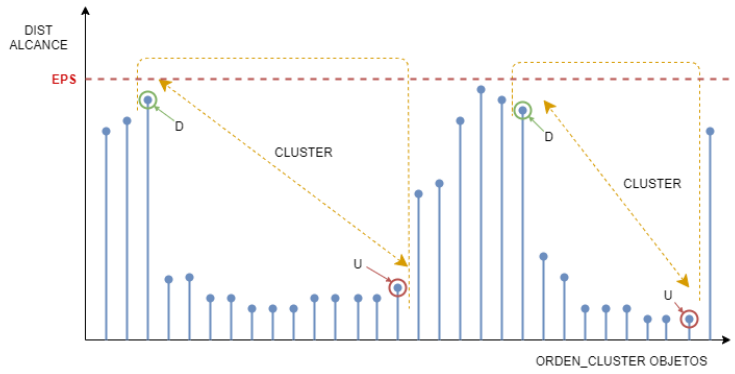


Figura 8.5: Diagrama Alcance y extracción de clusters

punto y el índice actual. Se utiliza un mib Global que es el máximo entre el final del último área empinada y el índice actual y un mib por cada región descendente que es el máximo entre el final de la región y el índice.

Algoritmo 4: expandirCluster

Datos: D Conjunto de datos a agrupar, **dato** elemento a ordenar,
Resultado: D actualizado

```
1  Inicio;
2  vecinosDato=obtenerRegion(dato,D);
3  dato.ordenProcesamiento=marcarProcesado(dato,D);
4  dato.distAlcance=-1;
5  dato.distNucleo=actualizarDistNucleo(vecinos,dato,D);
6  si NOT dato.esPuntoCentral() entonces
7  |  semillas=actualizarSemillasOrden(dato,vecinos,D) ; // actualiza la lista
   |   ordenada de vecinos en función de la distancia de alcance (de
   |   menor a mayor)
8  |  mientras NOT semillas.estaVacia() hacer
9  |  |  semillaActual=semillas[0];
10 |  |  vecinosSem=obtenerRegion(semillaActual,D);
11 |  |  semillaActual.procesado=marcarProcesado(semillaActual,D);
12 |  |  semillaActual.distNucleo=actualizarDistNucleo(vecinosSem,semillaActual,D);
   |  |
13 |  |  si NOT semillaActual.esPuntoCentral() entonces
14 |  |  |  semillasAux=actualizarSemillasOrden(semillaActual,vecinosSem,D) ;
15 |  |  |  semillas.actualizarSemillas(semillasAux) ;
16 |  |  fin
17 |  fin
18 fin
```

Algoritmo 5: extraccionCluster

```
   Datos: D Conjunto de datos a agrupar, Xi porcentaje para punto empinado,
   Resultado: grupos
 1 Inicio;
 2 orden,distAlcanceOrd=obtenerAlcanceOrdenado(D);
 3 mibInterAreaEmpDes=[] ;        // mib de cada area empinada descendente
 4 conjAreaEmpDes=[] ;           // conjunto de areas empinada descendente
 5 conjuntoCluster=[] ;
 6 indice=0 ;
 7 mibGlobal=0 ;      // máximo entre el fin de la última región empinada
    (descendente o ascendente) y el índice
 8 mientras indice<tamaño(D) hacer
 9 |  mibGlobal=max(mibGlobal,distAlcanceOrd[índice]) ;
10 |  areaDescendente=obtenerAreaEmpDes(distAlcanceOrd,indice,Xi) ;
11 |  areaAscendente=obtenerAreaEmpAsc(distAlcanceOrd,indice,Xi) ;
12 |  si areaDescendente != NULL entonces
13 |  |  si tamaño(mibInterAreaEmpDes)>0 entonces
14 |  |  |  mibInterAreaEmpDes,conjAreaEmpDes= actualizarMibInter(...) ;
15 |  |  fin
16 |  |  indice=areaDescendente[-1] ;     // fin de area no se incluye en el
   |  |   cluster,es el indice siguiente
17 |  |  conjAreaEmpDes.añadir(areaDescendente) ;
18 |  |  mibGlobal=distAlcanceOrd[indice] ;
19 |  fin
20 |  si no, si areaAscendente != NULL entonces
21 |  |  mibInterAreaEmpDes,conjAreaEmpDes= actualizarMibInter(...) ;
22 |  |  indice=areaAscendente[-1] ;
23 |  |  mibGlobal=distAlcanceOrd[indice] ;
24 |  |  para indDesc en tamaño(conjAreaEmpDes) hacer
25 |  |  |  areaD=conjAreaEmpDes[indDesc] ;
26 |  |  |  mibD=mibInterAreaEmpDes[indDesc];
27 |  |  |  clust=obtenerCluster(distAlcanceOrd,mibD,areaD,areaAscendente,Xi);
28 |  |  |  si clust != NULL entonces
29 |  |  |  |  conjuntoCluster.añadir(clust) ;
30 |  |  |  fin
31 |  |  fin
32 |  fin
33 |  en otro caso
34 |  |  indice+=1
35 |  fin
36 fin
```

OPTICS con Sklearn

Parámetro	Descripción
max_eps	Máxima distancia para que dos elementos sean vecinos, en OPTICS por defecto es infinito para poder detectar todos los grupos posibles.
min_samples	número de muestras que deben de estar cerca de un elemento para que se considere un punto central.
metric	Métrica de distancia utilizada, por defecto es la distancia euclidiana pero se puede usar otra función de distancia o indicar que la distancia ha sido calculada previamente pasando como parámetro "precomputed". Si es "precomputed", se pasará al algoritmo una matriz de distancia en vez del conjunto de datos.
cluster_method	Indica cómo extraer los clusters una vez ordenado los grupos por defecto se usa xi pero se puede emplear DBSCAN.
eps	Si se indica cluster_method como DBSCAN se emplea como ε y se realiza un clustering siguiendo el algoritmo DBSCAN con eps como parámetro.
xi	Pendiente mínima para ser un límite del cluster.
min_cluster_size	Número de elementos mínimo para componer un cluster (puede ser un entero o una fracción).

Los atributos de OPTICS son los mismos que DBSCAN incluyendo *core_sample_indices_* y *labels_*. De la misma forma que DBSCAN, a través de metric se puede aplicar OPTICS con otras medidas de distancia y con series de tiempo de diferente longitud.

8.3 Clustering Jerárquico

8.3.1 Aglomerativo

El otro algoritmo que se ha aplicado es el clustering aglomerativo con los enlaces completo, simple, promedio y Ward que se explicaron en el tema 6.

Clustering Aglomerativo con sklearn

Aunque el algoritmo no permite usar DTW si permite calcular una matriz de distancia usando DTW y pasarla al algoritmo estableciendo como parámetro de affinity="precomputed".

Parámetro	Descripción
n_clusters	Número de clusters o None en caso de usar el umbral de distancia para determinar las agrupaciones.
affinity	Indica la métrica utilizada, se puede pasar como parámetro "precomputed"para usar una matriz de distancia como entrada del algoritmo. La métrica por defecto es euclidiana
linkage	Enlace utilizado para determinar los grupos a fusionar puede ser çomplete", "single", .ªverage, o "ward".
distance_threshold	Umbral de distancia del enlace, determina el límite por encima del cual se dejan de fusionar los grupos, gráficamente en el dendograma los cortes que produce el límite determinan los grupos. Si se establece distance_threshold el n_clusters debe ser None.

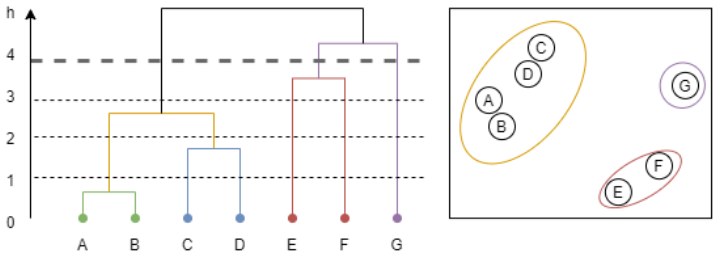


Figura 8.6: Umbral de distancia para determinar los grupos

Capítulo 9

Casos de estudio: Fresadoras

Queremos poder predecir cuándo fallará la máquina, detectando cuando presenta un comportamiento inusual. Sin embargo hay que tener en cuenta que no todos los valores anómalos se corresponde con un fallo de la maquina y puede deberse a múltiples motivos. Algunas razones pueden ser un cambio de pieza, la realización de una parte especifica o el tiempo que lleve funcionando la máquina (ya que presenta una mayor vibración al momento de arrancar).

Se deberá tener especial consideración en la detección de falsos positivos; que se indique va a fallar la herramienta cuando no es así implica un coste extra ya que implicaría un cambio o una reparación innecesaria así como el gasto en personal que se encargue de solucionarlo. También se deberá considerar a los falsos negativos que obvien una rotura en la herramienta con los perjuicios que ello conlleva tanto económico como en tiempo. Tal como se menciona en [70] prevenir falsos positivos es más importante que captura todas las anomalías.

Además hay que tener en cuenta que no todas las fallos de la herramienta están precedidas por anomalías.

9.1 Caso de Estudio: Fresadora CNC (CIDAUT)

Este trabajo intenta detectar patrones comunes e inusuales del comportamiento de la fresadora a través de sus sensores. Dado el fallo del acelerómetro se usará solo la corriente para el clustering, dejando para un futuro la posibilidad de utilizar ambas variables.

9.1.1 Análisis de los Datos

Análisis inicial de los datos

Se han obtenido datos del funcionamiento de la fresadora CNC de 28 días diferentes, 20 de esos días corresponden al mes de enero y los 8 restantes al mes de febrero.

Los datos recogidos se dividen en 21 campos de los cuales 10 no aportan información, los campos CNC_vallado, CNC_Inicio_Prog, CNC_Fin_Prog, CNC_Planeado, CNC_-Contonear, CNC_Fresar, CNC_Taladrar y CNC_Paro_Emergencia contienen todos el mismo valor 0 en todas las muestras estudiadas, de la misma forma CNC_Trigger_-Marcha y Bias tampoco aportan valor al tener en todas las muestras valores fijos de Null y -60 respectivamente.

Los 11 campos restantes son CNC_puerta que indica mediante un valor booleano cuando la puerta de la fresadora está abierta y las otras 10 son mediciones que realiza la fresadora sobre su estado, en estas mediciones se incluye la temperatura, el sonido, el acelerómetro en sus 3 ejes y la corriente también en sus tres fases.

La fresadora solo podrá estar realizando alguna operación de fresado cuando la puerta se encuentre cerrada (con valor 0 en ese parámetro).

Las muestras han sido tomadas irregularmente, aunque aproximadamente se toman muestras cada segundo.

Análisis de la Corriente

De las 4 características sobre las que se realizan las mediciones la más representativa es la corriente, sin embargo de los 28 días examinados solo hay recogidos datos de corriente de 14 días.

La forma de la gráfica para las distintas fases de la corriente es muy similar (aunque las mediciones de la fase 2 son ligeramente inferiores) por ello sólo se examinará la fase 1.

De la gráfica 9.1 se puede observar que la fresadora funciona entre las 8 a.m y las 17 p.m y que presenta numerosos picos que llegan hasta cerca de los 30 A, así como una gran densidad de mediciones en torno a 2.5 A y valores más pequeños al inicio y fin del periodo de funcionamiento que requerirán un análisis más en profundidad.

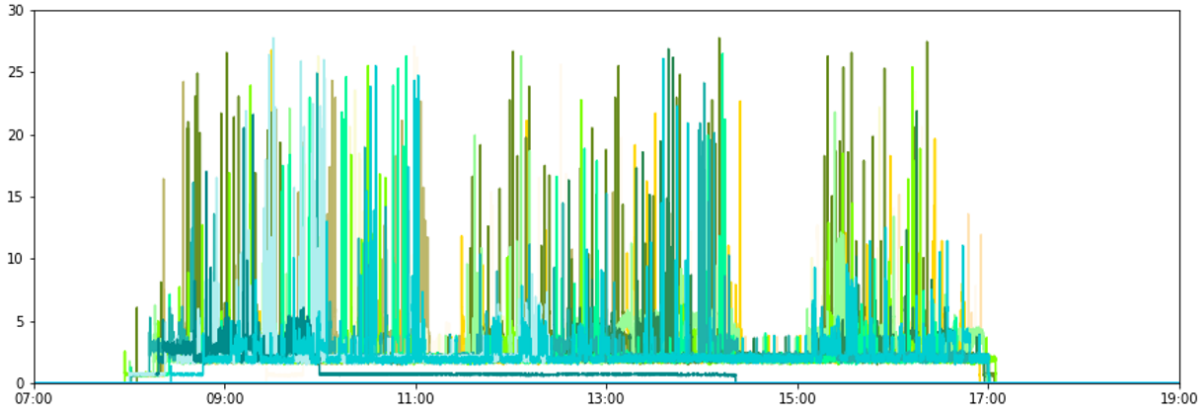


Figura 9.1: Corriente todos los días examinados

También se puede observar períodos de menor actividad donde no existen picos y no se superan los 5 A, el primero de estos periodos empieza a las 11 a.m y tiene una duración aproximada de media hora, el otro periodo empieza a las 2 p.m con una duración mayor que llega a alcanzar algo más de una hora.

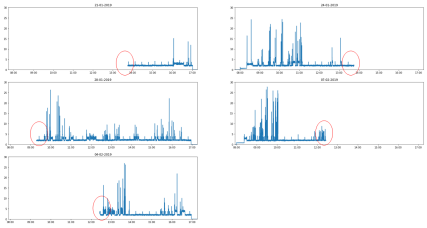


Figura 9.2: Comportamientos inusuales

Examinando de forma individual cada uno de los 14 días encontramos varios que presentan diferentes comportamientos anómalos o parciales.

Arranque y Calentamiento de la máquina Observando las señales que presentan un patrón similar en los instantes posteriores a su arranque, podemos sacar una serie de conclusiones. El arranque de la máquina se produce entre las 7:55 a.m y 8:05 a.m y tiene una fase de calentamiento que se prolonga entre 15 y 20 min. En esta fase de arranque se observa un pico más o menos pronunciado justo al arrancar la máquina seguido de un espacio de tiempo donde la corriente se mantiene a una medida similar a 1 A. Después se producen dos nuevos picos muy seguidos para después estabilizarse entre los 2 y 3 A durante el resto del funcionamiento de la máquina. En las mediciones se observa que después de estos dos picos la corriente se estabiliza a una corriente ligeramente superior durante unos minutos para finalmente estabilizarse entre los 2 y 3 A. Justo antes de descender se produce un ligero incremento durante un breve periodo de tiempo.

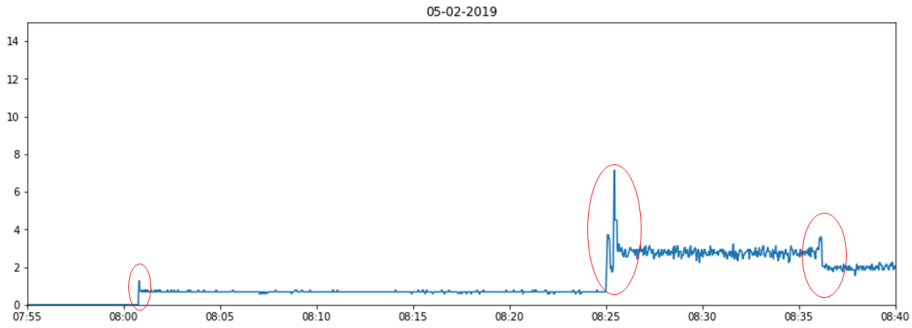


Figura 9.4: Corriente 05-02 Forma habitual de la corriente durante su arranque y calentamiento

El día 13 de febrero se observa un retraso de cerca de 25 minutos en el arranque de la máquina aunque el funcionamiento el resto del día parece el habitual; esto mismo se observa el día 22 de enero con un retraso superior alcanzando los 85 minutos. Además el día 13 de febrero ocurre otra peculiaridad en su calentamiento, tras los dos picos la corriente se estabiliza a una corriente ligeramente inferior a la que posteriormente se mantiene en el resto de su funcionamiento.

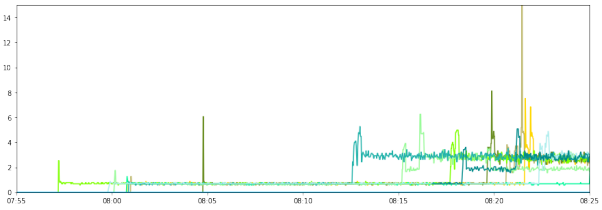


Figura 9.3: Corriente en el arranque de la fresadora

Máquina en funcionamiento Además de lo que se observa en la gráfica con la corriente de todos los días, examinando las horas centrales de cada uno de los días de forma individual se observan varios patrones de comportamientos que se repiten.

Como se veía en la gráfica 9.1, existen picos de corriente que llegan a alcanzar hasta casi los 30 A durante todas las horas centrales en distintos momentos que no parecen seguir una pauta y que seguramente tengan que ver con su uso habitual, las piezas, materiales y otras características propias de la fresadora.

Durante su uso habitual se observa un ligero incremento de la corriente de forma periódica en el tiempo (aproximadamente cada 20 minutos). Es más apreciable en periodos donde no presenta incrementos la corriente y se mantiene al valor donde se estabiliza, cerca de los 2 A.

Sobre los picos se observa dos comportamientos típicos. El primer comportamiento muestra que después del pico la corriente desciende a un valor superior al valor estable o base y se mantiene a ese valor un periodo de tiempo variable que suele estar en torno a los 2 o 4 minutos antes de volver al valor estable. En el otro comportamiento antes de volver al valor base se produce un segundo pico.

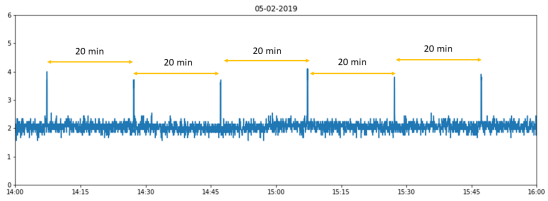


Figura 9.5: Corriente Incrementos pequeños de corriente a periodos constantes de tiempo

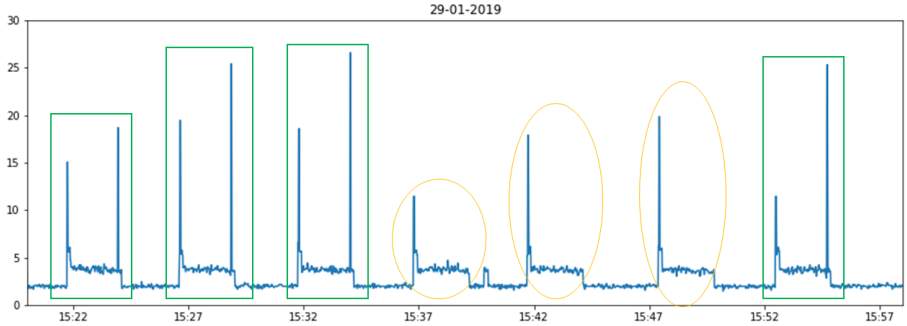


Figura 9.6: Corriente 29-01 Picos de Corriente Habituales

Horas de descanso Se producen dos descansos donde no se producen incrementos destacables en la corriente. Si se dan los ligeros incrementos que se producen cada 20 minutos. El primer descanso se produce sobre las 11 a.m y dura aproximadamente media hora, aunque este descanso puede estar un poco desplazado en los datos observados no se supera los 5 A ningún día entre las 11:08 y 11:28. El otro descanso se produce cerca de las 2:20 p.m mostrando el mismo comportamiento que en el otro descanso pero prolongado más tiempo, hasta las 15:06 en ninguno de los datos examinados se vuelven a producir picos. Estos descansos son claramente visibles en la figura 9.1.

Apagado de la máquina El apagado de la máquina se realiza alrededor de las 17 p.m y se observan dos comportamientos distintos aunque similares. La máquina suele presentar un tiempo breve sin incrementos antes del apagado que suele ser como mínimo de unos 5 minutos. Antes del apagado suele reducirse el nivel base de corriente, a cerca de 1 A y permanece así un tiempo antes de caer completamente a cero. Este comportamiento varía

en varios de los casos donde aunque desciende brevemente la corriente a menos de 1 A cae rápidamente a cero sin mantenerse un tiempo constante a ese nivel.

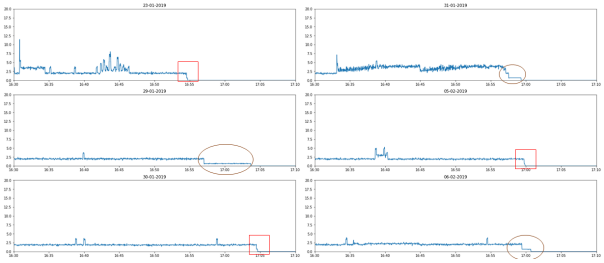


Figura 9.7: Corriente diversos días Final

Acelerómetro

Los valores del acelerómetro se encuentran en torno a -4 y 4 hz y al igual que pasaba con las corrientes las mediciones en los 3 ejes son bastantes similares. Aunque en el eje Z es considerablemente inferior como se observa en la figura 9.8. Como en el caso de la corriente para el Acelerómetro examinaré sólo uno de sus ejes, en este caso el eje X.

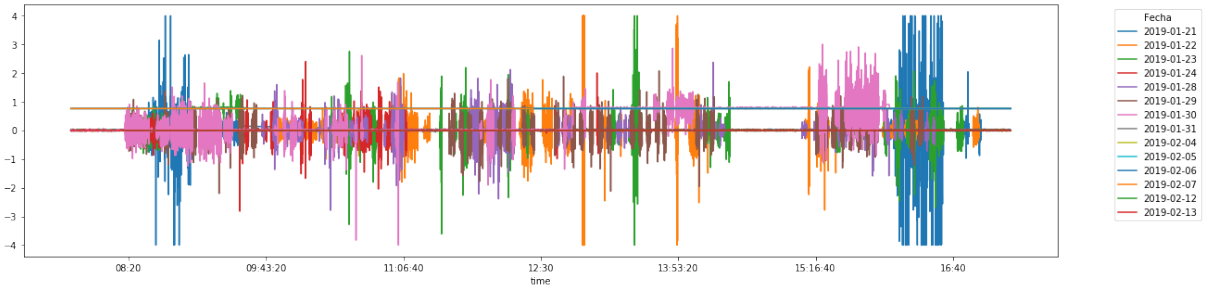


Figura 9.8: Acelerómetro

Con el acelerómetro también se puede apreciar el inicio y el fin de la fresadora así como las horas de descanso. En estas medidas se ve claramente dos comportamientos muy diferenciados debido a una rotura del acelerómetro que se produjo a finales de enero y que afectó a todas las mediciones del mes de febrero. Tras la rotura solo muestra un valor constante y esos datos no sirven para la detección de anomalías.

Habitualmente el acelerómetro se mantiene a un valor base de 0 hz que durante la fabricación de las piezas oscila a valores positivos y negativos , estos momentos coinciden con aquellos en que la corriente registra picos y por tanto al igual que pasaba con los picos de corriente no parecen seguir un patrón.

La rotura del acelerómetro se produjo el día 30 de enero cerca de la 13 de la tarde como se ve en la figura 9.9. Se ve que el valor base cambia, en los ejes X e Y pasa a estar en 0.8 hz y en el eje Z a -0.9 hz aunque ese día el acelerómetro sigue registrando la actividad de la fresadora mostrando las oscilaciones que presenta en el momento de fabricar las piezas. El resto de días no registra las oscilaciones y se mantiene constante a los valores fijos que adoptó después de la rotura.

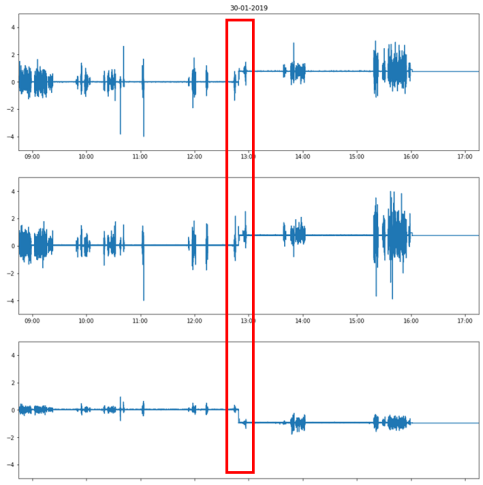


Figura 9.9: Acelerómetro rotura del 30 de Enero

Se observa un caso anómalo donde el valor base se mantuvo más elevado durante aproximadamente media hora cerca de las 9 a.m del 23 de enero.

Conclusiones y observaciones conjuntas

Observando las mediciones en conjunto por días se observa como ya se mencionó que las oscilaciones del acelerómetro coinciden con los picos de la corriente y con un aumento del valor mínimo del sonido en esos momentos estando cerca de los 40 db en vez de los 30db.

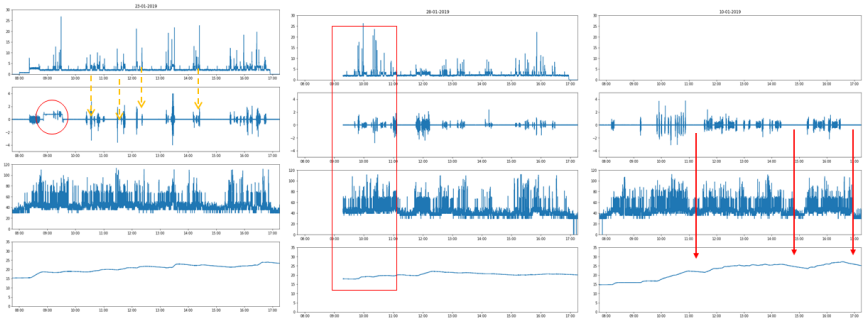


Figura 9.10: Corriente, Acelerómetro, Sonido y Temperatura de los días 10, 23 y 28 de enero

La temperatura también está relacionada con el resto de medidas, los incrementos en la temperatura coinciden con los picos de corriente y sus bajadas con periodos de inactividad.

9.1.2 Preparación y limpieza de datos

Para aplicar el clustering en los datos de corriente nos enfrentamos a dos problemas:

- Los datos presentan un muestreo irregular y la mayoría de métodos requieren que los datos de entrada tengan un muestreo regular para poderlos aplicar.
- No hay información que indique cómo segmentar las series diarias para poder identificar el arranque, el apagado, patrones de funcionamiento...

Muestreo de los datos

Para solucionar el inconveniente del muestreo irregular se ha optado por crear un método que permita obtener una representación PAA para representar los datos como el valor medio de las muestras tomadas en un intervalo de tiempo fijo. En este caso he aplicado intervalos cada 5 segundos (el valor de la corriente a las 10:45:05 se corresponde con la media de valores que tomó la corriente entre las 10:45:05 y las 10:45:10).

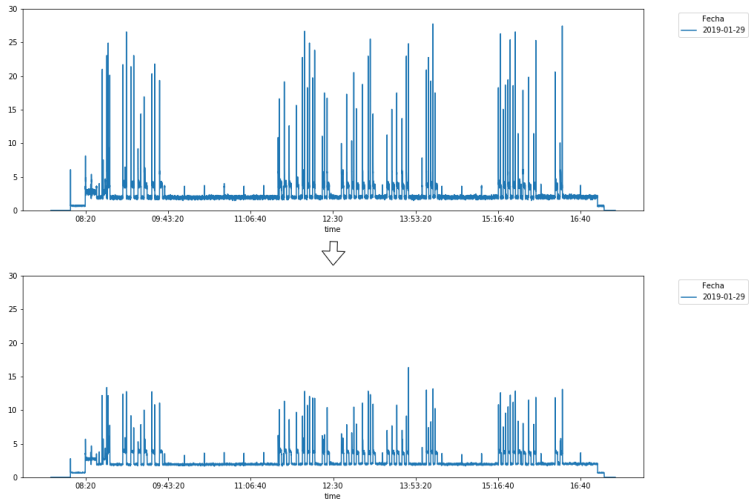


Figura 9.11: Muestreo PAA en fresadora CIDAUT

Como se puede ver en 9.11 la forma de la señal se mantiene bastante bien y además no solo solventa el problema del muestreo irregular sino que permite reducir el tamaño de las muestras pasando a obtener 6839 muestras de cada uno de los días lo que supone una reducción de más del 50 % en cualquiera de los días (los datos originales presentaban entre 18884 y 82674 muestras diarias).

Sin embargo no hay que olvidar los inconvenientes de este método. A parte de la pérdida de información, nos encontramos a que tras aplicar este muestreo los datos se suavizan lo que puede ser un inconveniente para la detección de anomalías, pudiendo ayudar a enmascarar valores anómalos.

Para el caso en que en un intervalo no existan mediciones se han tomado varias estrategias:

- El muestreo PAA se ha realizado entre las 7:45 y las 17:15 para dejar algo de margen ante posibles valores anómalos, sin embargo el arranque y apagado se encuentra por lo general a las 08:00 y a las 17:00 por lo que si el intervalo de mediciones se encuentra antes del arranque o después del apagado se establece que el valor de ese intervalo es 0.
- En el caso de que el valor nulo se encuentre en las horas centrales se encuentra las muestras anterior y posterior más cercanas que no tienen valor nulo y se establece un intervalo de 5 muestras a cada lado y se asigna el valor de la moda.

Segmentación

Para dividir la corriente en segmentos que permitan descubrir patrones se probó varias formas:

- Utilizando el parámetro puerta: Se realizó una inspección visual de la corriente diferenciando cuando la puerta de la fresadora está abierta o cerrada (se sabe que la fresadora solo puede estar realizando una pieza cuando la puerta se encuentra cerrada). Sin embargo los segmentos obtenidos de esta forma no parecen permitir visualmente diferenciar si la fresadora está trabajando o no y comportamientos aparentemente similares presentan valores distintos de este parámetro, la segmentación obtenida no parece ser buena. Además hay muchas características de este parámetro que no se conocen: no se sabe si el parámetro cambia instantáneamente cuando se cierra la puerta, ni si se podrían realizar varias piezas seguidas sin necesidad de abrir

la puerta para sacar la que se acaba de realizar y tampoco si la puerta se puede quedar cerrada después de terminar la elaboración.

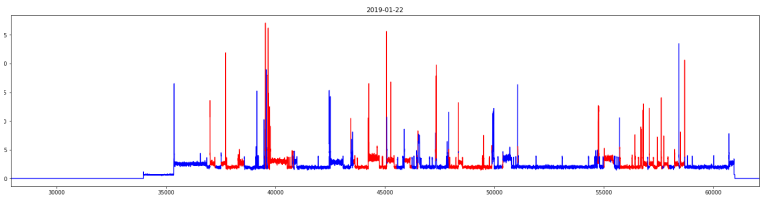


Figura 9.12: Segmentación Puerta

- Segmentación por aproximación de mínimos cuadrados a polinomio de grado 1: Se creó un método de ventana que va calculando la recta a la que se aproximan los datos a través del método de mínimos cuadrados discretos. La ventana va creciendo e incrementando el número de datos y recalculando la recta obtenida hasta que la recta supera un error que marca el final del segmento (se ha implementado para error medio, máximo y cuadrático aunque la segmentación se ha probado con medio). El método segmenta a parte los datos anteriores al arranque y después del apagado y empieza la segmentación cuando la corriente empieza a no ser 0. Se realizó una modificación del método para usar una aproximación a un polinomio de grado superior a 1 también usando un método de mínimos cuadrados. Las fórmulas utilizadas para hallar los coeficientes de las rectas son:

$$a_0 = \frac{n\sum_{i=1}^{n} x_i^2 \cdot \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} x_i \cdot y_i \cdot \sum_{i=1}^{n} x_i}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2} \qquad a_1 = \frac{\sum_{i=1}^{n} x_i \cdot y_i - \sum_{i=1}^{n} x_i \cdot \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2} \tag{9.1}$$

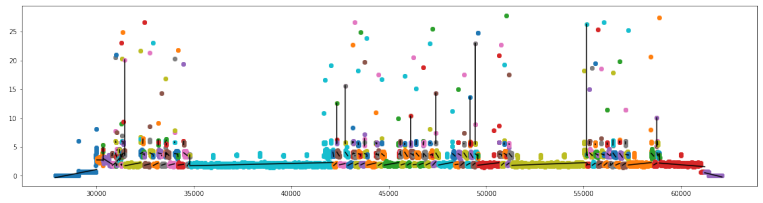


Figura 9.13: Segmentación Aproximación Lineal

- Método Propio Segmentación CB: Tras observar el comportamiento diario de la corriente se comprobó que la corriente presenta un valor estable al que regresa tras cada incremento de la misma; a este valor, o más bien al intervalo de valores, lo he llamado corriente base. A través de este valor puede dividir la señal de corriente en varias etapas:

- Arranque: Desde que la corriente está a 0 hasta que se estabiliza a la corriente base.
- Funcionamiento: La corriente supera el valor de la corriente base y supongo que es cuando la fresadora realiza alguna operación.
- Inactivo: La corriente se encuentra entre el rango de valores de la corriente base, supongo que en este caso la fresadora se encuentra encendida pero no está realizando ninguna operación.
- Apagado: La corriente se encuentra a un valor menor que la corriente base y posteriormente toma el valor 0, la corriente solo baja del límite inferior de la corriente base cuando se está apagando.

La idea que se ha utilizado es un método de ventana. Primero se identifica la fase del segmento y el segmento continúa hasta que deja de cumplir las condiciones de la fase identificada (se utiliza un parámetro de mínimo de puntos no aislados para evitar que se identifique como el final de un segmento por un dato anómalo si supera ese umbral se considera que esos datos son de otro segmento). Esta segmentación requiere definir

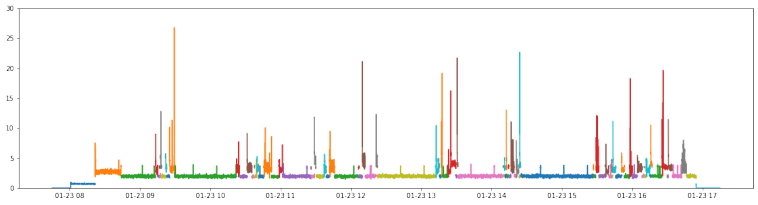


Figura 9.14: Segmentación propia CB

los límites superior e inferior de la corriente base. Se probaron distintas formas para hallarlo. Se probó a través de la segmentación por aproximación lineal escogiendo los segmentos más largos (que supongo que serán los del mayor descanso), con los que tienen la pendiente más similar a 0 y escogiendo los valores mínimos y máximos usando una recta de regresión o la media del segmento ± su desviación típica.

A través del tercer método de segmentación consigo dividir los datos en secuencias de tamaño irregular y hacer una clasificación inicial de estas secuencias en las fases mencionadas (arranque, activa y apagado).

Segmentación del arranque La parte que más se conoce de la fresadora es su arranque y se sabe que consta de las siguientes fases:

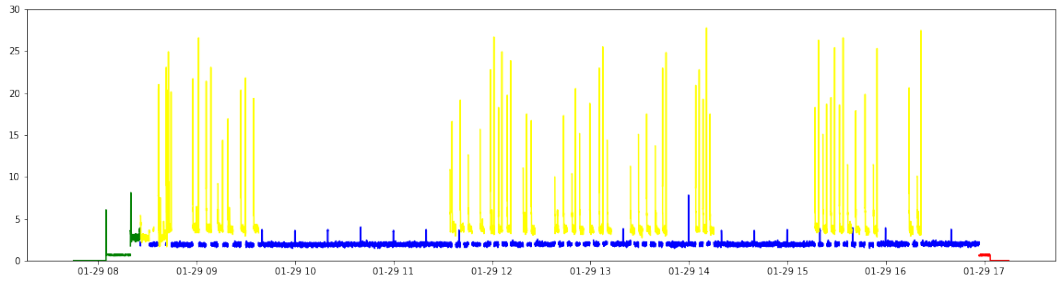


Figura 9.15: Segmentación propia CB

1. Estado de parada (máquina apagada)
2. Pico de arranque
3. Ciclo de trabajo del PLC
4. Arranque del motor
5. Calentamiento del motor
6. Funcionamiento (fin del arranque)

Para poder agrupar las secuencias que se corresponden con cada una de estas frases primero he de segmentar el arranque y posteriormente agrupar estos segmentos.

Con la segmentación CB y una inspección visual de los resultados parece identificar de forma correcta el arranque.

Por ello he utilizado esta segmentación para identificar el arranque y posteriormente seguir otra estrategia para segmentarlo de nuevo en las fases del arranque.

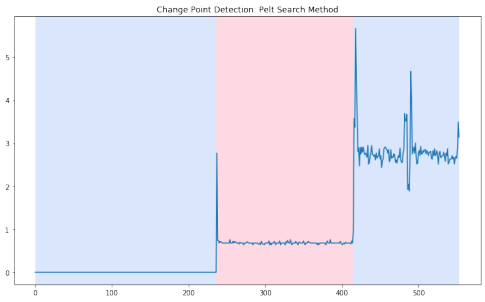


Figura 9.16: Segmentación con Pelt

Para poder identificar las fases del arranque, parto del arranque obtenido con segmentación CB (independientemente de si se identificaron como varios segmentos o como uno). Se aplica sobre los datos de arranque identificados mediante CB otra segmentación empleando el método PELT [58] de la librería Ruptures . PELT es un algoritmo de tiempo lineal exacto podado, se trata de un algoritmo de coste lineal y exacto que consiste en minimizar una función de coste sobre diferentes número de puntos (de la serie) y las posiciones de los mismos. Se utilizó PELT con la desviación absoluta como función de costo para detectar los puntos de corte.

Después de aplicar PELT se emplea un algoritmo creado para identificar picos entre dos segmentos. La idea que sigue es recorrer la lista de segmentos hallados por PELT de dos en

dos segmentos, en el segmento anterior se identifica si hay un pico al final y en el segmento posterior si hay un pico al inicio; en caso de existir se dividen los dos segmentos en tres (para determinar si es pico se comprueba si son valores atípicos del segmento usando el primer y tercer cuartil al que se le resta y se le suma el rango intercuartílico multiplicado por 1.5).

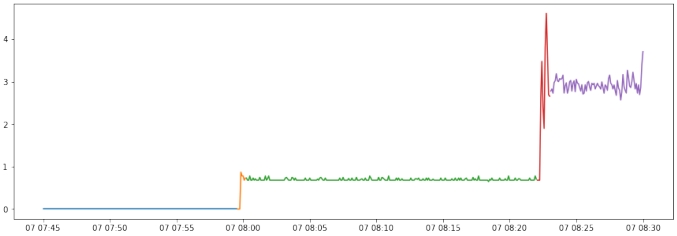


Figura 9.17: Segmentación Arranque

9.1.3 Análisis de los Modelos Utilizados

El comportamiento de la fresadora queda por tanto definido por las siguientes fases 9.18.

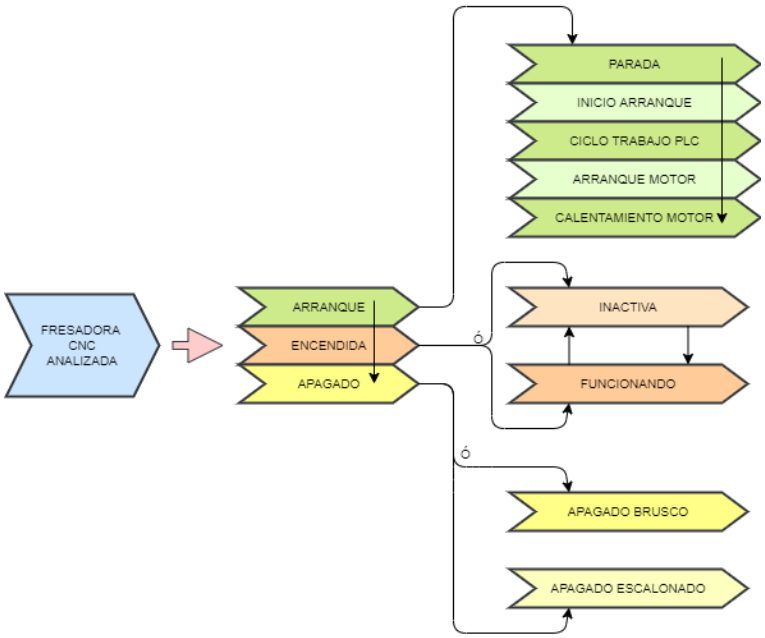


Figura 9.18: Fases Fresadora

Se ha aplicado los algoritmos de clustering DBSCAN, OPTICS, K-MEANS y aglomerativo (con diferentes enlaces) a las fases de los datos de la fresadora obtenidos del muestreo PAA y la segmentación CB.

Los algoritmos DBSCAN, OPTICS y aglomerativo (salvo enlace Ward) se han aplicado tanto con distancia euclidiana como DTW. K-Means y aglomerativo con enlace Ward solo se han aplicado con distancia euclidiana.

En las distintas pruebas primero se ha intentado una agrupación completa (con el conjunto de los segmentos diarios de la fase que se agrupe como una de las 14 entradas, cada día analizado) y luego se agrupan los segmentos de esa fase independientemente del día.

Además del arranque y apagado se ha examinado la fase activa donde se han englobado los segmentos de funcionamiento e inactivo.

Se probó con una representación basada en un vector de características formado por coeficientes de la aproximación a una recta restando al término independiente el valor de inicio (debido a que en este problema no me importa el instante en el que aparece un patrón pudiéndose fabricar la misma pieza a las 11:00 que a las 15:30 por ejemplo) y las medidas de asimetría, media, máximo, mínimo, autocorrelación y el tamaño del segmento. Sin embargo los resultados no mejoran y en lo único que mejora es en el tiempo de ejecución de los algoritmos, siendo mucho más rápido agrupar vectores que series de tiempo, sin embargo al no suponer una mejora no se analizaran los resultados.

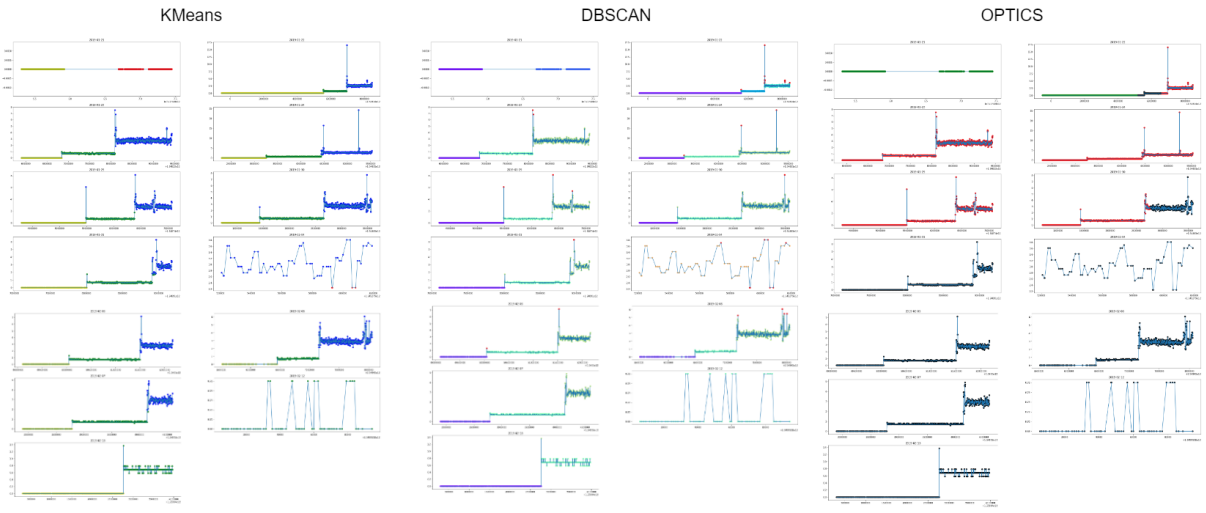


Figura 9.19: Clustering de puntos de serie de tiempo

También se realizaron pruebas de clustering de puntos de series de tiempo en el arranque, los resultados fueron malos aplicando OPTICS pero aceptables en K-Means y en DBSCAN. Los resultados fueron similares con ambos algoritmos, siendo ligeramente mejores en DBSCAN donde se pudo detectar en algunos días los arranque de la fresadora y del

motor al considerarlos como outlier.

Análisis Resultados Clustering Aplicado al Arranque

Basado en Jerárquico Al aplicar el clustering jerárquico al arranque completo, aunque cambie el dendrograma con los diferentes enlaces, la agrupación obtenida en cada uno es la misma. En los 14 días examinados se observa un día anómalo el 22 de enero y 3 grupos con varios días en cada uno. En uno de los grupos se encuentran los arranques donde el pico de arranque es mínimo; en otro los días donde no se aprecia arranque (salvo el 31 de enero que a pesar de agruparse en ese grupo parece tener un comportamiento normal); y otro donde se encuentra los días donde los arranques parecen más típicos y se distingue a simple vista los arranque de la fresadora y del motor.

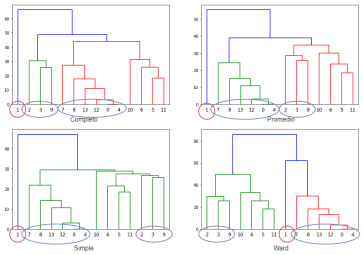


Figura 9.20: Dendrograma Jerárquico Arranque Completo

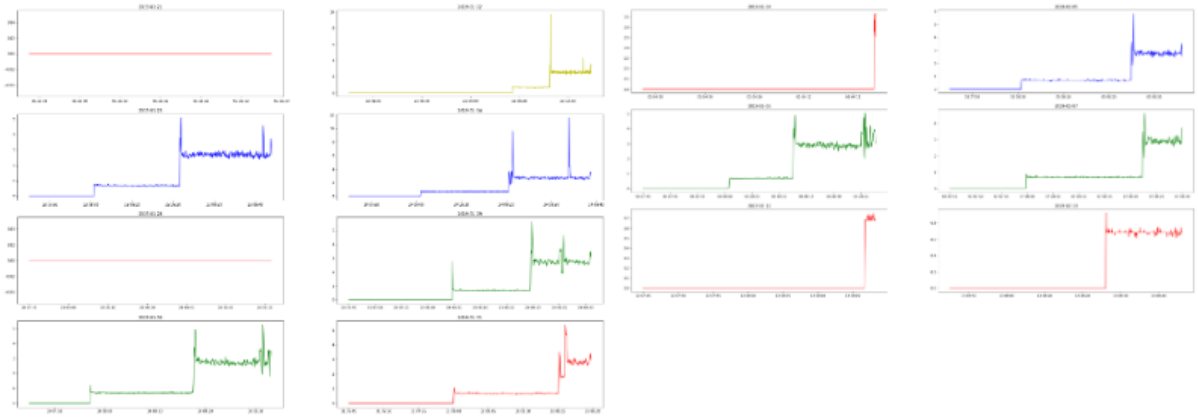


Figura 9.21: Clustering jerárquico del arranque completo

En el clustering de subsecuencias del arranque el mejor resultado obtenido se ha conseguido con enlace completo y distancia DTW donde por lo general se agrupan de forma correcta: la parada, el ciclo de trabajo de PLC, el arranque del motor y el calentamiento. Sin embargo no consigue distinguir entre el PLC y el arranque de la fresadora.

En el resto de clustering con enlace promedio no se consiguió diferenciar entre el arranque del motor y su calentamiento; mientras que con el simple pasa algo similar con las etapas de parada, arranque de la fresadora y ciclo PLC.

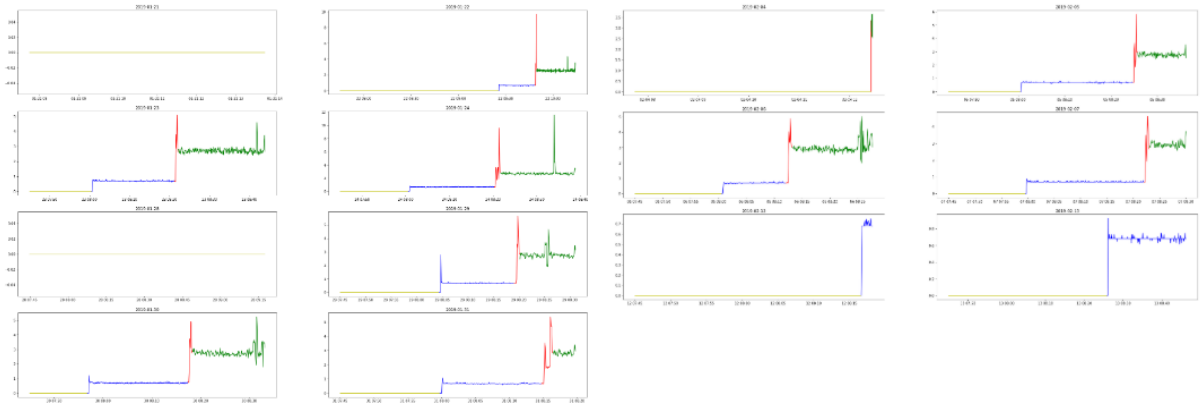


Figura 9.22: Clustering jerárquico enlace completo con distancia DTW

Con la distancia euclidiana suele agrupar en un mismo grupo las fases que no son el calentamiento del motor.

Basado en Partición Aplicando K-Means al arranque completo la agrupación obtenida es la misma que se obtenía en el clustering jerárquico y que se muestra en la figura 9.21. Aplicando K-Means a los segmentos del arranque primero se realizó el diagrama del codo. Observando el diagrama de codo se observa que hay una curva suave entre 2 y 6 y aunque el cambio más brusco se produce con K=2 al conocer que al menos hay 5 grupos (ya que hay 5 fases de arranque) se establece k=6.

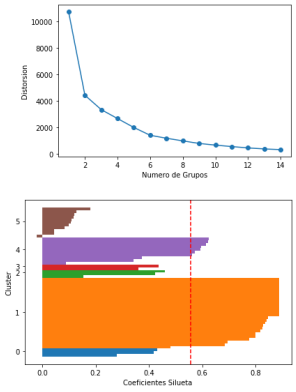


Figura 9.23: Diagrama del codo y silueta

Los resultados muestran, que no se agrupa correctamente el calentamiento del motor y no consigue distinguir el arranque de la fresadora de la máquina apagada; pero es capaz de agrupar el ciclo del PLC y el arranque del motor.

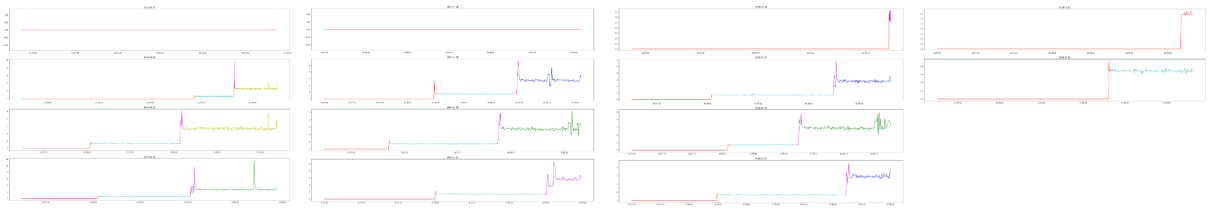


Figura 9.24: Arranque K-means

Basado en Densidad Al aplicar DBSCAN y OPTICS con la distancia DTW sobre el arranque completo el resultado varía. Con DBSCAN hay un grupo más grande donde se encuentran todos los días que presentan un arranque normal, mientras que con OPTICS este grupo se divide en dos grupos uno de ellos con arranques que presentan un pico de arranque de la fresadora más pronunciado y un calentamiento del motor ligeramente más largo que otros.

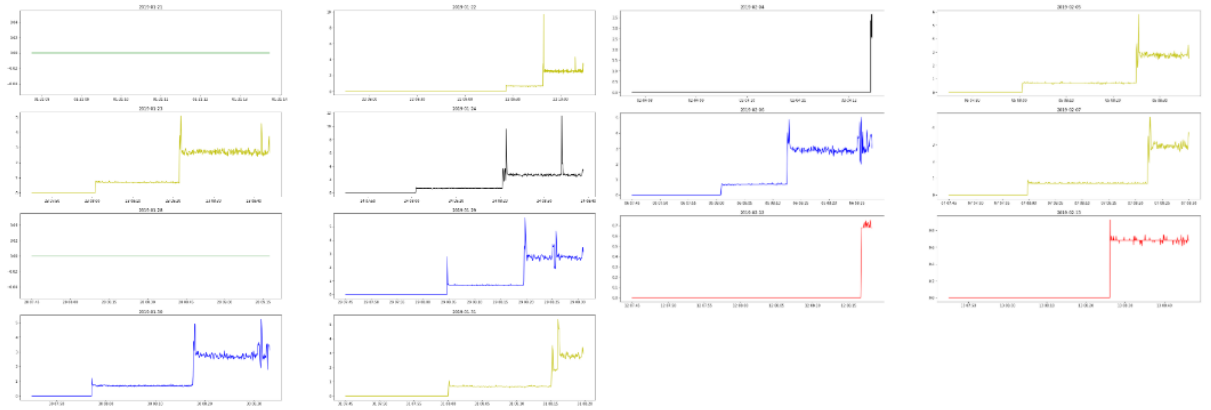


Figura 9.25: OPTICS con DTW aplicado al arranque completo

En el clustering de subsecuencias del arranque al aplicar DBSCAN no se agrupa bien ni el arranque del motor ni el calentamiento calificando ambos como ruido, independientemente de la medida de distancia utilizada. Sin embargo, parece agrupar bien el ciclo del PLC y el arranque de la fresadora (salvo cuando el pico es más pronunciado) con la distancia DTW.

Los resultados con ambas distancias mejoran aplicando OPTICS. Sin embargo OPTICS con la distancia euclidiana sigue dando peores resultados que DBSCAN con DTW. Aplicando OPTICS con la distancia DTW se obtienen buenos resultados y se distinguen las 5 fases del arranque.

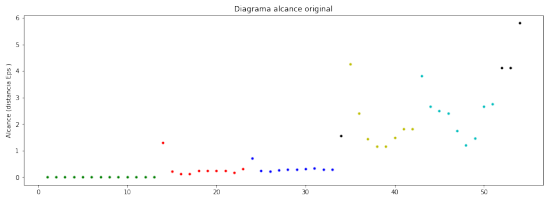


Figura 9.26: Diagrama de alcance arranque

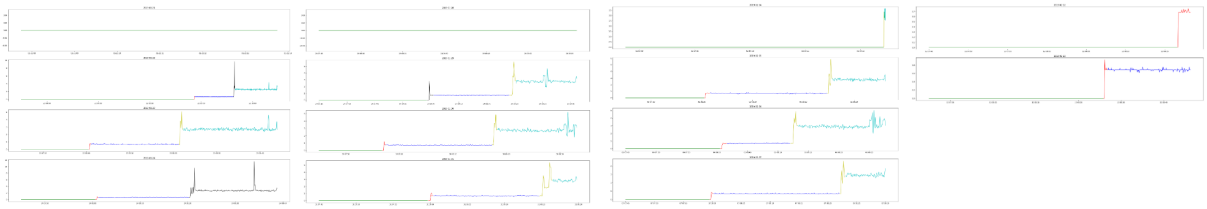


Figura 9.27: OPTICS con DTW aplicado al arranque

Análisis Resultados Clustering Aplicado al Apagado

De cada día solo hay un segmento correspondiente al apagado por lo que solo se ha realizado el clustering del apagado completo.

Con el clustering jerárquico con distancia euclidiana no se identifican distintos tipos de apagado pero si se identifican apagados anómalos (29 y 31 de enero y 12 de febrero), los resultados varían poco según el enlace utilizado (la única diferencia es que Ward agrupa dos anómalos juntos).

Los resultados obtenidos con K-Means son los mismos que se obtuvieron con el clustering jerárquico.

En cambio los algoritmos basados en densidad permiten distinguir diferentes apagados al menos dos apagados distintos uno cuyo apagado empieza por encima de 1 y otro entorno a 0.6. Los resultados entre OPTICS y DBSCAN no varían mucho como se muestra en la figura 9.28.

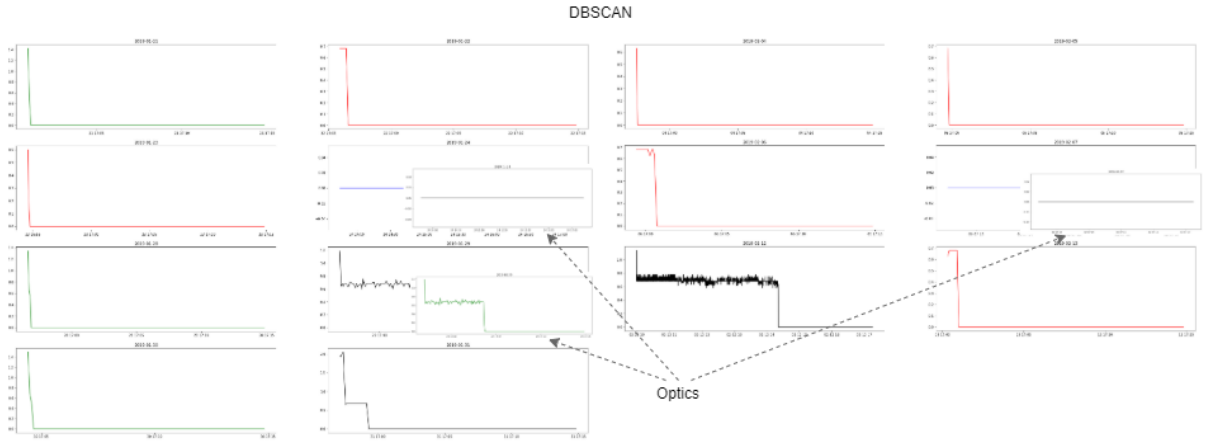


Figura 9.28: DBSCAN y OPTICS aplicado al apagado

Análisis Resultados Clustering Aplicado a la máquina Activa

Basado en Jerárquico

Con distancia euclidiana los resultados no son buenos para ninguno de los enlaces, siendo algo mejores con enlace completo y con enlace Ward, donde se consiguen identificar algunos comportamientos de interés.

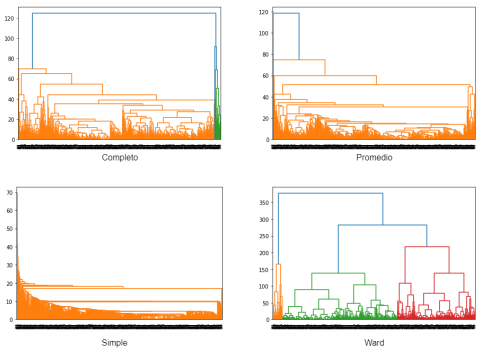


Figura 9.29: Dendrogramas Clustering Jerárquico Euclidiana

- Con enlace completo se consigue identificar un par de comportamientos no habituales y consiguen varios grupos con comportamientos de inactividad.
- Con enlace Ward la distribución de secuencias en los grupos es más homogénea permitiendo distinguir varios comportamientos habituales pero sin identificar patrones únicos.

Los resultados obtenidos con DTW son mejores que los obtenidos con la distancia euclidiana.

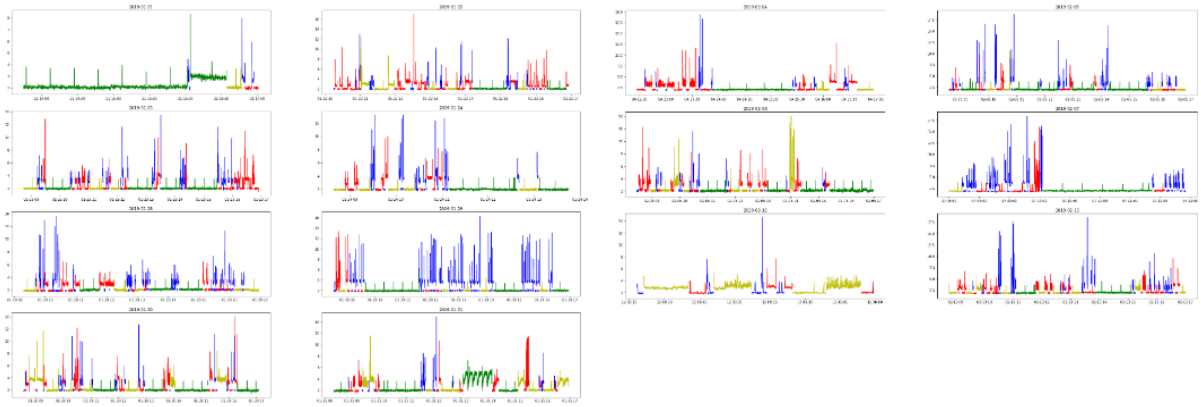


Figura 9.30: Clustering Activa Jerárquico con dist euclidiana y enlace ward

Figura 9.31: Clustering obtenido aplicando clustering jerárquico con enlace completo y distancia DTW a la fresadora en activo

Con enlace simple, aunque agrupa casi todas secuencias en un grupo es capaz de identificar varios comportamientos únicos de forma similar a Ward con distancia euclidiana; esto mejora al utilizar enlace promedio y completo donde no solo se distinguen patrones únicos sino que parece identificar etapas inactivas y de funcionamiento.

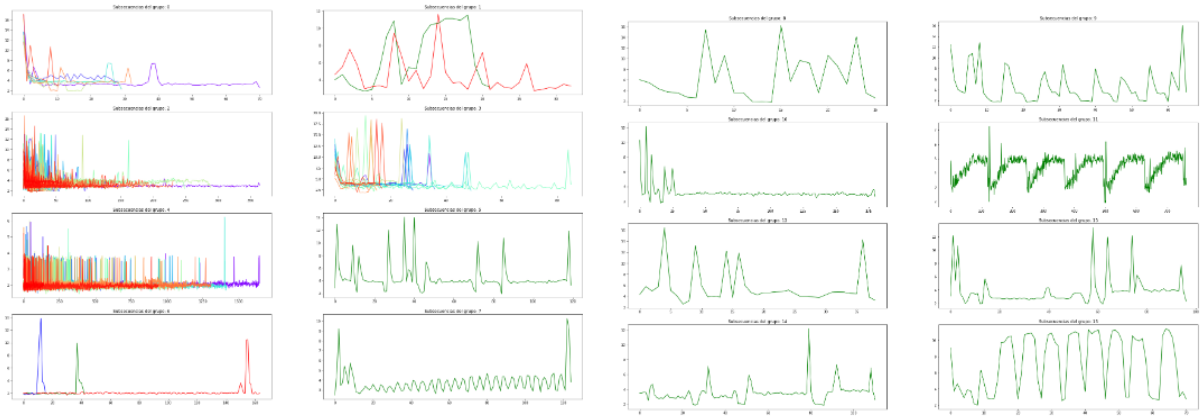


Figura 9.32: Clusters identificados de la fresadora activa aplicando clustering jerárquico con enlace promedio y distancia DTW

Basado en Partición Aplicando K-Means a las secuencias en activa y agrupando los resultados en 3 grupos se identifica un grupo mejor que el resto, donde se encuentra las secuencias correspondientes a partes de funcionamiento con comportamientos habituales y agrupándose en los otros dos grupos secuencias tanto inactivas como patrones inusuales. Además los grupos peores, que muestran peor coeficiente silueta, se observa que contienen elementos mal agrupados presentando valores negativos en el gráfico.

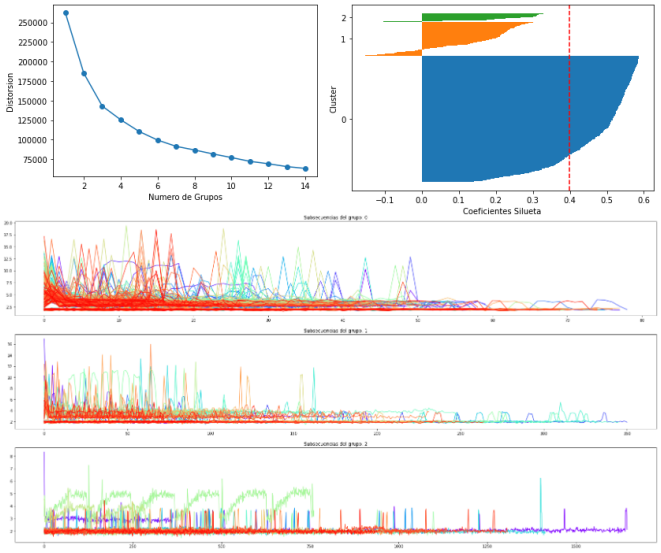


Figura 9.33: Diagramas de codo e índices de silueta y clusters obtenidos con K-means

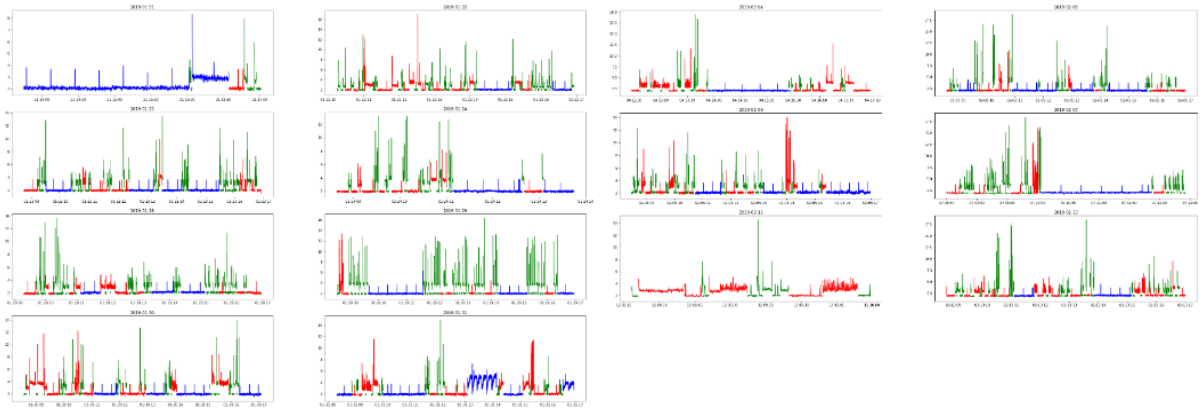


Figura 9.34: Resultado de K-means y distancia euclidiana con la fresadora en activo

Basado en Densidad Aplicando métodos basados en densidad, la mayoría de segmentos se agrupan como ruido y en general los grupos que se obtienen son pequeños. Sin embargo esto hace que se encuentren grupos compuestos por segmentos muy similares entre ellos. De los 4 clustering realizados basados en densidad el que menos ruido ha agrupado es DBSCAN con la distancia DTW.

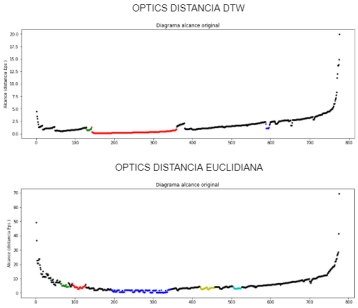


Figura 9.35: Diagramas de alcance obtenido con distancia euclidiana y DTW

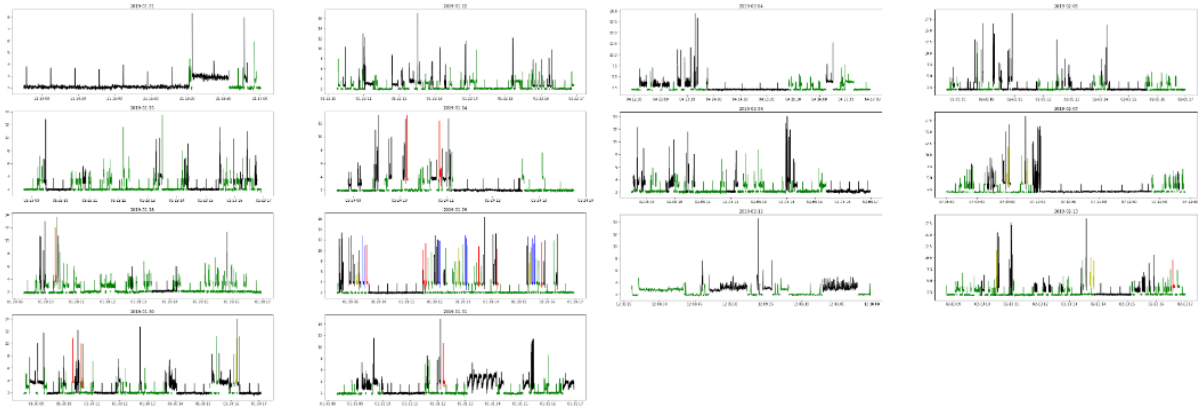


Figura 9.36: Resultado de DBSCAN y distancia DTW con la fresadora en activo

9.1.4 Resumen y conclusiones

En esta fresadora los resultados han sido más difíciles de evaluar, pero observando las secuencias del arranque parece obtener algo similar a los resultados esperados.

En el caso del arranque y el apagado de la fresadora los resultados más prometedores se consiguen con OPTICS y la distancia DTW. En el arranque se llegan a agrupar las secuencias en las 5 fases que se conocen del arranque (apagada,arranque, ciclo del PLC, arranque del motor y calentamiento del motor) y se llegan a identificar dos diferentes comportamientos para el apagado de la fresadora.

Sin embargo, al aplicar este algoritmo a la fresadora en activo los resultados son peores agrupando casi todas las secuencias como ruido. En este caso parece que se obtienen mejores resultados con un algoritmo de clustering jerárquico y la distancia euclidiana. Mientras que el enlace completo parece ser mejor en diferenciar las secuencias en las que la fresadora está inactiva o trabajando; el enlace Ward parece el mejor para identificar diferentes patrones de comportamiento.

9.2 Caso de Estudio: Fresadora Kaggle

En este apartado se aplicarán los algoritmos de clustering mencionados a los datos de una fresadora CNC que fueron publicados por el smart lab de la universidad de Michigan, que han sido publicados en [54].

9.2.1 Análisis datos

Los datos utilizados se corresponden con los datos de monitorización de 18 experimentos idénticos realizados con diferentes configuraciones de la fresadora.

El experimento consiste en realizar una pieza rectangular con una forma de S grabada en la cara superior. Todas las piezas se realizan con un bloque de cera de las mismas dimensiones.

En cada uno de los experimentos se modifican 3 parámetros de configuración:

- velocidad avance (mm/s): Velocidad de desplazamiento de la fresa durante la realización de la pieza.
- estado de la herramienta: Se realizan experimentos con una fresa en buenas condiciones y otros con una desgastada.
- presión (bar): Presión utilizada para sujetar la pieza .

Los datos de monitorización se recopilan cada 0.1 segundos y su evaluación consiste en dos pasos: la comprobación de que la fabricación de la pieza se completó y la comprobación visual de que el resultado es satisfactorio.

Cada uno de los experimentos se compone de diferentes fases. Para completar la realización de la pieza se requieren de 3 capas de fresado para la parte superior y para la parte inferior de la pieza (el fresado de la parte inferior y superior de la pieza es secuencial y no simultáneo).

La fresadora se compone de 4 motores los de los ejes X,Y y Z y el correspondiente al husillo. De cada uno de los 4 motores se recopila la posición, velocidad y aceleración tanto la real como la de referencia así como la corriente de realimentación, el voltaje de bus DC y la corriente y el voltaje de salida. Además se mide la inercia del husillo y de los ejes X,Y y del husillo la potencia.

Además se recopilan otros datos que no se corresponden a la acción de fresado, pero que son necesarios para el proceso como son el arranque, apagado, la preparación del experimento y el reposicionamiento de la pieza tras la realización de cada capa completa.

Tras terminar el experimento se obtiene el estado de la herramienta (si es utilizable o está desgastada), si se ha completado el mecanizado y si la pieza completada pasó la inspección visual.

Tras analizar los datos por las distintas capas se ha observado que los experimentos que quedaron sin completar presentan muchas menos muestras que los experimentos que si se completaron y contienen varias fases que ni se llegaron a empezar.

Los datos han sido utilizados para aprendizaje supervisado en clasificación con datos referentes a los experimentos y con datos muéstrales. En este trabajo se probarán para aprendizaje no supervisado y clustering y se aplicarán a los experimentos y a las fases.

Experimento	Velocidad Relativa	Presión	Herramienta	Mecanizado	Inspeccion
1	6	4.0	UTILZABLE	COMPLETADO	APROBADA
2	20	4.0	UTILZABLE	COMPLETADO	APROBADA
3	6	3.0	UTILZABLE	COMPLETADO	APROBADA
4	6	2.5	UTILZABLE	INACABADO	FALLIDA
5	20	3.0	UTILZABLE	INACABADO	FALLIDA
6	6	4.0	DESGASTADA	COMPLETADO	FALLIDA
7	20	4.0	DESGASTADA	INACABADO	FALLIDA
8	20	4.0	DESGASTADA	COMPLETADO	FALLIDA
9	15	4.0	DESGASTADA	COMPLETADO	FALLIDA
10	12	4.0	DESGASTADA	COMPLETADO	FALLIDA
11	3	4.0	UTILZABLE	COMPLETADO	APROBADA
12	3	3.0	UTILZABLE	COMPLETADO	APROBADA
13	3	4.0	DESGASTADA	COMPLETADO	APROBADA
14	3	3.0	DESGASTADA	COMPLETADO	APROBADA
15	6	3.0	DESGASTADA	COMPLETADO	APROBADA
16	20	3.0	DESGASTADA	INACABADO	FALLIDA
17	3	2.5	UTILZABLE	COMPLETADO	APROBADA
18	3	2.5	DESGASTADA	COMPLETADO	APROBADA

Figura 9.37: Resumen Experimentos

El equilibrio entre las muestras correspondientes a la fresa desgastada (6982 muestras) y a la no desgastada (5961 muestras) parece indicar que, los experimentos que obtuvieron que la fresa estaba desgastada empezaron el experimento con la fresa en esa condición.

9.2.2 Preparación de datos

Para la preparación de datos se han realizado una serie de pasos comunes tanto para obtener los datos divididos por fases como para dividirlos por experimentos.

- Cargar los datos de salida del fichero.
- Cargar datos de los experimentos del fichero y añadir columnas referentes a su salida correspondiente (añadiendo velocidad relativa, presión, si es utilizable o desgastado, si es completado e inacabado y si el resultado es aprobado o suspenso.
- Juntar datos de los experimentos.
- Eliminar etapas de no funcionamiento (solo se trabaja con las etapas referentes a la realización a cada una de las capas de la pieza).

Exp	Fin	C1abajo	C1arriba	C2abajo	C2arriba	C3abajo	C3arriba	Prep	Repon	Arranc	Herramienta	Mecanizado	Inspeccion
1	0	148	172	132	203	142	194	30	25	1	UTILZABLE	COMPLETADO	APROBADA
2	25	232	212	204	143	134	144	174	400	0	UTILZABLE	COMPLETADO	APROBADA
3	109	167	230	236	131	98	210	81	259	0	UTILZABLE	COMPLETADO	APROBADA
4	40	0	387	0	0	0	0	105	0	0	UTILZABLE	INACABADO	FALLIDA
5	380	0	72	0	0	0	0	10	0	0	UTILZABLE	INACABADO	FALLIDA
6	30	161	224	118	229	178	126	74	156	0	DESGASTADA	COMPLETADO	FALLIDA
7	21	93	195	0	39	0	0	51	166	0	DESGASTADA	INACABADO	FALLIDA
8	61	34	27	74	173	49	55	41	91	0	DESGASTADA	COMPLETADO	FALLIDA
9	31	109	105	103	98	121	89	11	73	0	DESGASTADA	COMPLETADO	FALLIDA
10	161	132	181	167	233	130	151	50	96	0	DESGASTADA	COMPLETADO	FALLIDA
11	292	156	405	179	499	157	213	90	323	0	UTILZABLE	COMPLETADO	APROBADA
12	196	245	346	236	213	251	223	43	523	0	UTILZABLE	COMPLETADO	APROBADA
13	169	272	342	253	197	228	274	150	348	0	DESGASTADA	COMPLETADO	APROBADA
14	93	348	363	206	316	268	342	221	175	0	DESGASTADA	COMPLETADO	APROBADA
15	232	53	138	171	70	70	267	245	135	0	DESGASTADA	COMPLETADO	APROBADA
16	264	144	69	0	0	0	0	122	3	0	DESGASTADA	INACABADO	FALLIDA
17	283	201	244	253	259	189	182	216	323	0	UTILZABLE	COMPLETADO	APROBADA
18	198	160	373	196	301	339	324	81	281	0	DESGASTADA	COMPLETADO	APROBADA

Figura 9.38: Resumen Fases por experimentos

- Eliminar mediciones inexactas que se corresponden a valores específicos de la posición actual X y a valores del programa de CNC.

- Eliminar columnas sin información de las mediciones, se eliminan datos sobre la configuración de la fresadora el programa CNC y la línea de código G y las variables que presentan un único valor en todas las muestras independientemente del experimento.

Una vez aplicados estos pasos los datos se separan en experimentos y en fases junto con los resultados por fases y por experimentos.

Los métodos se realizaron para poder indicar los experimentos específicos que seleccionar para comprobar los efectos que tienen el desequilibrio del número de muestras de los experimentos en los resultados.

9.2.3 Análisis Modelos

Se han aplicado los algoritmos de clustering basados en densidad OPTICS y DBSCAN, el algoritmo basado en partición K-Means y el algoritmo de clustering jerárquico con enlaces simple, completo, promedio y Ward.

En los algoritmos donde se puede aplicar se realizará el clustering con la distancia euclidiana y la distancia DTW.

Clustering Aplicado a los experimentos

Tras aplicar clustering a los datos de cada uno de los experimentos (se agrupan 18 experimentos), se comprobó si los datos se agrupaban en función de: el desgaste, de si se completo la pieza, de si pasó la inspección visual o si se agrupan por alguna combinación de estas tres variables (considerando 8 posibles resultados desde Utilizable-completado-aprobado hasta desgastado-sin completar - suspenso).

A raíz de los resultados, se observa que los datos se agrupan en función de si aprobaron o suspendieron la inspección visual y no parecen servir para identificar el desgaste. Dada la relación que existe entre los completados y la inspección visual (ya que solo los completados son candidatos a pasar la inspección visual), el clustering parece dar resultados adecuados para medir si el experimento se completó o no (aunque sus resultados son ligeramente inferiores que en el caso de la inspección).

Se observa que la distancia euclidiana en este caso da mejores resultados que la distancia DTW. Se distinguen que independientemente del algoritmo los resultados son similares presentando dos grupos uno donde se agrupan sólo aprobados y otro donde se agrupan los suspensos junto con otros aprobados. Se ha probado en métodos que permiten fijar el número de grupos (K-Means y jerárquico) que al aumentar el número de grupos se dividen los grupos de aprobados mientras que los suspensos suelen permanecer en un mismo grupo todos.

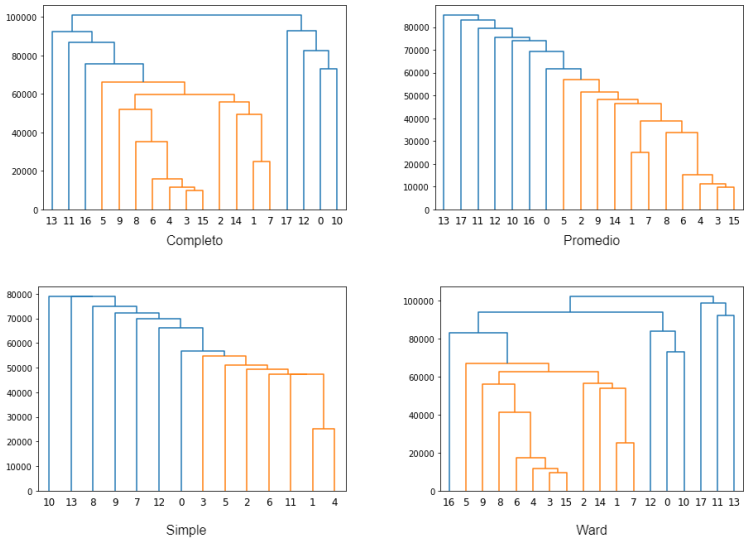


Figura 9.39: Comparación enlaces Jerárquicos en Experimentos

Con enlace jerárquico se observa que independientemente del enlace, los resultados obtenidos han sido los mismos, aunque cambie el dendrograma obtenido.

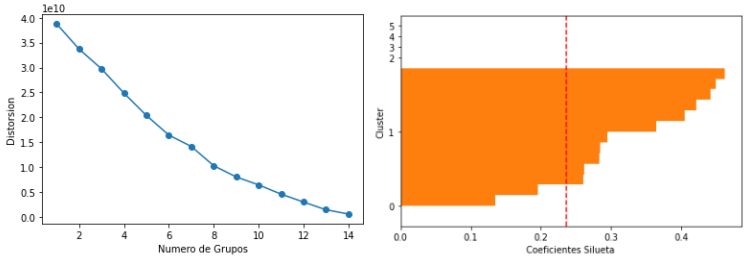


Figura 9.40: Resultados K-Means experimentos

En K-Means a pesar de que los resultados obtenidos son buenos vale la pena mencionar que los diagramas no lo son. El diagrama de codo no muestra ningún cambio brusco en la distorsión que permita determinar un número bueno de grupos y el índice de silueta presenta un valor considerablemente bajo.

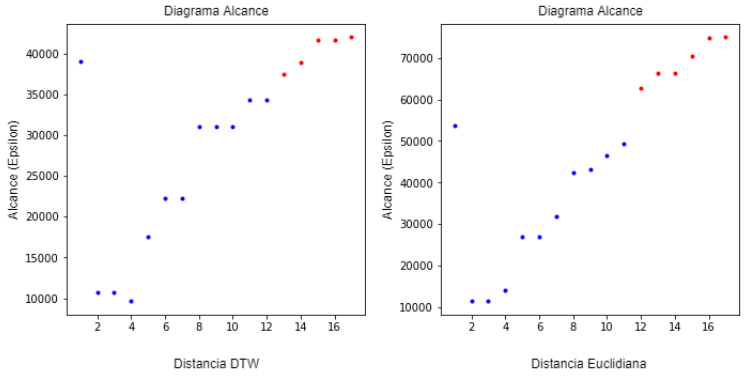


Figura 9.41: Diagrama alcance Optics Distancia Eucliana y DTW

Los resultados al aplicar OPTICS con ambas distancias son muy similares, se observa en el diagrama de alcance la similitud entre ambos. A pesar de ser ligeramente diferentes en su forma, los puntos empinados descendentes y ascendentes se encuentran en alturas similares. Hay que tener en cuenta que el primer punto del diagrama de alcance tiene distancia infinita y por eso en el diagrama solo aparecen 17 puntos, en el caso de DTW el sexto elemento identificado como ruido es el de la primera posición.

	DBSCAN		OPTICS		K MEANS		JERARQUICO mismos resultados para los diferentes enlaces	
Grupos	Aprob	Susp	Aprob	Susp	Aprob	Susp	Aprob	Susp
-1	6	0	6	0				
0	4	8	4	8	1	0	3	8
1					5	8	1	0
2					1	0	1	0
3					1	0	1	0
4					1	0	1	0
5					1	0	1	0
6							1	0
7							1	0

Tabla 9.1: Clustering Experimentos distancia Euclidiana

	DBSCAN		OPTICS		JERARQUICO COMPLETO		JERARQUICO SIMPLE		JERARQUICO PROMEDIO	
Grupos	Aprob	Susp	Aprob	Susp	Aprob	Susp	Aprob	Susp	Aprob	Susp
-1	7	1	6	0						
0	3	7	4	8	6	1	1	1	8	2
1					0	5	3	7	2	6
2					4	2	1	0		
3							1	0		
4							1	0		
5							1	0		
6							1	0		
7							1	0		

Tabla 9.2: Clustering Experimentos distancia DTW

Clustering Aplicado a los fases

En el caso del clustering aplicado a las fases, se agrupan 91 fases provenientes de los 18 experimentos y sin distinguir la capa que está realizando cada fase. En el caso de la agrupación de fases sigue sin dar buenos resultados para detectar el desgaste, pero da resultados relativamente buenos para identificar si se completó el experimento y si pasó la inspección. A diferencia del clustering de experimentos, en las fases los resultados parecen ser mejores para identificar si se completó la pieza que para detectar si pasó la inspección visual.

En el caso de las fases, los resultados obtenidos con la distancia euclidiana y DTW es bastante similar. Al igual que pasaba en el caso de los experimentos, la forma de los resultados es similar, presentando en general dos agrupaciones, un cluster con solo completados y otro donde se agrupan todos los inacabados junto con algunos completados e igual que pasaba en el caso de los experimentos al aumentar el número de grupos se

dividen los completados y los inacabados se mantienen juntos en un mismo cluster (hasta cierto límite). En jerárquico se puede observar que se puede incrementar el número de grupos hasta 70 grupos con distancia euclidiana y enlace simple manteniéndose todas las fases inacabadas juntas en el mismo cluster.

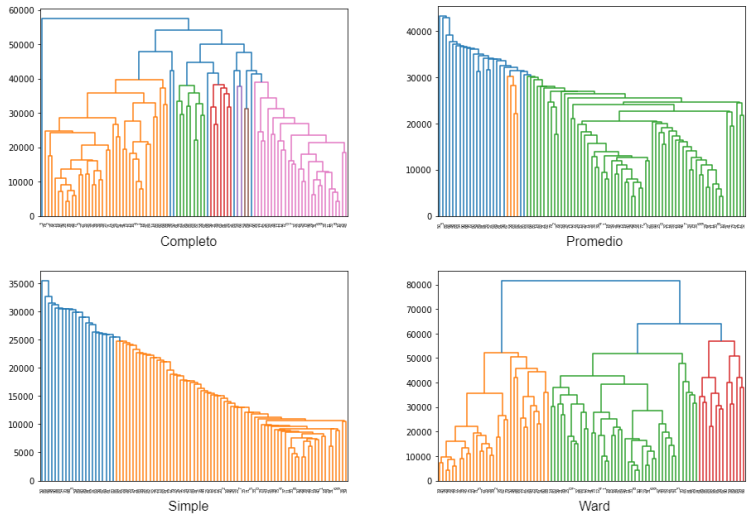


Figura 9.42: Comparación enlaces Jerárquicos en Fases

El enlace Ward parece dar resultados ligeramente peores que el resto de algoritmos.

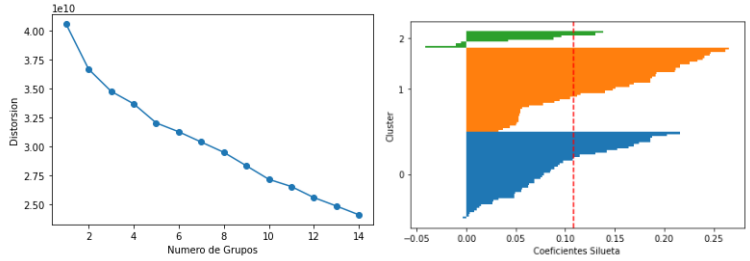


Figura 9.43: Resultados K-Means fases

Igual que pasaba en el caso de los experimentos, aunque el resultado observado parece ser aceptable tras aplicar K-Means, el índice de silueta obtenido es muy bajo y los elementos de un mismo grupo no son muy similares entre sí. Además el diagrama del codo no presenta ningún suavizado evidente que pueda servir para identificar el número óptimo de grupos.

En el caso de OPTICS se observa un cambio en la forma del diagrama con la distancia DTW. En el clustering de fases con DTW ahora se observa dos ligeros mínimos. Variando los parámetros relativos a la pendiente del diagrama de alcance y al tamaño mínimo de

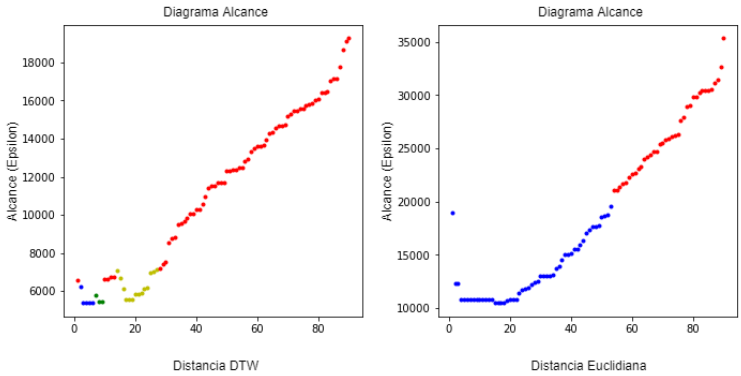


Figura 9.44: Diagrama alcance de las Fases con distancia Euclidiana y DTW

los clusters se pueden obtener más de un grupo (sin contar el ruido), sin embargo a la vista de los resultados de las tablas se observa que este aumento en el número de grupos empeora los resultados.

	DBSCAN		OPTICS		K MEANS		JERARQUICO COMPLETO		JERARQUICO SIMPLE		JERARQUICO PROMEDIO		JERARQUICO WARD	
Grupos	Comp	Inac	Comp	Inac	Comp	Inac	Comp	Inac	Comp	Inac	Comp	Inac	Comp	Inac
-1	53	0	37	0										
0	31	7	47	7	42	0	34	0	15	7	73	7	14	0
1					34	7	17	0	1	0	1	0	27	6
2					8	0	2	0	1	0	1	0	43	1
3							1	0	1	0	1	0		
4							30	7	1	0	1	0		
...									1	0	1	0		
11									1	0	1	0		
...									1	0				
70									1	0				

Tabla 9.3: Clustering Fases distancia Euclidiana

	DBSCAN		OPTICS		JERARQUICO COMPLETO		JERARQUICO SIMPLE		JERARQUICO PROMEDIO	
Grupos	Comp	Inac	Comp	Inac	Comp	Inac	Comp	Inac	Comp	Inac
-1	53	0	68	1						
0	31	7	5	0	21	0	19	7	37	7
1			3	0	51	0	1	0	15	0
2			8	6	12	7	1	0	19	0
3							1	0	6	0
...							1	0	1	0
11							1	0	1	0
...							1	0		
66							1	0		

Tabla 9.4: Clustering Fases distancia DTW

9.2.4 Resumen y conclusiones

En general se consiguen resultados similares con los diferentes algoritmos aplicados. El resultado más común en el caso de agrupar los experimentos es que se aíslen los que suspendieron la inspección visual en un único grupo junto con algunos experimentos que la aprobaron y en el resto de grupos se encuentren solo experimentos cuya conclusión fue satisfactoria. Tanto OPTICS como el algoritmo jerárquico con enlace simple han obtenido buenos resultados independientemente de la distancia aplicada.

En el caso de analizar las fases, el resultado de la agrupación tiene más relación con la finalización completa de los experimentos. Aunque al analizar los experimentos, estos parecen agruparse en función de si pasaron la inspección visual, en ambos casos el comportamiento de la agrupación es similar. Esto se debe a la relación que existe entre ambos sucesos (sólo pueden aprobar la inspección los experimentos acabados).

De forma parecida al caso de los experimentos, se agrupan en un mismo grupo las fases inacabadas junto con fases acabadas dejando en el resto de grupos sólo fases acabadas. Los mejores resultados parecen obtenerse con DBSCAN y con el jerárquico empleando el enlace simple. En el caso de DBSCAN, se agrupa en un único grupo todas las fases de experimentos inacabados junto con varios completados y al resto los etiqueta como ruido. En el jerárquico simple se pueden aislar experimentos completados en grupos de un único elemento hasta determinado nivel.

Sobre la distancia a empleada al no haber diferencia en ambos resultados, la distancia euclidiana parece más adecuada al ser más rápida. Sin embargo hay que tener en cuenta que tanto las fases como los experimentos presentan diferentes números de muestras y la distancia DTW a diferencia de la euclidiana permite calcular la distancia en series de dife-

rentes longitudes. Aplicar la distancia DTW evitaría tener que aplicar alguna modificación a los datos para poder aplicar la distancia euclidiana.

Sin información del conjunto de datos los más adecuados a aplicar parecen ser OPTICS en el caso de experimentos y DBSCAN en el caso de analizar las fases.

Capítulo 10

Caso de Estudio: Dashboard

La visualización interactiva consiste en usar representaciones visuales para descubrir patrones y configurar los modelos, a través de la dirección del usuario y obtener con ello conocimiento de los datos.

El análisis visual combina el aprendizaje automático con la visualización interactiva . Consiste en realizar un análisis inicial para mostrar lo importante, para luego analizar en profundidad ampliando y filtrando para mostrar detalles por petición del usuario.

A través de la analítica visual los usuarios pueden interactuar con los algoritmos modificando la configuración de los algoritmos o el algoritmo, mostrar nuevos resultados solicitados ... Estas acciones se pueden llevar a cabo mediante interfaces interactivas, de esta forma los usuarios participan en el proceso y dan retroalimentación al sistema.

Las tareas de visualización se clasifican en: tareas de alto nivel (Análisis), que analizan los datos a través de herramientas de visualización para obtener información en diferentes dominios y obtener nueva información; tareas de nivel medio (Búsqueda), que buscan, exploran,localizan y exploran elementos de interés; y tareas de bajo nivel (consulta), que se realizan después del análisis y consisten en identificar, comparar o resumir los resultados.

En este trabajo se ha realizado un pequeño dashboard que permite visualizar los datos y alguna característica de los datos con los que se han trabajado, así como los resultados obtenidos .

10.1 Análisis

En este apartado se describe al usuario de la aplicación junto con los requisitos de usuario, los requisitos funcionales, no funcionales y de información que tendrá la aplicación.

10.1.1 Descripción de Actores

Solo habrá un tipo de usuario con acceso a toda la aplicación sin necesidad de registro.

Id	Actor	Descripción
A1	Usuario General	Cualquier usuario que accede a la aplicación, puede visualizar los resultados obtenidos en ambas fresadoras.

Tabla 10.1: Actores

10.1.2 Requisitos de Usuario

Las acciones que podrá realizar el usuario se indican en el siguiente diagrama.

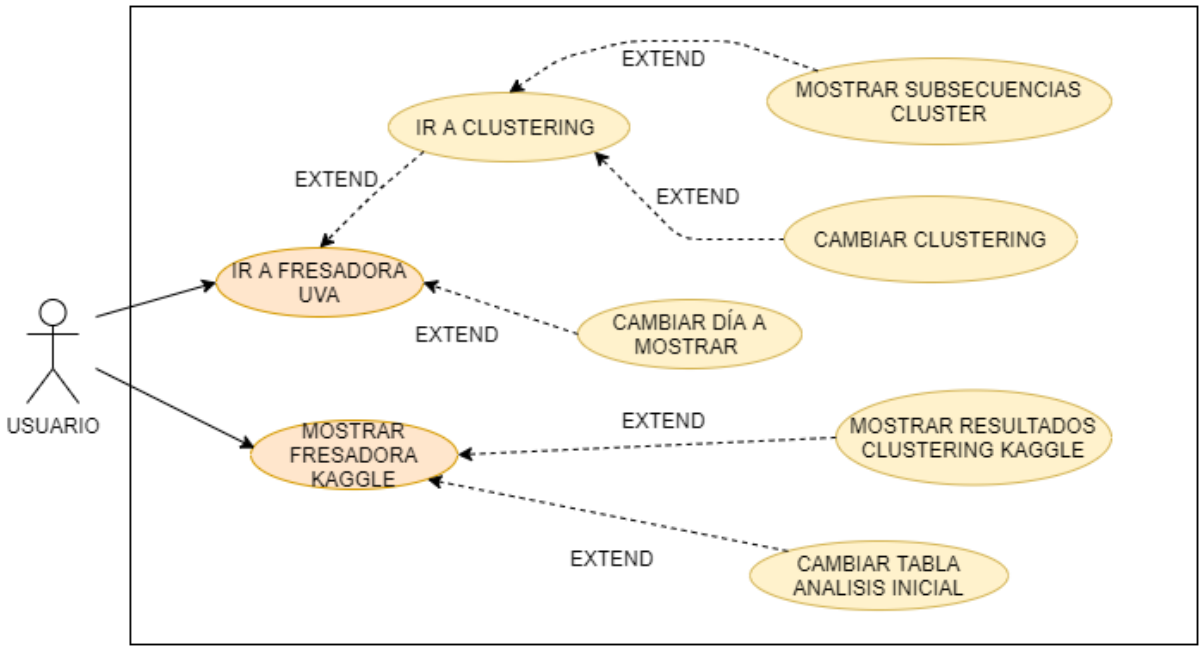


Figura 10.1: Diagrama casos de uso

Especificación de los Casos de Uso

Se explican los casos de uso que se han identificado en el diagrama.

CU-01	Mostrar Fresadora CIDAUT	
Descripción	El usuario accede a la visualización de los datos de corriente de la fresadora CIDAUT permitiendo visualizar la señal de los diferentes días así como las fases y secuencias empleadas para el clustering.	
Autor	Barreno Recio, Patricia Lucía	
Actor	Usuario General	
Precondiciones		
Secuencia Normal	Paso 1	El usuario selecciona ir a la fresadora CIDAUT.
	Paso 2	El sistema muestra diagrama de la corriente de los diferentes días (RI-06).
	Paso 3	El sistema muestra las fases y la corriente del primer día examinado.
Postcondiciones	La página de la fresadora CIDAUT se ha cargado adecuadamente.	

Tabla 10.2: CU-1 Mostrar Fresadora CIDAUT

CU-02	Cambiar día a mostrar	
Descripción	El usuario cambia el día del que se muestran las fases y las secuencias del arranque.	
Autor	Barreno Recio, Patricia Lucía	
Actor	Usuario General	
Precondiciones	El usuario ha realizado el CU-01 Ir a fresadora CIDAUT.	
Secuencia Normal	Paso 1	El usuario selecciona cambiar día a mostrar.
	Paso 2	El sistema muestra lista de días examinados (RI-06.04).
	Paso 3	El usuario selecciona el día a mostrar.
	Paso 4	El sistema muestra la corriente diaria seleccionada mostrando fases y secuencias de arranque (RI-06,RI-02).
Postcondiciones	La página de la fresadora CIDAUT muestra el día indicado.	

Tabla 10.3: CU-02 Cambiar día a mostrar

CU-03	Mostrar Fresadora Kaggle	
Descripción	El usuario accede a la página de visualización de los datos de salida de la fresadora Kaggle.	
Autor	Barreno Recio, Patricia Lucía	
Actor	Usuario General	
Precondiciones		
Secuencia Normal	Paso 1	El usuario selecciona ir a la fresadora Kaggle.
	Paso 2	El sistema muestra diagrama información de salida de los experimentos que se realizaron con la fresadora.(RI-03 y RI-04).
Postcondiciones	La página de la fresadora kaggle se ha cargado adecuadamente.	

Tabla 10.4: CU-03 Mostrar Fresadora Kaggle

CU-04	Ir a clustering	
Descripción	El usuario accede a la página de clustering de la fresadora CIDAUT.	
Autor	Barreno Recio, Patricia Lucía	
Actor	Usuario General	
Precondiciones	El usuario ha realizado el CU-01 Ir a fresadora CIDAUT	
Secuencia Normal	Paso 1	El usuario selecciona mostrar clustering.
	Paso 2	El sistema accede a la página de clustering CIDAUT .
	Paso 3	El sistema muestra por defecto clustering del arranque con DBSCAN y distancia dtw (RI-01, RI-02, RI-05, RI-06).
Postcondiciones	La página de clustering CIDAUT se ha cargado adecuadamente.	

Tabla 10.5: CU-04 Ir a clustering

CU-05	Seleccionar Clustering	
Descripción	El sistema visualiza el clustering realizado mostrando las series de tiempo analizadas mostrando con los mismos colores las secuencias del mismo grupo.	
Autor	Barreno Recio, Patricia Lucía	
Actor	Usuario General	
Precondiciones	El usuario ha realizado el CU-04 ir a clustering.	
Secuencia Normal	Paso 1	El usuario selecciona fase a realizar el clustering y distancia.
	Paso 2	El sistema muestra la lista de algoritmos disponibles para dicha fase (RI-01) .
	Paso 3	El usuario selecciona el algoritmo a aplicar.
	Paso 4	Si el algoritmo es jerárquico el usuario selecciona tipo de enlace.
	Paso 5	El usuario selecciona mostrar clustering.
	Paso 6	El sistema visualiza el clustering seleccionado, 14 diagramas mostrando del mismo color las secuencias del mismo cluster (RI-01, RI-02, RI-05, RI-06).
	Paso 7	El sistema actualiza la lista de clusters disponibles para mostrar (RI-01).
Excepciones	E5.1	Si el usuario había seleccionado K-Means o enlace Ward con distancia DTW, el sistema cambia la distancia a euclidiana y lanza un mensaje.
	E5.1	Si el usuario había seleccionado midbscan, miOptics o miKmedoides con distancia euclidiana, el sistema cambia la distancia a DTW y lanza un mensaje.
Postcondiciones	La página de clustering CIDAUT se ha actualizado para mostrar el clustering realizado y la lista de clusters correspondiente.	

Tabla 10.6: CU-05 Seleccionar Clustering

CU-06	Mostrar subsecuencias cluster	
Descripción	El sistema visualiza las distintas subsecuencias que se encuentran en un cluster seleccionado por el usuario.	
Autor	Barreno Recio, Patricia Lucía	
Actor	Usuario General	
Precondiciones	El usuario ha realizado el CU-05 seleccionar clustering.	
Secuencia Normal	Paso 1	El usuario selecciona mostrar cluster obtenidos.
	Paso 2	El sistema muestra la lista de clusters referentes al clustering seleccionado en CU-05.
	Paso 3	El usuario selecciona el cluster a mostrar .
	Paso 4	El sistema muestra un gráfico con las subsecuencias del cluster indicado (RI-01, RI-02, RI-05, RI-06) .
Postcondiciones	La página de clustering CIDAUT se ha actualizado para mostrar el cluster indicado.	

Tabla 10.7: CU-06 Ir a clustering

CU-07	Mostrar resultados clustering Kaggle	
Descripción	El sistema muestra resultados del clustering aplicado según un criterio de salida indicado por el usuario	
Autor	Barreno Recio, Patricia Lucía	
Actor	Usuario General	
Precondiciones	El usuario ha realizado el CU-02 Mostrar Fresadora Kaggle.	
Secuencia Normal	Paso 1	El usuario selecciona ir a clustering.
	Paso 2	El sistema muestra las opciones del clustering .
	Paso 3	El usuario indica a qué datos aplicar el clustering, distancia y criterio de resultados .
	Paso 4	El sistema muestra tabla con los resultados según la selección indicada por el usuario (RI-01,RI-03,RI-04,RI-05).
Postcondiciones	La página de clustering Kaggle se ha actualizado para mostrar la tabla con el resultado del clustering indicado.	

Tabla 10.8: CU-07 Mostrar resultados clustering Kaggle

CU-08	Cambiar Tabla Análisis Inicial	
Descripción	El sistema muestra una tabla resumen de los datos a los que se aplicará el clustering.	
Autor	Barreno Recio, Patricia Lucía	
Actor	Usuario General	
Precondiciones	El usuario ha realizado el CU-02 Mostrar Fresadora Kaggle.	
Secuencia Normal	Paso 1	El usuario selecciona entre salida de experimentos, fases de cada experimento y fases según criterio de salida.
	Paso 2	El sistema muestra la tabla que el usuario seleccione (RI-03 y RI-04).
Postcondiciones	La página de fresadora Kaggle se ha actualizado para mostrar la tabla indicada.	

Tabla 10.9: CU-08 Cambiar Tabla Análisis Inicial

10.1.3 Requisitos Funcionales

Para poder satisfacer las necesidades del usuario y poder realizar los casos de uso, el sistema debe proporcionar las siguientes funciones.

Identificador	Requisito Funcional
RF-01	El sistema muestra el diagrama con todas las señales de corriente de los días examinados.
RF-02	El sistema mostrará las fases identificadas de una señal de corriente de uno de los días examinados.
RF-03	El sistema mostrará el arranque de un día concreto identificando las subsecuencias que se emplearán para el clustering .
RF-04	El sistema permitirá cambiar el día seleccionado para examinar las fases y secuencias .
RF-05	El sistema permitirá mostrar la tabla de información sobre los datos de la fresadora kaggle que se han aplicado.
RF-06	El sistema permitirá cambiar la tabla con información de la fresadora Kaggle.
RF-07	El sistema mostrará el clustering realizado en fresadora CIDAUT mostrando los días examinados.
RF-08	El sistema permitirá seleccionar la fase de la fresadora CIDAUT de la que observar el clustering.
RF-09	El sistema permitirá seleccionar la distancia que se empleará en el clustering que se quiere observar.
RF-10	El sistema permitirá seleccionar el algoritmo de clustering del que se quieren comprobar los resultados .
RF-11	El sistema permitirá seleccionar el tipo de enlace utilizado en el algoritmo jerárquico del que se quieren observar los resultados.
RF-12	El sistema permitirá seleccionar el tipo de enlace utilizado en el algoritmo jerárquico del que se quieren observar los resultados.
RF-13	El sistema permitirá actualizar el clustering de la fresadora CIDAUT con los datos seleccionados.
RF-14	El sistema cambiará la distancia seleccionada por euclidiana en el caso de haber sido seleccionado como algoritmo K-Means o enlace Ward.
RF-15	El sistema cambiará la distancia seleccionada por DTW si se ha seleccionado como algoritmo miDbscan ,miOptics o miKmedoides.
RF-16	El sistema mostrará un mensaje en caso de modificarse la selección de parámetros del usuario .
RF-17	El sistema permitirá seleccionar uno de los cluster del clustering seleccionado por el usuario.
RF-18	El sistema mostrará un gráfico que contenga las secuencias que pertenecen al cluster que haya sido seleccionado.
RF-19	El sistema muestra la tabla de resultados del clustering empleado en la fresadora Kaggle.
RF-20	El sistema permite modificar cambiar la tabla de resultado del clustering según tipo, distancia y criterio.

Tabla 10.10: RFuncional

10.1.4 Requisitos de Información

En este apartado se describen los datos que se utilizaran en el dashboard. Estos datos no coinciden del todo con los que se trabajó en el apartado 9, son solo una parte de ellos para mostrar los resultados obtenidos.

Modelo Entidad Relación

El diagrama de entidad relación muestra los elementos con los que se ha trabajado y cómo interactúan entre ellos. En el diagrama mostrado aparece una entidad y una relación en gris dado que aunque semánticamente está presente en el dashboard no se utilizan sus datos y por tanto no se tendrán en cuenta en la aplicación.

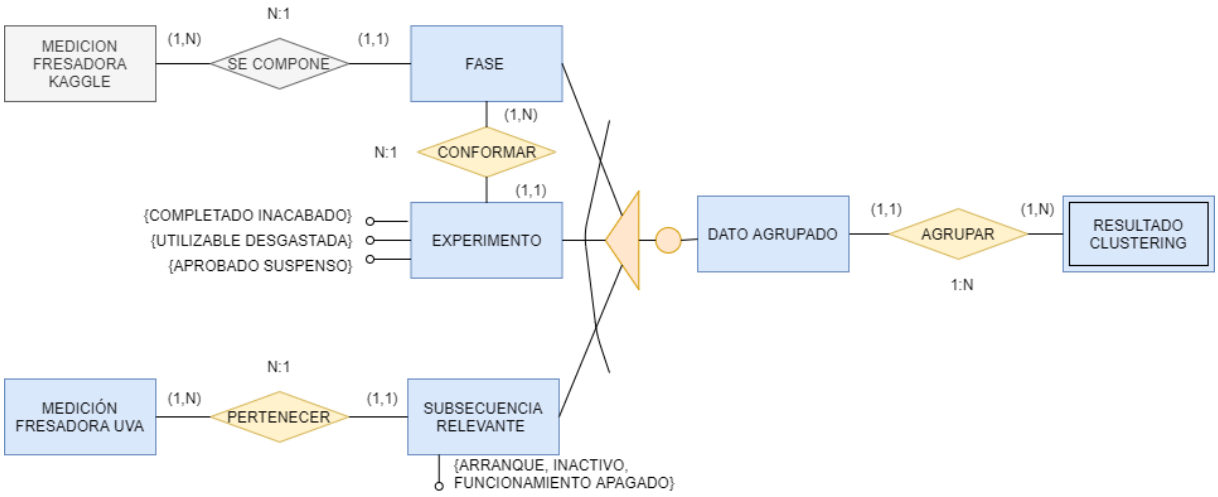


Figura 10.2: Diagrama entidad-relación

Diccionario de datos de la aplicación

En este apartado se explicaran cada una de las entidades y relaciones que aparecen en el diagrama entidad relación y que se utilizaran en la aplicación.

Un resultado de clustering queda determinado por el elemento al que se corresponde el resultado y las características del clustering (sin considerar los parámetros utilizados en los algoritmos) que son el algoritmo, la medida de distancia y el conjunto de datos que se agruparon.

RI-01	ResultadoClustering					
Definición	Contiene las etiquetas correspondientes a los clustering realizados a los conjuntos de datos examinados.					
Consideraciones	Entidad débil de datos agrupados, un resultado por objeto, conjunto, algoritmo y distancia.					
Atributos						
ID	Nombre	Descripción	Dominio	Unique	Null	Notas
RI-01.01	id	índice del resultado auto incremental.	Entero	No	No	
RI-01.02	conjunto	datos que se han agrupado para obtener este resultado	Text	No	No	Un mismo objeto se ha podido utilizar en varios conjuntos.
RI-01.03	etiqueta	grupo del elemento obtenido al agrupar el conjunto indicado con el algoritmo y la distancia también indicados.	Entero	No	No	Valor mínimo -1
RI-01.04	algoritmo	algoritmo de clustering aplicado .	ENUM(K-Means, jerárquico, DBSCAN, Optics, miDbscan, miOptics, miKmedoides)	No	No	Los 3 algoritmos del final solo se han aplicado en arranque.
RI-01.05	enlace	tipo de enlace empleado con clustering jerárquico.	ENUM(simple, completo, promedio, ward)	No	Si	solo en jerárquico
RI-01.06	distancia	medida de distancia aplicada en el algoritmo de clustering .	ENUM(DTW, euclidiana)	No	No	

Tabla 10.11: RI-01 ResultadoClustering

RI-02	subsecuencia relevante					
Definición	Conjunto de mediciones continuas de corriente que presentan cierta relación en su comportamiento.					
Consideraciones						
Atributos						
ID	Nombre	Descripción	Dominio	Unique	Null	Notas
RI-02.01	numSegmento	Identifica la secuencia y la ordena según el momento de tiempo en que ocurrió	Entero	Si	No	
RI-02.02	TipoSegmento	Identifica a qué etapa de la fresadora corresponde la secuencia	ENUM (ARRANQUE, INACTIVO, FUNCIONAMIENTO, APAGADO)	No	No	

Tabla 10.12: RI-02 subsecuencia relevante

RI-03	Fase					
Definición	Etapa de trabajo en la fresadora Kaggle durante un experimento. que está conformada por un conjunto de mediciones.					
Consideraciones	Entidad débil de experimento					
Atributos						
ID	Nombre	Descripción	Dominio	Unique	Null	Notas
RI-03.01	Id	número de ejecución de la fase	Entero	Si	No	
RI-03.02	NombreFase	Tipo de fase que se realiza	Enum (C1Inf, C1Sup, C2Inf, C2Sup, C3Inf, C3Sup)	No	No	

Tabla 10.13: RI-03 Fase

RI-04	Experimento					
Definición	Trabajo de mecanizado para elaborar completamente la pieza indicada					
Consideraciones						
Atributos						
ID	Nombre	Descripción	Dominio	Unique	Null	Notas
RI-4.01	Experimento	Número del experimento	Entero	Si	No	
RI-4.02	Herramienta	Estado de la herramienta de corte (fresa)	ENUM (UTILIZABLE, DESGASTADA)	No	No	
RI-4.03	Mecanizado	Resultado de finalización del experimento	ENUM (COMPLETADO , INACABADO)	No	No	
RI-4.04	Inspección	Resultado de la evaluación de la inspección visual	ENUM (APROBADA, FALLIDA)	No	No	
RI-4.04	VelocidadRel	Velocidad relativa de la fresa durante el experimento	Real	No	No	
RI-4. {05-14}	Fin, C1Inf, C1Sup, C2Inf, C2Sup, C3Inf, C3Sup, Prep, Repon, Arranc	Número de muestras que tiene el experimento en esa etapa	Entero	No	No	

Tabla 10.14: RI-04 Experimento

RI-05	Dato Agrupado					
Definición	Elemento que se pasa como entrada al algoritmo de agrupamiento					
Consideraciones						
Atributos						
ID	Nombre	Descripción	Dominio	Unique	Null	Notas
RI-05.01	identificador	Identifica un elemento utilizado para el clustering	Entero	No	No	

Tabla 10.15: RI-05 Dato Agrupado

RI-06		medición fresadora CIDAUT				
Definición		Mediciones tomadas de la fresadora entre las 7:45 y las 17:15 en intervalos de 5 segundos				
Consideraciones						
Atributos						
ID	Nombre	Descripción	Dominio	Unique	Null	Notas
RI-06.01	FechaHora	contiene la fecha completa en que se tomó la medición	Datetime	Si	No	
RI-06.02	numMuestra	indica el orden de la muestra	Entero	Si	No	
RI-06.03	HoraEpoca	Hora en formato época en que se tomó la medición (sin contar la fecha)	Entero	No	No	
RI-06.04	numDia	Indica que día de los examinados es la medición	Entero	No	No	
RI-06.05	Corriente Fase 1	valor de la corriente de la fase 1	Real	No	No	

Tabla 10.16: RI-06 medición fresadora CIDAUT

RI-R1	conforma			
Definición	Asocia una fase de trabajo con el experimento en el que se realizó			
Consideraciones				
Atributos				
ID	Nombre	Participació	Cardinalidad	Notas
RI-02	Fases	1	1	
RI-03	Experimentos	1	N	

Tabla 10.17: RI-R1 conforma

RI-R2	pertenece			
Definición	conecta una subsecuencia de la serie de tiempo con las mediciones continuas que la conforman			
Consideraciones				
Atributos				
ID	Nombre	Participación	Cardinalidad	Notas
RI-05	subsecuencia relevante	1	N	
RI-09	mediciones fresadora CIDAUT	1	1	

Tabla 10.18: RI-R2 pertenece

RI-R3	agrupa			
Definición	asocia unos datos con los resultados de su agrupación			
Consideraciones				
Atributos				
ID	Nombre	Participación	Cardinalidad	Notas
RI-04	Datos Agrupados	1	N	
RI-01	resultadosClustering	1	1	

Tabla 10.19: RI-R3 agrupa

10.1.5 Requisitos No Funcionales

Requisitos sobre algunas restricciones y características que deberá cumplir el dashboard.

Requisitos de Calidad

Se establecen unos requisitos para facilitar la navegación del usuario a través del tablero.

Identificador	Requisito No Funcional
	Escalabilidad
RNF-01	La aplicación será compatible con navegadores genéricos.
	Accesibilidad
RNF-02	La aplicación se adaptará a diferentes tamaños de pantalla.
	Rendimiento
RNF-03	El sistema no tardará más de 8 segundos en responder.

Tabla 10.20: RNFuncional

Reglas de Negocio

Debido a las características del clustering y a los métodos realizados se han establecidos las siguientes reglas de negocio.

Identificador	Regla de negocio
ReN-01	Un dato a agrupar solo puede tener un resultado con los mismos valores en conjunto, algoritmo, enlace y distancia.
ReN-02	Si el algoritmo es jerárquico se debe indicar el tipo de enlace aplicado.
ReN-03	Si el algoritmo no es jerárquico el enlace deberá ser nulo.

Tabla 10.21: RNegocio

10.2 Diseño

En este apartado se recoge la información relacionada con el diseño del dashboard incluyendo la arquitectura del framework utilizado como algunos diagramas del diseño aplicado.

10.2.1 Arquitectura

Como se comentó en la introducción, la aplicación se desarrollará en Dash. Las aplicaciones Dash consisten en servidores web que ejecutan Flask y comunican paquetes JSON a través de solicitudes http. En la interfaz procesa componentes usando React.js, estos componentes son clases de python que codifican las propiedades y los valores de un componente react que se serializa con JSON.

Las aplicaciones Dash se dividen en dos partes, una que se encarga del aspecto de la aplicación y otra de su interacción con el usuario.

Dash describe el aspecto de la aplicación a través del app.layout que consiste en un árbol jerárquico de componentes. Dash ofrece una serie de componentes para crear la interfaz de usuario. Estos componentes se dividen en dos paquetes el paquete de dash_html_component donde se encuentran las etiquetas HTML y el paquete dash_core_component que contienen los componentes interactivos que se encargan de la funcionalidad del tablero.

La interacción del tablero se controla mediante llamadas. Dash permite definir llamadas que al actualizar un componente del interfaz invoquen código python capaz de realizar algún proceso y cambiar la interfaz.

La arquitectura realizada sigue la estructura indicada por la documentación de Dash para aplicaciones multipágina . Separando en un fichero la clase principal con la aplicación, el servidor y el estilo principal. Esto se hace para evitar importaciones circulares.

El diseño seguido consiste en dividir la aplicación por las páginas que la componen teniendo una clase por cada página que contenga tanto su diseño como su interacción.

10.2.2 Diagrama Lógico de Datos

La aplicación es solo de consulta y no se realizan inserciones, actualizaciones o borrados en los datos. Los datos utilizados se utilizan en la aplicación para visualización de resultados.

Los datos de la aplicación identificados en el diagrama entidad relación, se han manipulado y se han guardado en ficheros CSV para su uso en la aplicación.

En concreto se utilizan 9 ficheros diferentes, 5 de los cuales corresponden a los resultados de aplicar clustering a las fases, experimentos, arranque, activa y apagado de la fresadora.

El resto de ficheros se corresponden con los datos de secuencia que permiten visualizar las series de tiempo que se utilizarán para evaluar de forma visual el clustering aplicado y del mismo modo ,para los otros datos se utilizan dos ficheros con la salida (información de la herramienta y del resultado del mecanizado y la inspección visual) de los experimentos y fases.

Además de otro fichero con información del número de muestras de cada fase según los criterios de salida mencionados.

10.2.3 Diagramas de Clases

El diagrama de clases que se ha realizado por el dashboard consiste en una clase por cada página además del index y de la clase miApp que contiene la instancia Dash. Además se han creado tres clases con los métodos para realizar los diagramas y tabla, así como de acceder a los ficheros de datos. Las clases de métodos serán invocados mediante las llamadas de las diferentes páginas.

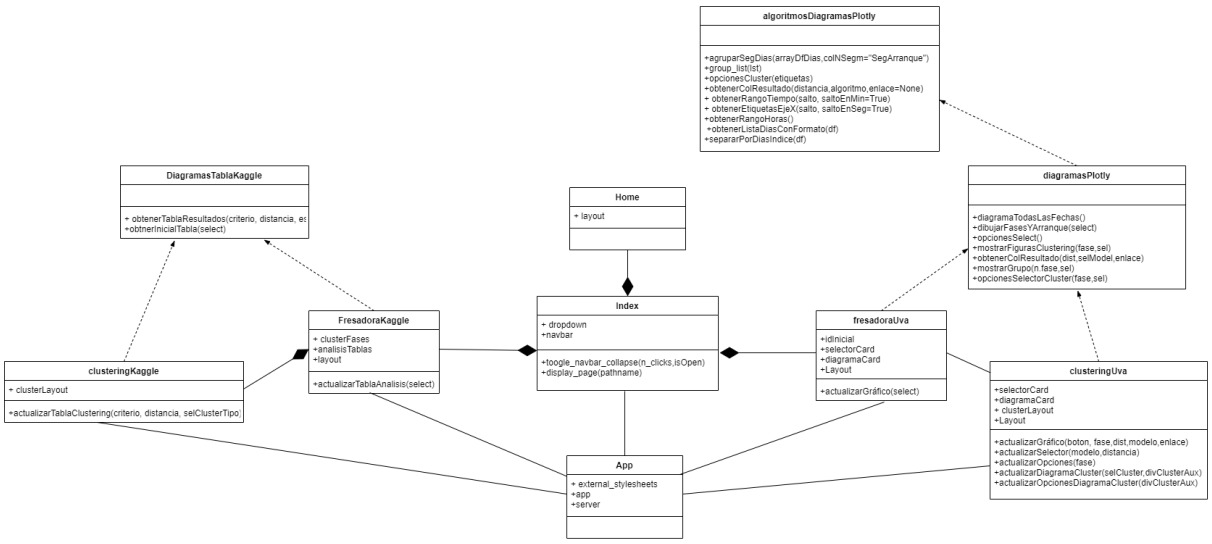


Figura 10.3: Diagrama de clases

10.2.4 Diagramas de Secuencia

Los diagramas de secuencia representan la interacción del usuario con el sistema. En el dashboard realizado la mayoría de estas interacciones consisten únicamente en seleccionar

una opción ya sea por un botón u otro elemento. Por ello solo se ha realizado un diagrama de secuencia del caso de uso que requiere de más interacción entre el usuario y el sistema para realizarse.

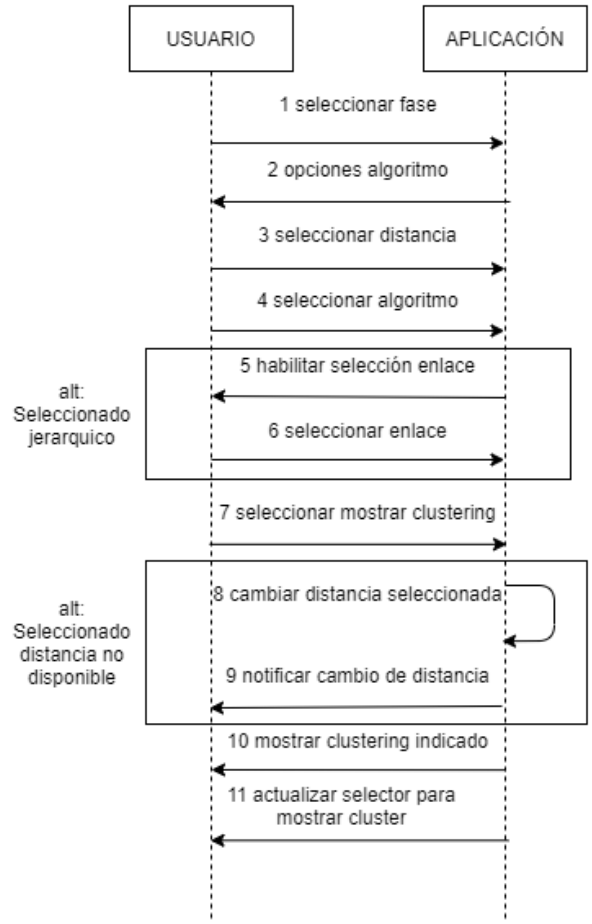


Figura 10.4: Diagrama de secuencia de CU-05

10.2.5 Diseño de la Interfaz

Similar al caso del diagrama de secuencia el diseño de la interfaz es bastante simple y contienen pocos elementos con pocos eventos. Por ello solo se ha realizado un diagrama.

Las páginas de las que no se han realizado un diagrama son la página de inicio, que tendrá solo dos botones para ir a una de las dos fresadoras, las páginas referentes a los datos de Kaggle que tendrán selectores y una tabla que se actualizará al cambiar cualquiera de los selectores y por último, la página de la otra fresadora con 3 diagramas, uno con todas las

series juntas en el que se podrá pulsar para quitar o poner un día y dos diagramas para mostrar las fases con un selector que controle el día mostrado.

Nombre	Clustering CIDAUT
Descripción	Página de visualización de los resultados del clustering a las fases de la fresadora mediante diagramas de las series de tiempo indicando los grupos.
Activación	Desde la pantalla fresadora CIDAUT pulsar botón clustering CIDAUT
Datos de entrada	
Eventos	• Desplegable algoritmos: al seleccionar jerárquico habilita el desplegable para indicar el tipo de enlace. • Mostrar clustering: actualiza los diagramas diarios que se muestran en pantalla con el clustering seleccionado y en caso de ser necesario actualiza dicha selección y notifica el cambio. • Mostrar Cluster: actualiza el diagrama que muestra un cluster del clustering previamente seleccionado.

Tabla 10.22: DI-01 Diseño de la ventana de resultados fresadora CIDAUT

10.3 Detalles de Implementación

En una aplicación Dash al ejecutar la clase con la instancia Dash se cargan todos los archivos, después, dash-renderer solicita el diseño inicial junto con las devoluciones de llamadas. Esto implica que se carguen los datos al inicio.

Las llamadas de la aplicación que se han implementado para la actualización de gráficos y tablas a petición del usuario, se muestran a continuación:

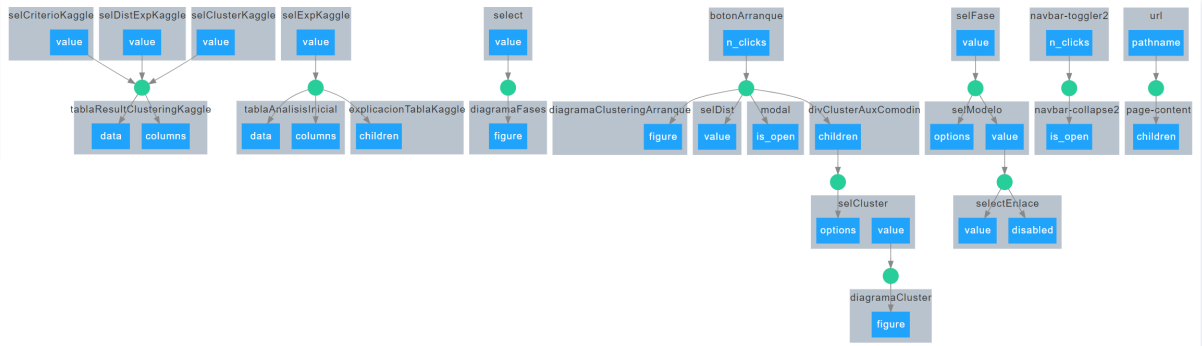


Figura 10.5: Visualización dashboard

Cada una de estas llamadas llama a un método específico para cumplir con la tarea.

10.4 Pruebas realizadas

Las pruebas realizadas son de caja negra probando opciones y comprobando que se obtenía el resultado esperado.

Las pruebas han sido muy simples y se adjunta una única prueba del CU-05, que es él que requiere de más comprobaciones.

P-01	Algoritmo aplicado con solo una distancia
Propósito	Comprobar que al intentar mostrar los resultados de un algoritmo con una distancia que no se aplicó el sistema cambia la distancia para mostrar un resultado.
Prerrequisitos	Tener seleccionado miKmedoides, miDbscan o miOptics con distancia euclidiana o K-Means o enlace Ward con distancia DTW.
Datos de entrada	fase, algoritmo, distancia y enlace seleccionados.
Resultado esperado	distancia seleccionada cambiada, muestra por pantalla del clustering con la selección cambiada y mensaje de aviso por pantalla.
Resultado obtenido	distancia seleccionada cambiada, muestra por pantalla del clustering con la selección cambiada y mensaje de aviso por pantalla.

Tabla 10.23: P-01 prueba visualización de algoritmo aplicado con solo una distancia

10.5 Manual de Instalación

La aplicación además de Python, Numpy y Pandas, requiere instalar Dash y los componentes de Dash Bootstrap.

El entorno utilizado para realizar el tablero se ha exportado a un fichero entornoTFG-dash.yml.

10.6 Manual de Usuario

Al entrar en la aplicación se accede a la pantalla de inicio que permite acceder a las dos fresadoras analizadas mediante dos botones centrales o con el menú de la barra de navegación. El menú de navegación se mantiene en toda la aplicación permitiendo cambiar de fresadora o volver a la página de inicio en cualquier momento.

Si se accede a fresadora CIDAUT, se cambia a una página donde se muestra el gráfico de corriente de todos los días y un gráfico individual donde se muestran tanto las fases como el arranque. En el gráfico principal se pueden quitar días pulsando en la leyenda del gráfico que se muestra en el lateral y en el diagrama individual se puede cambiar el día que se muestra a través de un selector. Esta página contiene un botón para ir a la página

de clustering.

En la página de clustering se permite seleccionar el clustering que se desea mostrar indicando a través de un menú distancia, fase, algoritmo y enlace (en caso de ser necesario) si la selección sufre cambios se le indicará al usuario. El usuario tras seleccionar un clustering que mostrar, podrá observar de forma individual un cluster específico de dicha agrupación a través de un menú donde aparecerán los clusters junto con el número de subsecuencias que contiene cada uno. Para volver a la pantalla de inicio se podrá usar el menú de navegación.

En el caso de haber seleccionado la otra fresadora se mostrará una tabla con resultados de salida de los experimentos. La tabla se podrá cambiar por otras que muestren otras características . Pulsando la pestaña de clustering, se podrá observar los resultados obtenidos en los clustering realizados a través de tablas que mostrarán la distribución de las muestras de cada cluster obtenido según algún criterio de salida indicado por el usuario. El usuario podrá cambiar el tipo, la distancia y el criterio de la tabla que se muestra.

10.7 Visualización de los datos y los resultados

En este apartado se adjuntan capturas de pantalla de cada una de las páginas del dashboard realizado .

Figura 10.6: Visualización dashboard: Inicio

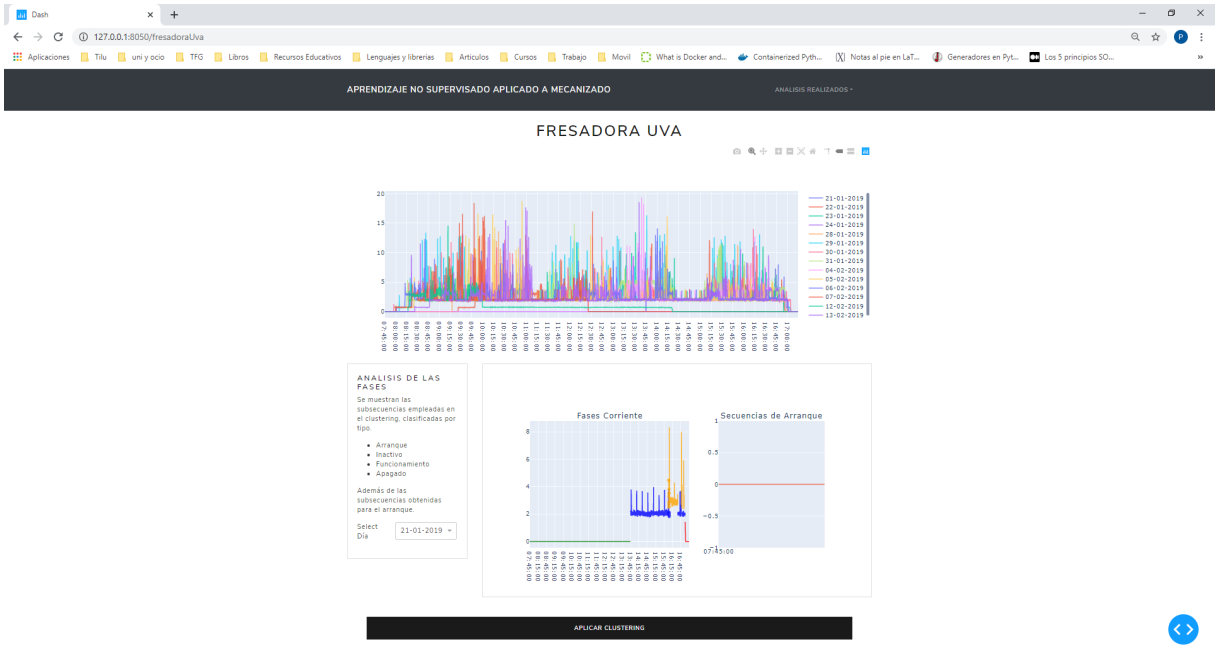


Figura 10.7: Visualización dashboard: Análisis CIDAUT

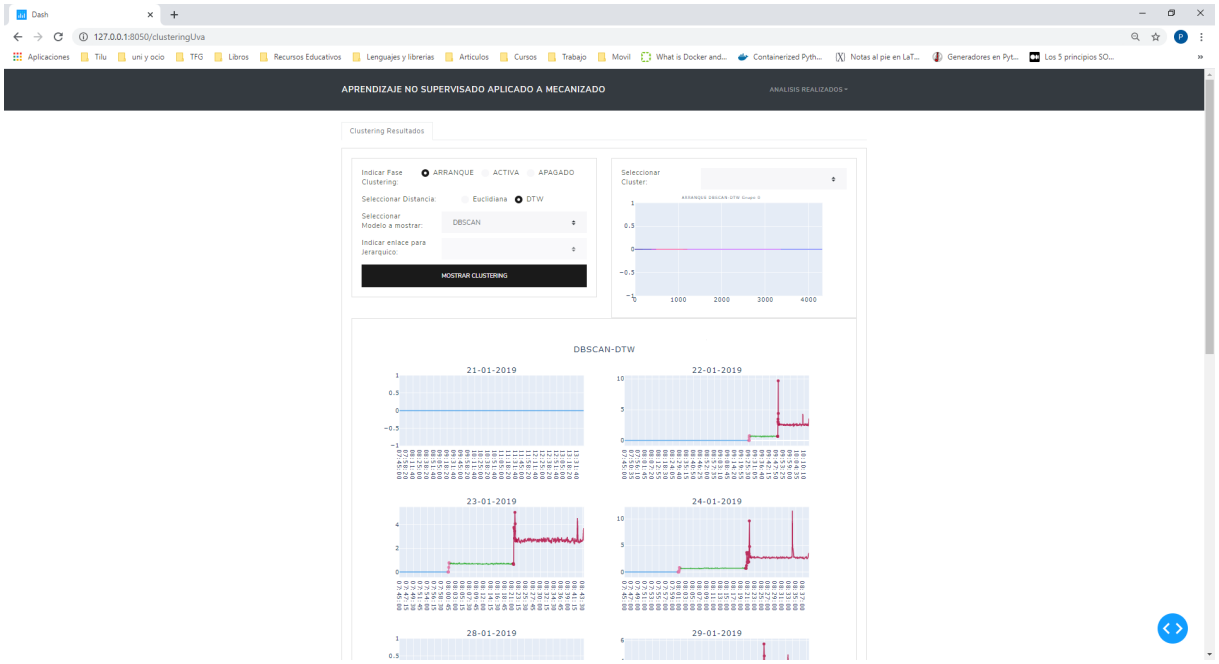
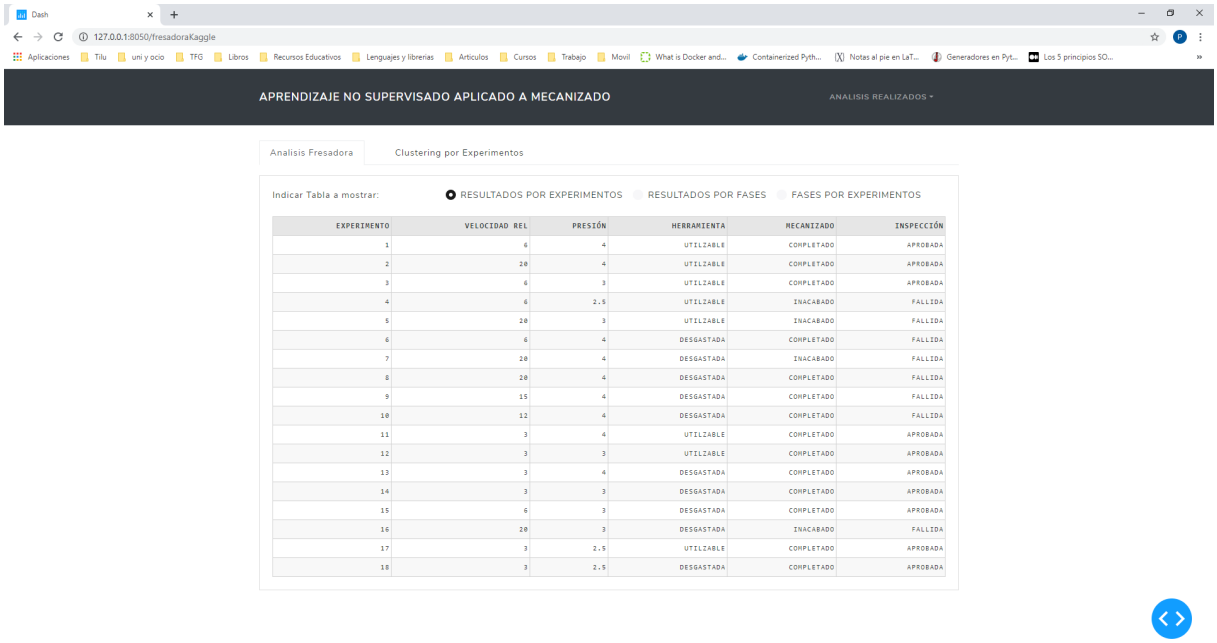
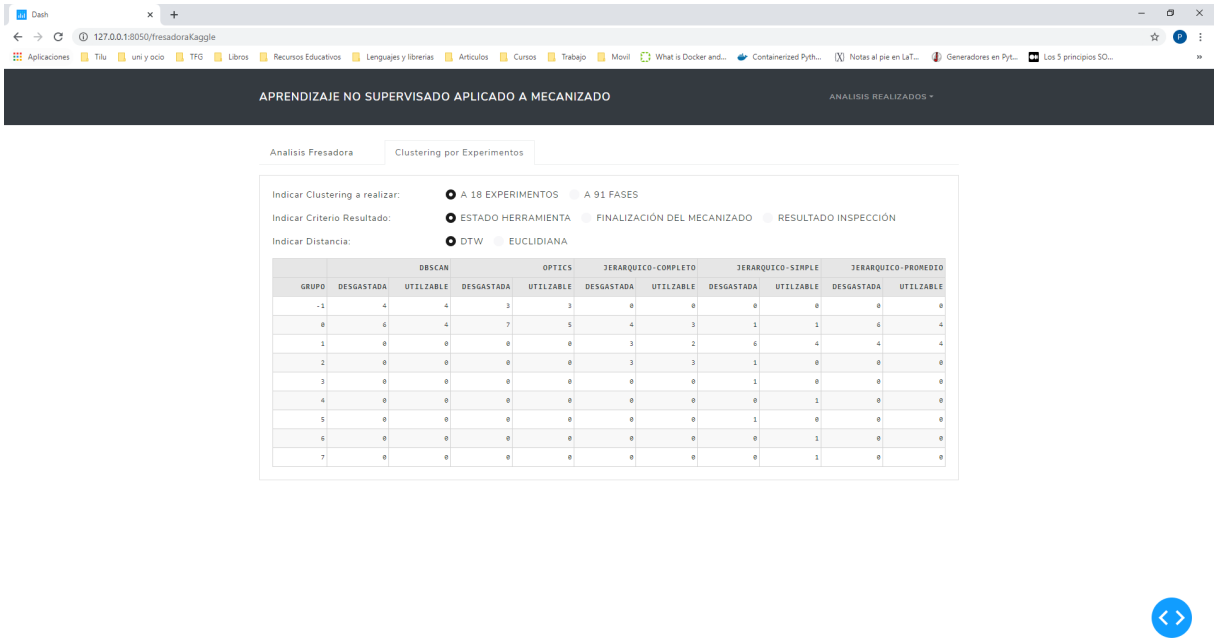


Figura 10.8: Visualización dashboard: Clustering CIDAUT



EXPERIMENTO	VELOCIDAD REL	PRESIÓN	HERRAMIENTA	MECANIZADO	INSPECCIÓN
1	6	4	UTILIZABLE	COMPLETADO	APROBADA
2	20	4	UTILIZABLE	COMPLETADO	APROBADA
3	6	3	UTILIZABLE	COMPLETADO	APROBADA
4	6	2.5	UTILIZABLE	INACABADO	FALLIDA
5	20	3	UTILIZABLE	INACABADO	FALLIDA
6	6	4	DESGASTADA	COMPLETADO	FALLIDA
7	20	4	DESGASTADA	INACABADO	FALLIDA
8	20	4	DESGASTADA	COMPLETADO	FALLIDA
9	15	4	DESGASTADA	COMPLETADO	FALLIDA
10	12	4	DESGASTADA	COMPLETADO	FALLIDA
11	3	4	UTILIZABLE	COMPLETADO	APROBADA
12	3	3	UTILIZABLE	COMPLETADO	APROBADA
13	3	4	DESGASTADA	COMPLETADO	APROBADA
14	3	3	DESGASTADA	COMPLETADO	APROBADA
15	6	3	DESGASTADA	COMPLETADO	APROBADA
16	20	3	DESGASTADA	INACABADO	FALLIDA
17	3	2.5	UTILIZABLE	COMPLETADO	APROBADA
18	3	2.5	DESGASTADA	COMPLETADO	APROBADA

Figura 10.9: Visualización dashboard: Análisis Kaggle



	DBSCAN		OPTICS		JERARQUICO-COMPLETO		JERARQUICO-SIMPLE		JERARQUICO-PROMEDIO	
GRUPO	DESGASTADA	UTILIZABLE	DESGASTADA	UTILIZABLE	DESGASTADA	UTILIZABLE	DESGASTADA	UTILIZABLE	DESGASTADA	UTILIZABLE
-1	4	4	3	3	0	0	0	0	0	0
0	6	4	7	5	4	3	1	1	6	4
1	0	0	0	0	3	2	6	4	4	4
2	0	0	0	0	3	3	1	0	0	0
3	0	0	0	0	0	0	1	0	0	0
4	0	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	1	0	0
7	0	0	0	0	0	0	0	1	0	0

Figura 10.10: Visualización dashboard: Clustering Kaggle

Parte IV

Parte Final

Capítulo 11

Conclusiones

En este trabajo se ha presentado un estudio teórico y práctico de técnicas de aprendizaje automático que se pueden aplicar a datos de monitorización industrial principalmente clustering y técnicas de detección de anomalías. La aplicación de estas técnicas en los datos de monitorización pueden ser de interés para la industria 4.0 de la que se han repasado sus objetivos, principios y aplicaciones (principalmente de los pronósticos y de la estimación de la vida útil de la herramienta).

Los datos de monitorización son secuenciales y dependen del tiempo por lo que, tanto en clustering como en detección de anomalías se ha indicado cómo aplicarlos con este tipo de datos.

Finalmente, se han aplicado algoritmos de clustering típicos y las distancias más utilizadas en series de tiempo para analizar el comportamiento de dos fresadoras distintas. En ambos conjuntos, se han obtenido resultados aceptables.

11.1 Conclusiones CIDAUT

De las dos fresadoras la que más tiempo ha requerido y la que ha desencadenado la mayor parte del estudio ha sido la fresadora del CIDAUT.

En parte esto se ha debido a la dificultad del problema, debido a la falta de información sobre los datos. No solo se desconocía sus resultados (cuáles de los datos eran anómalos), tampoco se conocía ningún comportamiento normal (salvo el relativo al arranque) y el comportamiento de la fresadora era considerablemente diferente cada uno de los días

examinados. Además tampoco había información sobre la actividad que había realizado la fresadora, lo que habría facilitado la obtención de diferentes firmas de ciclos de fabricación a las que se podría haber aplicado el clustering.

A todo esto hay que sumarle la dependencia temporal de los datos y la falta de homogeneización en la toma de mediciones que dificulto la aplicación de una gran parte de técnicas pensadas para datos estáticos o para datos dinámicos pero con muestreo regular.

Tras todo el estudio realizado creo que lo más relevante para este problema es el clustering de las subsecuencias de la serie de tiempo más que el de puntos de serie de tiempo o el de la serie de tiempo completa.

De las distancias estudiadas la distancia DTW me parece la más indicada dado que interesa similitud en forma más que similitud en tiempo y por la dificultad para determinar firmas de comportamiento de la fresadora. Estas secuencias además tendrán probablemente diferentes longitudes. Por ello la distancia DTW presenta una ventaja frente a aplicar la distancia euclidiana, a pesar de que hay más métodos capaces de aplicar esta distancia.

Después del trabajo realizado, creo que se pueden llegar a obtener unos resultados medianamente decentes dividiendo los datos por días y aplicando un muestreo PAA. El muestreo permite obtener un muestreo regular de los datos, ampliando con ello el número de posibles técnicas a aplicar. Aunque se pierda parte de información que afectaría sobre todo a encontrar anomalías puntuales, se ha observado que la forma de la señal se mantiene bastante bien y facilitaría encontrar tanto secuencias anómalas como patrones relevantes que se vayan repitiendo.

Una vez realizado el muestreo PAA habría que realizar una primera segmentación de los datos en diferentes secuencias con un comportamiento común. Aunque los métodos estudiados no parecían dar buenos resultados, el método aplicado a partir de la corriente estable de la fresadora parece ser la mejor opción en este caso.

Una vez obtenidas estas secuencias quedaría finalmente aplicar el clustering. Para el arranque y el apagado de la máquina los mejores resultados se obtuvieron con el algoritmo OPTICS con la distancia DTW, sin embargo para el resto del trabajo de la fresadora, este algoritmo presenta demasiado ruido.

En las horas de trabajo parece ser más conveniente aplicar un algoritmo de clustering que agrupe todas las secuencias en algún grupo. De los métodos probados y a raíz de los resultados los más indicados parecen ser el clustering jerárquico con enlace Ward (en caso

de querer identificar diferentes patrones) o completo (en caso de buscar (comportamientos donde la fresadora parece que estuvo inactiva), ambos con distancia euclidiana.

Sin embargo estos resultados son solo una primera aproximación al problema y habría que seguir probando métodos y ampliando el estudio para encontrar un método que consiga dar una mejor solución.

11.2 Conclusiones KAGGLE

A raíz de las dificultades encontradas en la fresadora del CIDAUT, se amplió el estudio con otros datos obtenidos de Kaggle para poder comprobar la aplicabilidad de los métodos probados.

Entre las ventajas que presentaban estos datos, se encuentra un muestreo regular de las mediciones de la fresadora y conocimiento de las fases y mediciones inexactas lo que facilitó el filtrado y la preparación de los datos.

Los resultados mostraron no ser significativos en el caso de aplicar el clustering a las muestras y revelaron la necesidad de aplicar el clustering a secuencias de datos ya sean de experimentos o de fases. Los resultados muestran pocas diferencias entre los obtenidos por fases y por experimentos. Esto podría indicar un comportamiento similar entre las fases de un mismo experimento, además la salida de los datos es referente a los experimentos. Por ello aunque el clustering por fases aporta mayor granularidad considero que tiene más sentido en este caso aplicar el clustering a los experimentos.

Similar a lo observado en el CIDAUT aunque el muestreo en este caso es regular, el número de muestras de cada experimento es diferente. Esto implica una ventaja de aplicar la distancia DTW frente a la distancia euclidiana que requiere transformar los datos para que tengan la misma longitud para poder ser aplicada. Debido además a la obtención de resultados similares con ambas distancias, considero más recomendable aplicar la distancia DTW.

Finalmente se han obtenido buenos resultados con los algoritmos aplicados, sobre todo con el algoritmo OPTICS y la distancia DTW, este algoritmo parece ser el indicado en este caso.

11.3 Líneas de Trabajo Futuro

Centrándonos en los datos estudiados se podrían hacer multitud de estudios. He probado solo los algoritmos más típicos de clustering pero existen otros muchos de algoritmos que se podrían estudiar con estos datos. A pesar de haber estudiado métodos de detección de anomalías no se han aplicado a ninguno de los conjuntos de datos y creo que serían aplicables; sobre todo creo que serían útiles en los datos de la fresadora univariable donde a pesar de haber usado solo una variable las mediciones parecen tener un comportamiento más variado.

Además en el análisis de la fresadora del CIDAUT se ha limitado el estudio a la corriente y podría ser útil usar datos de la corriente y el acelerómetro aunque eso reduzca el número de datos a utilizar. También se podrían investigar y probar otros métodos de segmentación de series de tiempo y representaciones que pudieran dar mejor resultado y del mismo modo probar con más medidas de similitud.

Creo que podría obtenerse buenos resultados si tras lograr una buena segmentación de los datos se les aplicara algún método de reducción de la dimensionalidad antes de agrupar los datos. Hice algunas pruebas con la segmentación utilizada y algoritmos de reducción de dimensionalidad no lineal (proporcionados por la librería Sklearn) y algunos como T-SNE parecía que podrían dar buenos resultados.

Una parte de este trabajo ha sido la investigación referente al perfil de matriz, los shapelets y los motivos de series de tiempo. De forma práctica me limite a un par de pruebas en los datos de la corriente y creo que podrían llegar a dar buenos resultados tanto para descubrir firmas de pieza como para detectar secuencias anómalas o al menos sería interesante ampliar su estudio tanto teórico como práctico para comprobar su aplicación a este problema.

A pesar de todo esto no creo que ninguno de los conjuntos de datos sirvan para estimar la vida útil de la herramienta de corte. Aunque con más datos, creo que se podrían utilizar para obtener una predicción sobre ellos.

Finalmente en el dashboard, se podrían añadir funcionalidades, para que además de permitir visualizar los resultados permita realizar nuevos clustering a selección del usuario o incluso otras técnicas.

Bibliografía

[1] Issam Abu-Mahfouz y Amit Banerjee. "Drill Wear Feature Identification under Varying Cutting Conditions Using Vibration and Cutting Force Signals and Data Mining Techniques". En: *Procedia Computer Science* 36 (2014), págs. 556-563. ISSN: 18770509. DOI: 10.1016/j.procs.2014.09.054. URL: https://linkinghub.elsevier.com/retrieve/pii/S1877050914013039 (visitado 15-06-2019).

[2] Charu C. Aggarwal. *Data Mining: The Textbook*. Springer Publishing Company, Incorporated, 2015. ISBN: 3319141414.

[3] Charu C. Aggarwal y Chandan K. Reddy. *Data Clustering: Algorithms and Applications*. 1st. Chapman Hall/CRC, 2013. ISBN: 1466558210.

[4] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi y Teh Ying Wah. "Time-series clustering – A decade review". En: *Information Systems* 53 (oct. de 2015), págs. 16-38. ISSN: 03064379. DOI: 10.1016/j.is.2015.04.007. URL: https://linkinghub.elsevier.com/retrieve/pii/S0306437915000733 (visitado 15-06-2019).

[5] M. Ali y col. "Clustering and Classification for Time Series Data in Visual Analytics: A Survey". En: *IEEE Access* 7 (2019), págs. 181314-181338.

[6] Mihael Ankerst y col. "OPTICS: Ordering Points to Identify the Clustering Structure". En: *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*. SIGMOD '99. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 1999, págs. 49-60. ISBN: 1581130848. DOI: 10.1145/304182.304187. URL: https://doi.org/10.1145/304182.304187.

[7] Mihael Ankerst y col. "OPTICS: Ordering Points to Identify the Clustering Structure". En: *SIGMOD Rec.* 28.2 (jun. de 1999), págs. 49-60. ISSN: 0163-5808. DOI: 10.1145/304181.304187. URL: https://doi.org/10.1145/304181.304187.

[8] Simon Duque Anton y col. "Time is of the Essence: Machine Learning-based Intrusion Detection in Industrial Time Series Data". en. En: *2018 IEEE International Conference on Data Mining Workshops (ICDMW)* (nov. de 2018). arXiv: 1809.07500, págs. 1-6. DOI: 10.1109/ICDMW.2018.00008. URL: http://arxiv.org/abs/1809.07500 (visitado 12-08-2020).

[9] Vepa Atamuradov y col. "Prognostics and Health Management for Maintenance Practitioners - Review, Implementation and Tools Evaluation". En: 8.060 (2017), págs. 1-31.

[10] Abla Chouni Benabdellah, Asmaa Benghabrit e Imane Bouhaddou. "A survey of clustering algorithms for an industrial context". En: *Procedia Computer Science* 148 (2019), págs. 291-302. ISSN: 18770509. DOI: 10.1016/j.procs.2019.01.022. URL: https://linkinghub.elsevier.com/retrieve/pii/S1877050919300225 (visitado 30-06-2019).

[11] Ane Blázquez-García y col. "A review on outlier/anomaly detection in time series data". en. En: *arXiv:2002.04236 [cs, stat]* (feb. de 2020). URL: http://arxiv.org/abs/2002.04236 (visitado 01-05-2020).

[12] Raghavendra Chalapathy y Sanjay Chawla. "Deep Learning for Anomaly Detection: A Survey". en. En: *arXiv:1901.03407 [cs, stat]* (ene. de 2019). URL: http://arxiv.org/abs/1901.03407 (visitado 03-08-2019).

[13] Varun Chandola, Arindam Banerjee y Vipin Kumar. "Anomaly detection: A survey". en. En: *ACM Computing Surveys* 41.3 (jul. de 2009), págs. 1-58. ISSN: 03600300. DOI: 10.1145/1541880.1541882. URL: http://portal.acm.org/citation.cfm?doid=1541880.1541882 (visitado 24-06-2019).

[14] Pete Chapman y col. *CRISP-DM 1.0 Step-by-step data mining guide*. Inf. téc. The CRISP-DM consortium, 2000. URL: https://maestria-datamining-2010.googlecode.com/svn-history/r282/trunk/dmct-teorica/tp1/CRISPWP-0800.pdf.

[15] J.R. Chen. "Making Subsequence Time Series Clustering Meaningful". En: *Fifth IEEE International Conference on Data Mining (ICDM'05)*. Houston, TX, USA: IEEE, 2005, págs. 114-121. ISBN: 978-0-7695-2278-4. DOI: 10.1109/ICDM.2005.91. URL: http://ieeexplore.ieee.org/document/1565669/ (visitado 24-10-2019).

[16] Lucas Santos Dalenogare y col. "The expected contribution of Industry 4.0 technologies for industrial performance". En: *International Journal of Production Economics* 204 (oct. de 2018), págs. 383-394. ISSN: 09255273. DOI: 10.1016/j.ijpe.2018.08.019. URL: https://linkinghub.elsevier.com/retrieve/pii/S0925527318303372 (visitado 17-07-2019).

[17] Alberto Diez-Olivan y col. "Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0". En: *Information Fusion* 50 (2019), págs. 92-111. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2018.10.005.

[18] Philippe Esling y Carlos Agon. "Time-Series Data Mining". En: *ACM Comput. Surv.* 45.1 (dic. de 2012). ISSN: 0360-0300. DOI: 10.1145/2379776.2379788. URL: https://doi.org/10.1145/2379776.2379788.

[19] Martin Ester y col. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". En: KDD'96 (1996), págs. 226-231.

[20] Tak-chung Fu. "A review on time series data mining". En: *Engineering Applications of Artificial Intelligence* 24.1 (feb. de 2011), págs. 164-181. ISSN: 09521976. DOI: 10.1016/j.engappai.2010.09.007. URL: https://linkinghub.elsevier.com/retrieve/pii/S0952197610001727 (visitado 21-06-2019).

[21] Óscar Marbán Gallego. "Modelo matemático paramétrico de estimación para proyectos de data mining". 2003. URL: http://oa.upm.es/282/.

[22] Guojun Gan, Chaoqun Ma y Jianhong Wu. *Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability)*. USA: Society for Industrial y Applied Mathematics, 2007. ISBN: 0898716233.

[23] Mert Onuralp Gokalp y col. "Big Data for Industry 4.0: A Conceptual Framework". En: *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*. Las Vegas, NV, USA: IEEE, dic. de 2016, págs. 431-434. ISBN: 978-1-5090-5510-4. DOI: 10.1109/CSCI.2016.0088. URL: http://ieeexplore.ieee.org/document/7881381/ (visitado 10-07-2019).

[24] Markus Goldstein y Seiichi Uchida. "A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data". en. En: *PLOS ONE* 11.4 (abr. de 2016). Ed. por Dongxiao Zhu, e0152173. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0152173. URL: https://dx.plos.org/10.1371/journal.pone.0152173 (visitado 17-07-2019).

[25] David Guijo-Rubio y col. "Time series clustering based on the characterisation of segment typologies". En: (2018). arXiv: 1810.11624 [cs.LG].

[26] Ville Hautamäki, Pekka Nykänen y P. Fränti. "Time-series clustering by approximate prototypes". En: *ICPR*. 2008.

[27] Yawei Hu y col. "Remaining Useful Life Model and Assessment of Mechanical Products: A Brief Review and a Note on the State Space Model Method". En: *Chinese Journal of Mechanical Engineering* 32.1 (dic. de 2019), pág. 15. ISSN: 1000-9345, 2192-8258. DOI: 10.1186/s10033-019-0317-y. URL: https://cjme.springeropen.com/articles/10.1186/s10033-019-0317-y (visitado 02-08-2020).

[28] A. K. Jain, M. N. Murty y P. J. Flynn. "Data clustering: a review". En: *ACM Computing Surveys (CSUR)* 31.3 (sep. de 1999), págs. 264-323. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/331499.331504. URL: http://dl.acm.org/doi/10.1145/331499.331504 (visitado 07-03-2020).

[29] Micheline Kamber Jiawei Han y Jian Pei Professor. *Data Mining: Concepts and Techniques*. Morgan Kaufmannger, 2011. ISBN: 9780123814791.

[30] Eamonn Keogh, Jessica Lin y Wagner Truppel. "Clustering of Time Series Subsequences is Meaningless: Implications for Previous and Future Research". En: *Proceedings of the Third IEEE International Conference on Data Mining*. ICDM '03. USA: IEEE Computer Society, 2003, pág. 115. ISBN: 0769519784.

[31] Eamonn Keogh y col. "Segmenting Time Series: A Survey and Novel Approach". En: *In an Edited Volume, Data mining in Time Series Databases. Published by World Scientific*. Publishing Company, 1993, págs. 1-22.

[32] Maqbool Khan y col. "Big data challenges and opportunities in the hype of Industry 4.0". En: *2017 IEEE International Conference on Communications (ICC)*. Paris, France: IEEE, mayo de 2017, págs. 1-6. ISBN: 978-1-4673-8999-0. DOI: 10.1109/ICC.2017.7996801. URL: http://ieeexplore.ieee.org/document/7996801/ (visitado 10-07-2019).

[33] Dimitrios Kotsakos y col. "Time-Series Data Clustering". en. En: *Data Clustering*. Ed. por Charu C. Aggarwal y Chandan K. Reddy. 1.ª ed. Chapman y Hall/CRC, sep. de 2018, págs. 357-380. ISBN: 978-1-315-37351-5. DOI: 10.1201/9781315373515-15. URL: https://www.taylorfrancis.com/books/9781315362786/chapters/10.1201/9781315373515-15 (visitado 23-10-2019).

[34] Jay Lee, Hung-An Kao y Shanhu Yang. "Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment". En: *Procedia CIRP* 16 (2014), págs. 3-8. ISSN: 22128271. DOI: 10.1016/j.procir.2014.02.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S2212827114000857 (visitado 14-07-2019).

[35] Yaguo Lei y col. "Machinery health prognostics: A systematic review from data acquisition to RUL prediction". En: *Mechanical Systems and Signal Processing* 104 (mayo de 2018), págs. 799-834. ISSN: 08883270. DOI: 10.1016/j.ymssp.2017.11.016. URL: https://linkinghub.elsevier.com/retrieve/pii/S0888327017305988 (visitado 02-08-2020).

[36] Guang Li y col. "Tool Breakage Detection using Deep Learning". En: *CoRR* abs/1808.05347 (2018). arXiv: 1808.05347. URL: http://arxiv.org/abs/1808.05347.

[37] R. G. Lins y col. "A novel methodology for retrofitting CNC machines based on the context of industry 4.0". En: *2017 IEEE International Systems Engineering Symposium (ISSE)* (2017), págs. 1-6. DOI: 10.1109/SysEng.2017.8088293.

[38] Yanchi Liu y col. "Understanding of Internal Clustering Validation Measures". En: *2010 IEEE International Conference on Data Mining*. 2010 IEEE 10th International Conference on Data Mining (ICDM). IEEE, dic. de 2010, págs. 911-916. ISBN: 978-1-4244-9131-5. DOI: 10.1109/ICDM.2010.35. URL: http://ieeexplore.ieee.org/document/5694060/ (visitado 16-12-2019).

[39] Harshada C Mandhare y S R Idate. "A Comparative Study of Cluster Based Outlier Detection, Distance Based Outlier Detection and Density Based Outlier Detection Techniques". en. En: (2017), pág. 5.

[40] J. B. Robles-Ocampo P. Y. Sevilla-Camacho G. Herrera-Ruiz y J. C. Jáuregui-Correa. "Tool breakage detection in CNC high-speed milling based in feed-motor current signals". En: *The International Journal of Advanced Manufacturing Technology* 53.9-12 (2011), págs. 1141-1148. DOI: https://doi.org/10.1007/s00170-010-2907-9.

[41] A.C. Pereira y F. Romero. "A review of the meanings and the implications of the Industry 4.0 concept". En: *Procedia Manufacturing* 13 (2017), págs. 1206-1214. ISSN: 23519789. DOI: 10.1016/j.promfg.2017.09.032. URL: https://linkinghub.elsevier.com/retrieve/pii/S2351978917306649 (visitado 10-07-2019).

[42] Thanawin Rakthanmanon y col. "MDL-based time series clustering". En: *Knowledge and Information Systems* 33.2 (nov. de 2012), págs. 371-399. ISSN: 0219-1377, 0219-3116. DOI: 10.1007/s10115-012-0508-7. URL: http://link.springer.com/10.1007/s10115-012-0508-7 (visitado 09-02-2020).

[43] Thanawin Rakthanmanon y col. "Time Series Epenthesis: Clustering Time Series Streams Requires Ignoring Some Data". En: *2011 IEEE 11th International Conference on Data Mining* (dic. de 2011), págs. 547-556. DOI: 10.1109/ICDM.2011.146. URL: http://ieeexplore.ieee.org/document/6137259/ (visitado 20-02-2020).

[44] Saeed Ramezani, Alireza Moini y Mohamad Riahi. "Prognostics and Health Management in Machinery: A Review of Methodologies for RUL prediction and Roadmap". En: *International Journal of Industrial Engineering* 6.1 (2019), pág. 23.

[45] Khashayar Danesh Narooeiand Rizauddin Ramli. "Application of artificial intelligence methods of tool path optimization in CNC machines: A review". En: *Research Journal of Applied Sciences, Engineering and Technology* 8.6 (2014), págs. 746-754.

[46] Theofanis P. Raptis, Andrea Passarella y Marco Conti. "Data Management in Industry 4.0: State of the Art and Open Challenges". En: *arXiv:1902.06141 [cs]* (feb. de 2019). URL: http://arxiv.org/abs/1902.06141 (visitado 10-07-2019).

[47] Sebastian Raschka y Vahid Mirjalili. *Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and TensorFlow, 2nd Edition*. 2nd. Packt Publishing, 2017. ISBN: 1787125939.

[48] Chotirat Ann Ratanamahatana y Eamonn J. Keogh. "Everything you know about Dynamic Time Warping is Wrong". En: 2004.

[49] Sura Rodpongpun, Vit Niennattrakul y Chotirat Ann Ratanamahatana. "Selective Subsequence Time Series clustering". En: *Knowledge-Based Systems* 35 (nov. de 2012), págs. 361-368. ISSN: 09507051. DOI: 10.1016/j.knosys.2012.04.022. URL: https://linkinghub.elsevier.com/retrieve/pii/S0950705112001189 (visitado 23-06-2019).

[50] D. Rodríguez y col. "Estudio del modelo paramétrico DMCoMo de estimación de proyectos de explotación de información". En: 2011.

[51] C. Santos y col. "Towards Industry 4.0: an overview of European strategic roadmaps". En: *Procedia Manufacturing* 13 (2017), págs. 972-979. ISSN: 23519789. DOI: 10.1016/j.promfg.2017.09.093. URL: https://linkinghub.elsevier.com/retrieve/pii/S235197891730728X (visitado 10-07-2019).

[52] Joan Serrà y Josep Lluis Arcos. "An Empirical Evaluation of Similarity Measures for Time Series Classification". en. En: *Knowledge-Based Systems* 67 (sep. de 2014). ISSN: 09507051. DOI: 10.1016/j.knosys.2014.04.035. URL: http://arxiv.org/abs/1401.3973 (visitado 07-07-2019).

[53] Khurram Shahzad y Mattias Onils. "Condition Monitoring in Industry 4.0-Design Challenges and Possibilities: A Case Study". En: *2018 Workshop on Metrology for Industry 4.0 and IoT*. Brescia: IEEE, abr. de 2018, págs. 101-106. ISBN: 978-1-5386-2497-5. DOI: 10.1109/METROI4.2018.8428306. URL: https://ieeexplore.ieee.org/document/8428306/ (visitado 10-07-2019).

[54] Sharon Sun. *CNC Mill Tool Wear Variational CNC machining data*. URL: https://www.kaggle.com/shasun/tool-wear-detection-in-cnc-mill.

[55] H. T. T. Thuy, D. T. Anh y V. T. N. Chau. "A Novel Method for Time Series Anomaly Detection based on Segmentation and Clustering". En: *2018 10th International Conference on Knowledge and Systems Engineering (KSE)*. 2018, págs. 276-281.

[56] Tiedo Tinga y Richard Loendersloot. "Aligning PHM, SHM and CBM by understanding the physical system failure behaviour". En: (2014), pág. 10.

[57] D.A. Tobon-Mejia, K. Medjaher y N. Zerhouni. "CNC machine tool's wear diagnostic and prognostic by using dynamic Bayesian networks". en. En: *Mechanical Systems and Signal Processing* 28 (abr. de 2012), págs. 167-182. ISSN: 08883270. DOI: 10.1016/j.ymssp.2011.10.018. URL: https://linkinghub.elsevier.com/retrieve/pii/S0888327011004493 (visitado 24-04-2020).

[58] Charles Truong, Laurent Oudre y Nicolas Vayatis. *ruptures: change point detection in Python*. 2018. arXiv: 1801.00826 [stat.CO].

[59] Liudmila Ulanova, Nurjahan Begum y Eamonn Keogh. "Scalable Clustering of Time Series with U-Shapelets". en. En: *Proceedings of the 2015 SIAM International Conference on Data Mining*. Society for Industrial y Applied Mathematics, jun. de 2015, págs. 900-908. ISBN: 978-1-61197-401-0. DOI: 10.1137/1.9781611974010.101.

[60] Saurabh Vaidya, Prashant Ambad y Santosh Bhosle. "Industry 4.0 – A Glimpse". En: *Procedia Manufacturing* 20 (2018), págs. 233-238. ISSN: 23519789. DOI: 10.1016/j.promfg.2018.02.034. URL: https://linkinghub.elsevier.com/retrieve/pii/S2351978918300672 (visitado 10-07-2019).

[61] José Luis del Val Román. *Industria 4.0: la transformación digital de la industria*. Facultad de Ingeniería de la Universidad de Deusto.

[62] Gregory W. Vogl, Brian A. Weiss y Moneer Helu. "A review of diagnostic and prognostic capabilities and best practices for manufacturing". En: *Journal of Intelligent Manufacturing* 30.1 (ene. de 2019), págs. 79-95. ISSN: 0956-5515, 1572-8145. DOI: 10.1007/s10845-016-1228-8. URL: http://link.springer.com/10.1007/s10845-016-1228-8 (visitado 27-04-2020).

[63] Xiaoyue Wang y col. "Experimental comparison of representation methods and distance measures for time series data". En: *Data Mining and Knowledge Discovery* 26.2 (mar. de 2013), págs. 275-309. ISSN: 1384-5810, 1573-756X. DOI: 10.1007/s10618-012-0250-5. URL: http://link.springer.com/10.1007/s10618-012-0250-5 (visitado 04-08-2019).

[64] Xiaozhe Wang, Kate Smith y Rob Hyndman. "Characteristic-Based Clustering for Time Series Data". en. En: *Data Mining and Knowledge Discovery* 13.3 (sep. de 2006), págs. 335-364. ISSN: 1384-5810, 1573-756X. DOI: 10.1007/s10618-005-0039-x. URL: http://link.springer.com/10.1007/s10618-005-0039-x (visitado 12-04-2020).

[65] T. Warren Liao. "Clustering of time series data—a survey". En: *Pattern Recognition* 38.11 (nov. de 2005), págs. 1857-1874. ISSN: 00313203. DOI: 10.1016/j.patcog.2005.01.025. URL: https://linkinghub.elsevier.com/retrieve/pii/S0031320305001305 (visitado 24-06-2019).

[66] Ka-Chun Wong. "A Short Survey on Data Clustering Algorithms". En: *2015 Second International Conference on Soft Computing and Machine Intelligence (ISCMI)*. 2015 Second International Conference on Soft Computing and Machine Intelligence (ISCMI). IEEE, nov. de 2015, págs. 64-68. ISBN: 978-1-4673-9819-0. DOI: 10.1109/ISCMI.2015.10. URL: http://ieeexplore.ieee.org/document/7414675/ (visitado 24-06-2019).

[67] Dongkuan Xu y Yingjie Tian. "A Comprehensive Survey of Clustering Algorithms". En: *Annals of Data Science* 2.2 (jun. de 2015), págs. 165-193. ISSN: 2198-5804, 2198-5812. DOI: 10.1007/s40745-015-0040-1. URL: http://link.springer.com/10.1007/s40745-015-0040-1 (visitado 07-03-2020).

[68] Ye Yuan y col. "Artificial Intelligent Diagnosis and Monitoring in Manufacturing". En: (2018), pág. 18.

[69] J. Zakaria, A. Mueen y E. Keogh. "Clustering Time Series Using Unsupervised-Shapelets". En: *2012 IEEE 12th International Conference on Data Mining*. 2012, págs. 785-794.

[70] Lou Zhang. *Detecting CNC Anomalies with Unsupervised Learning (Part 1-4)*. 2018. URL: https://medium.com/machinemetrics-techblog/using-pca-and-clustering-to-detect-machine-anomalies-part-1-ba89f6a6a8cd.

[71] Ray Y. Zhong y col. "Intelligent Manufacturing in the Context of Industry 4.0: A Review". En: *Engineering* 3.5 (oct. de 2017), págs. 616-630. ISSN: 20958099. DOI: 10.1016/J.ENG.2017.05.015. URL: https://linkinghub.elsevier.com/retrieve/pii/S2095809917307130 (visitado 10-07-2019).

[72] Y. Zhu y col. "Matrix Profile VII: Time Series Chains: A New Primitive for Time Series Data Mining (Best Student Paper Award)". En: *2017 IEEE International Conference on Data Mining (ICDM)*. 2017, págs. 695-704. DOI: 10.1109/ICDM.2017.79.

[73] Seyedjamal Zolhavarieh, Saeed Aghabozorgi y Ying Wah Teh. "A Review of Subsequence Time Series Clustering". En: *The Scientific World Journal* 2014 (2014), págs. 1-19. ISSN: 2356-6140, 1537-744X. DOI: 10.1155/2014/312521. URL: http://www.hindawi.com/journals/tswj/2014/312521/ (visitado 30-09-2019).