

DEPARTAMENTO DE INTELIGENCIA ARTIFICIAL

Escuela Técnica Superior de Ingenieros Informáticos
Universidad Politécnica de Madrid

Tesis de Doctorado

***Clustering* probabilístico dinámico para la
búsqueda de patrones de degradación de
elementos de máquina en el ámbito del
*Industria 4.0***

Autor

Javier Díaz Rozo

Ingeniero Mecánico, Master en Ciencias

Directores de tesis

Pedro Larrañaga

Doctor en Informática

Concha Bielza

Doctor en Informática

2019

*A mi mujer Natale,
a mis hijas Aroa y Amaia,
y a mis padres Martha y Javier*

Agradecimientos

Después de muchos años fuera del mundo académico se ha presentado la oportunidad de continuar con mi formación en donde claramente he necesitado el apoyo de un número importante de personas a las cuales me gustaría agradecer.

En primer lugar a mis directores de tesis, Concha Bielza y Pedro Larrañaga, por su constante dedicación y consejo para ayudarme a cumplir los objetivos planteados. Han sido unos años en los que gracias a ellos y a su paciencia, he logrado aprender sobre un mundo generalmente ajeno a las personas que estamos dedicados a la industria. Ha sido un privilegio poder compartir y aprender tanto de ellos.

Por otro lado, agradecer a las empresas Ikergune y Aingura IIoT, y en general al Grupo Etxe-Tar, por el apoyo prestado en todos los sentidos durante estos años. En especial, a Patxi Samaniego, el principal promotor de este trabajo y a su vez, fuente constante de apoyo e inspiración, y a José Juan Gabilondo, por dedicarme tantas horas de sabiduría. A mis compañeros pasados y actuales, Ainhoa, Albert, Beñat, Carlos, Dani, Gaizka, Javier Miguel y Rubén por su profesionalidad y apoyo a la hora de sacar datos de máquinas, no siempre de la manera más cómoda. Un especial agradecimiento a Paula Yurrita por su constante apoyo desde hace muchos años y en especial, durante la etapa final de este trabajo.

Parte de este trabajo ha sido financiado por el Centro para el Desarrollo Tecnológico Industrial (CDTI) a través del proyecto CARES (TIC-20150093) y LearnIIoT (IDI-20180156); y parcialmente apoyado por el Ministerio de Economía y Competitividad de España a través del proyecto TIN2016-79684-P y por el Gobierno Regional de Madrid a través del proyecto S2013/ICE-2845-CASI-CAM-CM.

Y por último agradecer a mi familia, en especial a mi mujer, Natale, por su constante ayuda para alcanzar mis metas, por su energía y motivación, pero también por aguantarme tantos años explorando nuevos retos de manera constante. A mis hijas Aroa y Amaia por darme ese impulso de juventud que siempre me dan en todos los apartados de mi vida. A mi padre, por enseñarme a tener constantemente inquietudes y no tener miedo a explorarlas y a mi madre, por heredarme esa capacidad de devorar libros muy necesaria en esta etapa, aunque nunca podré llegar a su nivel.

Resumen

La Cuarta Revolución Industrial y en particular, los sistemas ciberfísicos (CPS), han abierto una amplia gama de oportunidades en términos de análisis de rendimiento. Estas oportunidades pueden ser aplicables a sistemas de diagnóstico y predicción de fallos pero también, pueden contribuir a la mejora del diseño de los productos y a la optimización de los procesos industriales. Las capacidades de comunicación de los CPS a alta velocidad permiten adquirir, pre-procesar y procesar los datos que se extraen, por ejemplo, de las máquinas, pilar fundamental de los procesos productivos. Como resultado, la degradación de los elementos de máquina sujetos a comportamientos dinámicos puede detectarse de una manera más rápida mediante el estudio de los patrones que forman sus principales variables de funcionamiento a lo largo del tiempo. Esto permite generar herramientas de monitorización de elementos productivos, aplicables principalmente al mantenimiento y al control de calidad.

Sin embargo, este tipo de aproximaciones trabajan con sensores que envían datos de procesos dinámicos a alta velocidad en donde no es fácil generar información útil en el momento adecuado. Una parte del problema se refiere al procesamiento de una gran cantidad de datos, mientras que los fenómenos dinámicos subyacentes relacionados con la máquina posiblemente evolucionen con el tiempo dando lugar a un *concept drift*. Esto se debe a factores como la degradación, algo completamente normal en los sistemas físicos. Como resultado, cualquier modelo de datos puede volverse obsoleto y es necesaria su constante actualización.

Para hacer frente a este problema, se propone una aproximación desde el aprendizaje automático no supervisado. Específicamente, el uso de algoritmos de *clustering* dinámicos. Para ello, se trabaja en una metodología que primero estudia el rendimiento de los algoritmos de *clustering* en aplicaciones industriales. Posteriormente, se seleccionan aquellos algoritmos que tengan la capacidad de aportar nuevo conocimiento relacionado con los elementos productivos y sus patrones de degradación. El siguiente paso es adaptar el algoritmo seleccionado al comportamiento dinámico de las máquinas y al trabajo con *data streams*, mucho más cercano a la realidad industrial. De esta manera, partiendo de algoritmos de *clustering* como: *K*-medias, jerárquico aglomerativo, espectral, propagación de afinidad y modelos de mixturas de Gaussianas, se selecciona este último tipo como el más apto para esta aplicación.

Se propone un nuevo algoritmo de aprendizaje no supervisado, denominado *clustering* probabilístico dinámico basado en mixturas de Gaussianas (GDPC). GDPC integra y adapta tres algoritmos conocidos para poder ser usados en escenarios dinámicos: el algoritmo de esperanza-maximización (EM) responsable de estimar los parámetros del modelo de mixturas

y el test de hipótesis de Page–Hinkley que junto con las cotas de Chernoff permiten detectar los *concept drift*.

A diferencia de otros métodos no supervisados, el modelo inducido por el GDPC proporciona las probabilidades de asignación de cada instancia a cada clúster o componente. Esto permite determinar, a través de un análisis con el *Brier score*, la robustez de esta asignación y su evolución una vez detectado un *concept drift*. El GDPC trabaja con una ventana óptima de datos reduciendo de manera importante las necesidades de potencia de cómputo. Sin embargo, el algoritmo requiere un conocimiento del dominio profundo con el fin de seleccionar correctamente los parámetros (por ejemplo, el número de componentes). Además, puede ser inestable debido a otro fenómeno encontrado comúnmente en datos industriales relacionado con fases no estacionarias que ocurren cuando los elementos cambian de estado y se estabilizan en su valor esperado.

De esta manera, se proponen una nueva versión con serie de mejoras con el objetivo de aumentar el grado de robustez del algoritmo ante estas problemáticas. Esta nueva versión, denominada GDPC+, introduce las siguientes mejoras: (a) la selección automática del número de componentes de la mixtura de Gaussianas en función del criterio de información Bayesiano; y (b) la estabilización debida a los efectos transitorios, no estacionarios, durante el *concept drift* gracias a la integración de la divergencia de Cauchy–Schwarz con el test de Dickey–Fuller aumentado. Por lo tanto, el GDPC+ tiene un mejor desempeño que el GDPC en términos del número de falsos positivos en aplicaciones altamente dinámicas.

El desarrollo de estos algoritmos ha sido validado con pruebas sobre *data stream* de origen sintético y también originados a partir de un banco de pruebas y una máquina-herramienta produciendo piezas reales, en este caso, cigüeñales de automoción. Estos resultados se han validado en términos de diferentes medidas como precisión, *recall*, especificidad y *F-score*. Adicionalmente, partiendo de los resultados de *clustering* de datos reales de máquina, se han desarrollado conjuntos de reglas inducidas mediante un algoritmo de clasificación supervisada con el fin de proporcionar información sobre el proceso subyacente y sus *concept drift* asociados.

Abstract

The Fourth Industrial Revolution and in particular, the cyber-physical systems (CPS), have opened a wide range of opportunities in terms of performance analysis. They can be applied to fault diagnosis and prediction systems but also to improve the design of industrial products and processes optimization. In this way, the CPS communication capabilities at high-speed allow us to acquire, pre-process and process the data extracted from, e.g., machines, fundamental part of production processes. As a result, the degradation of machine elements subject to dynamic behavior can be detected more quickly by studying the patterns that produce the main operating variables over time. This allows the generation of fundamental monitoring tools, mainly applicable to maintenance and quality control.

However, this type of approach works with sensors that send data from dynamic processes at high speed, where it is not easy to generate actionable insights at the right time. One part of the problem concerns the processing of a large amount of data, while the underlying dynamic phenomena related to the machine, possibly evolve over time giving rise to a concept drift. This is due to factors such as degradation, something common in physical systems. Thus, if the model becomes obsolete, an update is necessary.

To deal with this problem, an approach from unsupervised machine learning is proposed. Specifically, through dynamic clustering algorithms. To do this, we work on a methodology that first allows us to study the behavior of clustering algorithms in industrial applications. Then, we select those algorithms that have the capabilities to provide new knowledge related to the productive elements and their degradation patterns. The next step is to adapt the selected algorithm to the dynamic behavior of the machines, working with data streams, much closer to the industrial reality. In this way, starting from clustering algorithms such as: K -means, agglomerative hierarchical, spectral, affinity propagation and Gaussian mixture models, the last one is selected as the most suitable for this type of application.

A new unsupervised learning algorithm called Gaussian-based dynamic probabilistic clustering (GDPC) is proposed. GDPC integrates and adapts three known algorithms for use in dynamic scenarios: the expectation-maximization algorithm (EM) responsible for parameter estimation of the mixture model and the Page-Hinkley test together with the Chernoff bounds, to detect concept drift.

Unlike other unsupervised methods, the model induced by GDPC provides the membership probabilities of each instance to each cluster or component. This allows us to determine, through an analysis with the Brier score, the membership robustness and its evolution each

time a concept drift is detected. In addition, the algorithm works with few data needs and significantly less computing power, which allow the algorithm to decide when to change the model. However, this algorithm requires a thorough knowledge of the analyzed domain to correctly select parameters such as the number of components. Also, it may be unstable due to another common phenomenon found in industrial data related to non-stationary phases.

Therefore, a series of improvements are proposed to increase the degree of robustness of the algorithm. This new version, called GDPC+, introduces the following improvements: (a) the automatic selection of the number of components of the mixture based on the Bayesian information criterion; and (b) the stabilization due to the transient effects during the concept drift thanks to the integration of the Cauchy–Schwarz divergence with the augmented Dickey–Fuller test. Therefore, GDPC+ can outperform the GDPC in highly dynamic scenarios in terms of the number of false positives.

The development of these algorithms has been supported with tests on synthetic data streams and also data originated on testbeds and a machine-tool during real production, in this case, automotive crankshafts. These results have been validated in terms of different figures of merit like accuracy, recall, specificity and F-score. Additionally, based on the results of clustering of real machine data, sets of rules induced by a supervised algorithm were developed in order to provide insights about the underlying process and its related algorithm.

Índice general

1. La Cuarta Revolución Industrial	1
1.1. La 4IR en el mundo	5
1.1.1. <i>Industrie 4.0</i>	5
1.1.2. <i>Industrial Internet of Things</i>	7
1.1.3. Otras estrategias internacionales	8
1.2. <i>Smartización</i> industrial	9
1.2.1. Nivel de componente	9
1.2.2. Nivel de máquina	10
1.2.3. Nivel de producción	11
1.2.4. Nivel de distribución	12
1.3. Desafíos del aprendizaje automático	12
1.3.1. Impacto en los negocios	13
1.3.2. Impacto en la tecnología	15
1.3.3. Impacto en las personas	16
1.4. Motivación de este trabajo de investigación	16
1.5. Hipótesis y objetivos	17
1.6. Organización del documento	18
I Antecedentes	19
2. Notación y terminología	21
2.1. Aprendizaje automático	21
2.2. Industria	21
3. Aprendizaje automático	23
3.1. Imputación de datos	25
3.2. Selección de variables	26
3.3. Reducción dimensional	27
3.4. Rendimiento de los algoritmos supervisados	28
3.4.1. Matriz de confusión y valores de mérito	29
3.4.2. <i>Brier score</i>	29

3.5. Criterio de información Bayesiano	30
4. Clustering	31
4.1. Clustering jerárquico	31
4.2. K-medias	33
4.3. Clustering espectral	33
4.4. Clustering por propagación de afinidad	34
4.5. Clustering basado en Mixturas Gaussianas	35
4.5.1. Paso-E	36
4.5.2. Paso-M	37
5. Clustering de data streams	39
5.1. Data streams	39
5.2. Clustering aplicado a data streams	40
II Clustering aplicado a procesos industriales	43
6. Tratamiento térmico por láser. Adquisición y tratamiento de datos industriales	45
6.1. Proceso térmico del láser	45
6.2. Metodología para el manejo de los datos	47
6.2.1. Adquisición de datos del tratamiento térmico con láser	48
6.2.2. Configuración experimental	49
6.3. Resultados	50
6.4. Discusión	51
6.4.1. Análisis de los resultados	51
6.4.2. Aplicabilidad de las técnicas de clustering	52
6.5. Conclusiones	53
7. Patrones de funcionamiento de motores industriales	55
7.1. Huella digital de motores	55
7.1.1. Mejora de la fiabilidad mediante huellas digitales	55
7.1.2. Clustering aplicado a elementos del banco de pruebas	58
7.1.3. Descripción del conjunto de datos extraído del banco de pruebas	62
7.2. Algoritmos de clustering para el desarrollo de huellas digitales	64
7.2.1. Detalles de la implementación	64
7.3. Resultados y discusión	65
7.4. Conclusiones	71
8. Aplicación de clustering para detectar condiciones de trabajo en el proceso de mecanizado de alto rendimiento	73
8.1. Clustering en mecanizado	73

8.2. Metodología	75
8.2.1. Adquisición y pre-procesamiento de datos del electromandrino	76
8.2.2. Desarrollo de patrones de comportamiento basados en el aprendizaje automático	77
8.3. Resultados de experimentación	78
8.3.1. <i>K</i> -medias	78
8.3.2. Jerárquico aglomerativo	79
8.3.3. Modelo de mixturas de Gaussianas	81
8.4. Discusión	83
8.4.1. Algoritmos de <i>clustering</i>	83
8.4.2. Comportamiento interno del electromandrino	85
8.5. Conclusiones	86
III Contribuciones al <i>clustering</i> de <i>data streams</i>	89
9. <i>Clustering</i> probabilístico de <i>data streams</i> con modelos dinámicos de mixturas de Gaussianas	91
9.1. Introducción	91
9.2. Metodología	92
9.2.1. Entrenamiento <i>offline</i> del modelo GMM	93
9.2.2. Test de ajuste	94
9.2.3. Detección de valores atípicos	94
9.2.4. Detección del <i>concept drift</i>	95
9.2.5. Asignación de cada instancia a cada clúster	96
9.2.6. Resumen del algoritmo GDPC	96
9.3. Evaluación del rendimiento del algoritmo GDPC	98
9.3.1. Datos experimentales	98
9.3.2. Diseño de experimentos	99
9.3.3. Resultados experimentales	101
9.4. Selección de parámetros	103
9.4.1. Clasificación de parámetros por relevancia	104
9.5. Medición del ajuste del modelo	106
9.5.1. Aplicación del <i>Brier score</i>	107
9.5.2. Resultados de experimentación y análisis	107
9.6. Comparación del GDPC	108
9.7. Aplicación GDPC para el banco de pruebas	109
9.7.1. Adquisición de datos	109
9.7.2. Resultados y discusión	110
9.8. Conclusiones	112

10. <i>Clustering</i> dinámico basado en modelos de mixturas Gaussianas para entornos desconocidos	115
10.1. Introducción	115
10.2. Metodología del algoritmo GDPC+	116
10.2.1. Estimación dinámica del número de componentes	118
10.2.2. Estabilización transitoria del <i>concept drift</i>	118
10.3. Rendimiento del GDPC	120
10.3.1. Evaluación del método de selección de componentes dinámico	121
10.3.2. Evaluación de la estabilización transitoria del <i>concept drift</i>	122
10.4. Aplicación real del GDPC+	125
10.4.1. Descripción del conjunto de datos	125
10.4.2. Detección del <i>concept drift</i> por el GDPC+	126
10.4.3. GDPC+: resultados de <i>clustering</i> sobre el proceso de mecanizado	128
10.4.4. Comparación entre GDPC y GDPC+	132
10.5. Conclusiones	133
IV Conclusiones	135
11. Conclusiones y trabajo futuro	137
11.1. <i>Clustering</i>	138
11.1.1. Aportaciones al <i>clustering</i> : GDPC+	139
11.2. Publicaciones y otros resultados	140
11.3. Trabajo futuro	141
Bibliografía	142

La Cuarta Revolución Industrial

En la actualidad las economías mundiales están experimentando una evolución tecnológica con algunas connotaciones positivas y negativas. La historia ha demostrado que los cambios tecnológicos enriquecen a la sociedad en términos de educación, cohesión y empleo. Sin embargo, algunos de estos movimientos han necesitado tiempo para construir la infraestructura requerida y dar el salto apropiado en el desarrollo industrial.

Los paradigmas tecnológicos mostrados en la Figura 1.1 se denominan revoluciones industriales porque están estrechamente relacionados con las mejoras a nivel de productividad. Estos han provocado cambios disruptivos en los procesos de fabricación desde el siglo XVIII gracias a las mejoras en campos específicos de las tecnologías de fabricación. La Primera Revolución Industrial utilizó el agua y el vapor para automatizar la producción. Durante la Segunda Revolución Industrial, el agua y la energía del vapor fueron reemplazados por la electricidad, aportando un impulso importante a la productividad. En la Tercera Revolución Industrial, los sistemas electrónicos y las tecnologías de la información (IT) se utilizaron para mejorar la automatización de las fábricas¹.

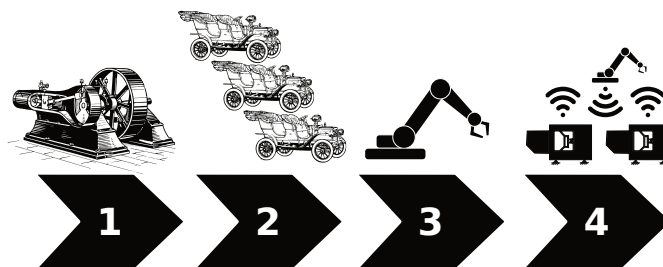


Figura 1.1: Diferentes paradigmas tecnológicos.

Así, el nuevo paradigma tecnológico se denomina Cuarta Revolución Industrial (4IR), y está definido como una mezcla entre los mundos digital y físico. Este paradigma aprovecha las tecnologías digitales emergentes para adquirir y analizar datos a nivel de máquinas, líneas

¹<https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/>

y plantas productivas, combinando las IT de la Tercera Revolución Industrial, es decir, la fabricación asistida por ordenador (Bennett, 1985), el aprendizaje automático (Samuel, 1959), Internet (Kleinrock, 1961), entre otras, con las tecnologías operativas (OT) como el mecanizado, ensamblado, etc. para crear las tecnologías disruptivas, columna vertebral de la 4IR. Un informe técnico publicado por PricewaterhouseCoopers (2017) enumera estas tecnologías disruptivas como:

1. **Materiales avanzados** con funcionalidad mejorada a nivel de propiedades mecánicas y químicas, por ejemplo, los nano-materiales.
2. **Tecnología cloud** capaz de ofrecer recursos computacionales a través de Internet sin necesidad de máquinas locales.
3. **Vehículos autónomos** que pueden navegar con poca o ninguna intervención humana, por ejemplo, los drones.
4. **Biología sintética** que usa principios de ingeniería para desarrollar sistemas biológicos, también llamada biotecnología.
5. **Realidad virtual (VR) o aumentada (AR)** generada por un ordenador con la capacidad de simular una superposición virtual sobre el mundo físico.
6. **Inteligencia artificial** que usa algoritmos para realizar tareas específicas emulando inteligencia humana, por ejemplo, el aprendizaje automático.
7. **Robótica** que utiliza robots para automatizar o asistir en las actividades humanas según un conjunto de instrucciones o de manera autónoma.
8. **Blockchain** que usa algoritmos de software y computación distribuida para registrar y confirmar transacciones en un libro de contabilidad electrónico.
9. **Fabricación aditiva** que construye objetos tridimensionales funcionales o no funcionales mediante la impresión aditiva de materiales.
10. **Internet de las cosas (IoT)** que conecta diferentes objetos embebidos con capacidades de adquisición, pre-procesamiento, procesamiento y comunicación a través de Internet, habilitando aplicaciones inteligentes.

Estas tecnologías se pueden ver a lo largo de todo el paradigma 4IR: mundos biológicos, digitales y físicos. Sin embargo, esta tesis se centra exclusivamente en la aplicación del IoT a la industria manufacturera, en la que los mundos digital y físico se expresan mediante las IT y los sistemas de fabricación avanzados disponibles en diferentes sectores industriales mediante las OT. Todo el conocimiento extraído del análisis de datos gracias a esta integración del mundo digital y físico, tiene el potencial de aumentar la productividad, eficiencia y flexibilidad de los sistemas de producción, mejorando así la competitividad del entramado industrial.

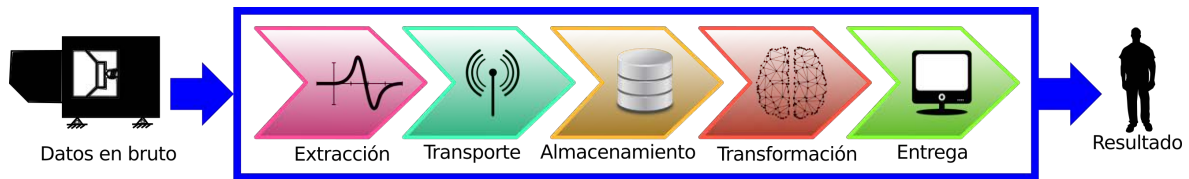


Figura 1.2: Generación de valor añadido desde los datos en bruto hasta la toma de decisiones en la 4IR.

Tanto en el mundo digital como en el físico, los datos podrían tratarse como la materia prima que se convertirá en información de utilidad que permitirá a la 4IR generar el valor añadido esperado. Como materia prima, los datos deben extraerse, transportarse, almacenarse, transformarse y entregarse al usuario final como información con un valor añadido definido como resultado o *actionable insight* (Figura 1.2). A continuación se describe cada etapa dentro de este ciclo de vida de los datos:

- **Extracción:** los dispositivos conectados alrededor del mundo generan 2,5 quintillones de bytes de datos diariamente². En el sector industrial, los datos son generados por los diferentes elementos de las máquinas: sistema de control, sensores y actuadores. Por lo tanto, el único requisito necesario para extraer datos sería una interfaz de conectividad con distintos dispositivos. Sin embargo, es necesario garantizar la calidad de los datos al adquirirlos de diferentes dominios, los cuales pueden tener tasas de muestreo distintas no necesariamente sincronizadas. Por esta razón, los sistemas de adquisición de datos necesitan características especiales, como tecnologías de comunicación avanzadas, estrategias de filtrado y fusión de sensores para poder capturar y enviar datos de manera determinista de forma eficiente.

En consecuencia, las tecnologías habilitadoras del 4IR, como el IoT, proporcionan capacidades de pre-procesamiento y conectividad de alto nivel que aumentan la eficiencia de este tipo de sistemas. Por ejemplo, las redes de sensores inalámbricos (WSN) tienen capacidades de procesamiento, detección y comunicación entre uno y otro, permitiendo compartir datos entre nodos. En este caso, los datos pueden adquirirse indirectamente utilizando la percepción sensorial (Akyildiz et al., 2002) que permite el filtrado de ruido. Además, Li et al. (2013) explica cómo se puede utilizar la auto-conciencia de las redes WSN para garantizar el determinismo durante la extracción.

- **Transporte:** los datos extraídos deben trasladarse del sistema de adquisición a la siguiente etapa de la manera más eficiente posible. Los protocolos de comunicación desempeñan un papel importante al permitir una comunicación rápida, flexible, segura y que facilite su trazabilidad. La 4IR está impulsando nuevos protocolos, como OPC-

²<https://www.ibm.com/blogs/insights-on-business/consumer-products/2-5-quintillion-bytes-of-data-created-every-day-how-does-cpg-retail-manage-it/>

UA³, DDS-Secure⁴ o MQTT⁵, que pueden cumplir estos requisitos, permitiendo la interoperabilidad entre diferentes dispositivos, la comunicación en tiempo real y un flujo continuo de datos.

- **Almacenamiento:** si se generan 2,5 quintillones de bytes de datos diariamente, se necesita un sistema de gestión y almacenamiento adecuado que permita consultas eficientes para respaldar la transformación de los datos en información útil. Debido a que esta producción de datos es cada vez mayor, es necesario un sistema de almacenamiento de alto rendimiento, escalable y utilizable. Por lo tanto, la 4IR ha desarrollado el concepto de *big data* que hace referencia a conjuntos de datos más grandes y complejos. Las tecnologías de almacenamiento de datos utilizadas comúnmente no son adecuadas para *big data*. Por esta razón, se han desarrollado soluciones basadas en Hadoop⁶, orientadas al almacenamiento distribuido y altamente escalable a grandes conjuntos de datos, como Cloudera, Hortonworks y MapReduce, (Strohbach et al., 2016).

En este caso, dependiendo de las necesidades del paso de transformación, el almacenamiento podría ser a largo plazo o instantáneo. El almacenamiento a largo plazo es necesario cuando los análisis se realizan sobre largos periodos de tiempo y sus resultados no son requeridos en tiempo real. Por ejemplo, Kezunovic et al. (2017) describen el uso de *big data* para predecir el impacto del clima en los sistemas energéticos, en los que se necesitan grandes conjuntos de datos para poder correlacionar correctamente los efectos y asegurar las capacidades de predicción del algoritmo.

Por otra parte, el almacenamiento instantáneo es necesario cuando la información requerida tiene requerimientos temporales como el trabajo en tiempo real. En este caso, uno de los tipos de almacenamiento temporal más usado son las bases de datos en memoria, las cuales utilizan una ventana o *buffer* temporal con un tamaño de almacenamiento relativamente pequeño. Este tipo de bases de datos son eliminadas una vez que se consumen.

- **Transformación:** este paso está relacionado con la transformación de datos en información útil. El aprendizaje automático es una de las técnicas clave capaz de generar modelos basados en datos que permitan predecir, clasificar o extraer nuevo conocimiento como soporte a la toma de decisiones. Existen otras técnicas, como la analítica visual, que permiten transformar los datos en información. Sin embargo, esta tesis se centra en su totalidad en técnicas de aprendizaje automático aplicadas en el sector industrial.
- **Entrega:** cuando la información es procesada y entregada al usuario final es importante tener en cuenta que éste puede ser el operador de la máquina, el gerente de la planta o el ingeniero de mantenimiento, el cual debe utilizar una interfaz hombre-máquina (HMI), o directamente a la máquina como retroalimentación dentro del lazo cerrado

³<https://opcfoundation.org/about/opc-technologies/opc-ua/>

⁴<https://www.rti.com/products/secure>

⁵<http://mqtt.org/>

⁶<http://hadoop.apache.org/>

de control. Esta fase debe tener esto en cuenta para tratar de entregar de una manera óptima el resultado, permitiendo así soportar la toma de decisiones racionales derivadas del mismo.

1.1. La Cuarta Revolución Industrial en el mundo

El ciclo de vida de los datos descrito anteriormente representa la columna vertebral de la 4IR, el cual, se ha adoptado en todo el mundo con diferentes matices que se describen brevemente a continuación.

1.1.1. *Industrie 4.0*

El concepto *Industrie 4.0* fue definido por [Kagermann et al. \(2013\)](#) como una iniciativa para asegurar el futuro de la industria manufacturera alemana. Es una definición amplia que cuenta con ocho áreas clave:

- **Arquitectura de referencia:** ésta es el área más activa ya que la plataforma *Industrie 4.0* entiende que la mejor manera posible de permitir la colaboración entre empresas es compartiendo datos e información. Sin embargo, este intercambio requiere estándares comunes y una arquitectura de referencia que permita una integración entre los diferentes socios con un despliegue sencillo.
- **Administración de sistemas complejos:** éste área se enfoca en el desarrollo de tecnología diseñada para gestionar productos y sistemas de fabricación cada vez más complejos. Los sistemas industriales de próxima generación serán más difíciles de administrar debido a características novedosas como su interconectividad y comportamiento adaptativo.
- **Una infraestructura de banda ancha integral para la industria:** el desarrollo de nuevas generaciones de redes de comunicación es importante para poder compartir de manera fiable datos de alta calidad entre diferentes compañías. El intercambio de datos e información tiene problemas de escalabilidad directamente asociados con el tamaño de la planta productiva que deben resolverse.
- **Seguridad y protección:** ésta es un área importante de actividad y desarrollo porque el intercambio de datos e información debe ser lo suficientemente fiable para garantizar que los productos y los procesos productivos no sean un peligro para las personas ni para el medio ambiente. Así pues, los datos y la información deben protegerse contra el uso indebido y el uso no autorizado. De esta manera, son necesarias nuevas tecnologías que sean capaces de gestionar grandes cantidades de datos e información críticos.
- **Organización y diseño del trabajo:** dado que el objetivo final de la 4IR es promover el desarrollo de fábricas inteligentes e interconectadas que compartan datos e información para mejorar la productividad de los sistemas de fabricación, es necesario

que los puestos de trabajo del futuro puedan adaptarse a los requisitos de las tareas. Por ejemplo, las tareas repetitivas o de baja destreza serán reemplazadas por actividades de mayor valor añadido que mejoren el desarrollo personal de los empleados.

- **Formación y desarrollo profesional continuado:** debido a los cambios descritos anteriormente relativos a los requisitos en habilidades de los empleados, las estrategias de formación deben adaptarse para proporcionar las herramientas necesarias para que los empleados puedan realizar sus actividades en el nuevo entorno de trabajo creado por la revolución industrial.
- **Marco regulatorio:** las colaboraciones promovidas por el *Industrie 4.0* se basan en el intercambio de datos e información que actualmente debe realizarse sobre una legislación que aún no se ha desarrollado. Es por ello que se debe diseñar un marco claro que permita definir los límites de propiedad de los datos y de la información, en donde, según el escenario de implementación, se debe hacer una distinción clara entre datos personales, corporativos, de productos y de procesos.
- **Eficiencia de recursos:** el sector industrial es el mayor consumidor de energía del mundo ya que necesita una gran cantidad de energía para transformar las materias primas en productos. Además, la fábrica conectada y toda la gestión de datos resultante, requerirán el uso de equipos de tecnología avanzada con mayores requisitos de consumo energético. Siendo así, resulta necesario estudiar el compromiso entre los recursos adicionales requeridos y los posibles ahorros de energía para mejorar en la eficiencia.

Para desarrollar estas áreas clave, se han introducido conceptos como los sistemas ciberfísicos (CPSs) (Baheti y Gill, 2011) para respaldar la introducción de la IoT en el entorno de fabricación. Por lo tanto, Kagermann et al. (2013) definen el CPS como sistemas inteligentes que permiten almacenar, intercambiar información, desencadenar acciones y controlar de una manera desatendida los medios productivos. Sin embargo, lo más importante es que constituyen el nexo entre los mundos digital y físico.

En consecuencia, al sustituir la Figura 1.2 por las características aportadas por la definición de Kagermann et al. (2013), se obtiene que un CPS debe ser capaz de adquirir, transportar, almacenar, transformar los datos y entregar la información. Para poder realizar esta tarea, un CPS deberá contar con inteligencia artificial que le permita operar sin supervisión gracias a las propiedades de auto-aprendizaje. Es en esta fase cuando el aprendizaje automático resulta ser la tecnología de inteligencia artificial específica que aporta estas capacidades, especialmente en la etapa de transformación.

Es importante resaltar que la definición de *Industrie 4.0* no incluye ninguna referencia explícita a la inteligencia artificial. Sin embargo, uno de los autores del término *Industrie 4.0*, el Prof. Dr. Wolfgang Wahlster, es el Director Ejecutivo del Centro Alemán de Investigación de Inteligencia Artificial (DFKI GmbH). El Prof. Dr. Wahlster usa su institución para posicionar a la inteligencia artificial como el principal impulsor de las fábricas inteligentes respaldadas por los CPSs.

Aunque el *Industrie 4.0* es una iniciativa alemana, el concepto general ha sido adoptado rápidamente por casi todos los países europeos. La adopción se ha llevado a cabo a muchos niveles; desde el gobierno local, en forma de políticas, hasta las empresas.

1.1.2. *Industrial Internet of Things*

Los primeros pasos hacia el *Industrial Internet of Things* (IIoT) se describieron en el marco del Consejo de Asesores sobre Ciencia y Tecnología del Presidente de los Estados Unidos en julio del 2012⁷. En este escenario, algunas de las tecnologías transversales seleccionadas para la fabricación avanzada fueron: sensórica avanzada, tecnologías de la información, fabricación y visualización digital.

En marzo de 2012, el Comité Directivo de Estados Unidos para la Innovación Fundamental en CPSs, dirigido por Universidad de Vanderbilt y Boeing, presentó un informe sobre las oportunidades estratégicas para el CPS en el siglo XXI (Sztipanovits et al., 2012). En este informe se definió el CPS como un sistema físico y cibernético estrechamente acoplado que exhibe un nivel de integración inteligente. Estos sistemas tienen procesos computacionales que interactúan con componentes físicos. Por lo tanto, se esperaba un cambio importante, ya que se entendía que las aplicaciones futuras de los CPSs serían más disruptivas de lo que era la IT durante la Tercera Revolución Industrial.

A finales de 2012, la rama digital de una compañía de los EE.UU., General Electric (GE), acuñó el término *Industrial Internet*, reuniendo máquinas inteligentes, análisis avanzados y personas en el trabajo. GE describió esta integración como una red de dispositivos conectados que pueden adquirir, transportar, almacenar, transformar y proporcionar información valiosa y útil que puede apoyar decisiones empresariales más rápidas dentro del tejido empresarial, aumentando así su competitividad⁸.

De esta manera, el IIoT está orientado principalmente a la aplicación del IoT, las comunicaciones máquina a máquina (M2M) y el análisis de grandes datos industriales a la industria, con un claro enfoque en el uso de datos para generar valor añadido. En vista de la necesidad de compartir datos e información, el enfoque de IIoT toma las principales tecnologías de IoT, como la detección inteligente, el tiempo real, las comunicaciones determinísticas e inalámbricas, la fusión de sensores para el pre-procesamiento de datos, la inteligencia artificial para el procesamiento de datos y la entrega de información a la industria. Además, el enfoque IIoT define diferentes capas de despliegue de tecnología:

- **Edge**: los elementos están cerca de los medios productivos, lo que es útil para el análisis en tiempo real y las tecnologías de control.
- **Cloud**: los datos se envían a los servidores externos a través de Internet, útil para análisis complejos a largo plazo y almacenamiento de datos.

⁷<https://energy.gov/eere/downloads/report-president-capturing-domestic-competitive-advantage-advanced-manufacturing>

⁸<https://www.ge.com/digital/blog/everything-you-need-know-about-industrial-internet-things>

En este contexto en marzo de 2014 GE, IBM y SAP fundaron el Industrial Internet Consortium (IIC)⁹ con el objetivo de reunir a empresas y tecnologías habilitadoras necesarias para acelerar el desarrollo, la adopción y el intercambio generalizado de datos e información, el análisis inteligente y la mejora de las condiciones de las personas en el trabajo. Aunque comenzó como una iniciativa principalmente de los EE.UU., el IIC ahora se ha globalizado con más de 200 compañías miembros de la mayoría de los países industrializados.

1.1.3. Otras estrategias internacionales

Como se explica en la Sección 1.1.1 y en la Sección 1.1.2, el concepto de 4IR se ha adoptado en todo el mundo inspirándose en las iniciativas originarias de Alemania y EE. UU. Sin embargo, hay variaciones específicas según el país. Algunos de estos enfoques se describen brevemente a continuación.

En Francia, el 4IR fue adoptado en abril de 2015 como *Industrie du Futur*, orientado hacia la transformación digital de la industria francesa. Es principalmente una implantación de las iniciativas de la Comisión Europea, como la Fábrica del Futuro. *Industrie du Futur* ha tomado prestadas cinco nociones principales de las iniciativas de la Unión Europea: (1) desarrollo de la cadena de suministro de tecnología para las fábricas del futuro en áreas donde Francia puede convertirse en líder en los próximos tres a cinco años gracias al apoyo de grandes proyectos industriales. El suministro de tecnologías se basará en la fabricación aditiva, IoT, realidad aumentada, etc.; (2) apoyo financiero para empresas; (3) formación para la próxima generación de empleados en el conocimiento y las habilidades necesarias para aplicar nuevas tecnologías en las fábricas del futuro; (4) apoyo a la cooperación europea e internacional, fomentando estrategias de innovación junto con otros países europeos, especialmente Alemania, y otras alianzas internacionales; (5) promoción de actividades orientadas a mostrar los desarrollos franceses relacionados con la 4IR y los conocimientos tecnológicos.

En España, la adopción de la 4IR está impulsada por *Industria Conectada* apoyada desde 2017 por el Ministerio de Industria, Comercio y Turismo y la *Estrategia Española de la Inteligencia Artificial* apoyada desde 2019 por el Ministerio de Ciencia, Innovación y Universidades. En este caso, las iniciativas están diseñadas para proporcionar apoyo financiero y asistencia para promover la transformación digital del sector industrial español. Al igual que en *Industrie du Futur*, el enfoque adoptado por *Industria Conectada* está alineado con la *Industrie 4.0* alemana. Sin embargo, requiere un enfoque de solución empresarial específica que se centre en *big data* y analítica, ciberseguridad, computación en la nube, conectividad y movilidad, fabricación aditiva, robótica y sensores y sistemas integrados como las principales áreas de desarrollo.

En Asia hay varios enfoques: *Made in China 2025*, *Made in India* y *ASEAN 4.0* para la Asociación de Naciones del Sudeste Asiático (ASEAN), incluyendo dentro de sus miembros líderes de desarrollo tecnológico como Singapur y Malasia. Todos estos enfoques están alineados con *Industrie 4.0* y están diseñados para impulsar sus respectivas industrias con el

⁹<http://www.iiconsortium.org/>

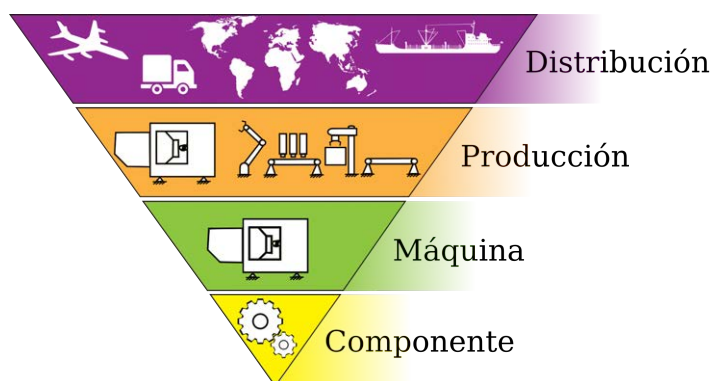


Figura 1.3: Diferentes niveles de *smartización* industrial.

fin de aumentar la competitividad. Por otra parte, Japón ha adoptado un enfoque diferente llamado *Society 5.0*. Esta iniciativa está orientada hacia la transformación de la sociedad en una sociedad súper inteligente. Esta política espera que el CPS, visto como elemento clave, tenga la capacidad de combinar el espacio físico y cibernético, produciendo un cambio social importante. Las máquinas y la inteligencia artificial serán los principales actores en esta quinta etapa de la sociedad.

En conclusión, la 4IR es más que un desarrollo tecnológico: es un cambio industrial que involucra componentes económicos, técnicos y sociales orientados a mejorar la competitividad industrial en todos los niveles con un impacto potencial en todo el mundo. Esta revolución, y las diferentes políticas adoptadas, están aprovechando la industria inteligente descrita en la Sección 1.2.

1.2. *Smartización* industrial

La palabra *smartización* se usa frecuentemente para describir un cambio evolutivo hacia un comportamiento inteligente. Las tecnologías relacionadas con este cambio están permitiendo a las industrias ser inteligentes en diferentes niveles. La *smartización* es el hilo conductor de esta tesis, utilizado para describir la aplicación del aprendizaje automático en entornos industriales y proporcionar cualidades de inteligencia a medios productivos. Para esto, se definen cuatro niveles diferentes de abstracción: *componente* (Sección 1.2.1), que hace parte de una *máquina* (Sección 1.2.2) dentro de una línea de *producción* (Sección 1.2.3) para luego repartir los productos mediante sistemas de *distribución* (Sección 1.2.4). La Figura 1.3 muestra un diagrama que ilustra este enfoque.

1.2.1. Nivel de componente

En la parte inferior de la Figura 1.3, se tienen los componentes de la máquina, por ejemplo, rodamientos, válvulas, husillos de bolas, guías y ejes. La *smartización* de componente se refiere a la introducción de auto-consciencia para así proporcionar capacidades de diagnóstico y predicción de fallos. Esto puede ayudar a aumentar la disponibilidad de todo el sistema

o subsistemas, tales como, máquinas, compresores de aire, etc. Los componentes se hacen conscientes al incorporar sensores dentro de su estructura.

El objetivo principal de estos sensores integrados es extraer variables relacionadas con fenómenos que pueden hacer que el componente falle. Por ejemplo, un sensor integrado en un rodamiento podría tener que medir vibración, temperatura, lubricación, humedad, presencia de partículas metálicas, etc. Estos datos se pueden procesar en el sensor o en una capa superior, activando alarmas relacionadas con potenciales fallos o reducción de vida útil (RUL). En este caso, los sensores se conocen como sensores inteligentes.

El Capítulo 7 trata una aproximación a nivel de componente, en este caso, de los rodamientos de los servomotores de los ejes de la máquina-herramienta.

1.2.2. Nivel de máquina

El siguiente nivel de abstracción en la industria es la máquina. En este caso, hay dos fuentes de inteligencia: (1) componentes conscientes de sí mismos, que pueden proporcionar diagnóstico y predicción de fallos, (2) consolidación de datos de diferentes componentes inteligentes y sensores que pueden generar información de contexto, útiles para complementar la información disponible sobre el sistema o sub-sistema.

Lee et al. (2014) explican que el IoT ha habilitado la disponibilidad de datos, en donde una máquina, con la ayuda de un CPS, puede extraer suficiente información para poder realizar una auto-evaluación. Como la disponibilidad es el problema más importante a resolver para una máquina industrial, las capacidades de auto-evaluación pueden proporcionar las condiciones pasadas, actuales y futuras de los sub-sistemas, permitiendo así solucionar este problema mediante el mantenimiento y el control adaptativo.

Así, una máquina auto-suficiente es capaz de evaluar su propio estado de salud y nivel de degradación. Esto es útil para el mantenimiento preventivo y predictivo que permiten reducir el tiempo de inactividad de la máquina, aumentando así su disponibilidad. Una máquina consciente de sí misma puede usar los datos para monitorizar las condiciones de operación actuales y evaluar su mejor estado operativo, ajustando los parámetros del proceso para asegurar la mayor eficiencia posible.

Sin embargo, el concepto de máquina inteligente es más amplio que el uso de datos para la auto-evaluación. Como se describe al comienzo de este capítulo, uno de los conceptos clave es el intercambio de datos e información. A este respecto, la comunicación máquina a máquina (M2M) es un concepto descrito por muchos autores, por ejemplo, Lee et al. (2014), Lin y Chen (2016), Li et al. (2016), Ali et al. (2017) y Tuna et al. (2017), los cuales destacan los datos y el intercambio de información entre las máquinas para realizar una comparativa entre ambas. Esto puede ser útil para detectar una degradación temprana o cualquier otra situación que pueda aumentar la disponibilidad de la máquina. M2M también es útil para crear una red de máquinas inteligentes cooperativas, en donde la coordinación adaptativa aumenta la flexibilidad y la productividad, generando el concepto de sistemas de producción inteligentes.

Los Capítulos 8, 9 y 10 describen cómo el aprendizaje automático puede aprovechar las capacidades de auto-conocimiento en una máquina de mecanizado de componentes de

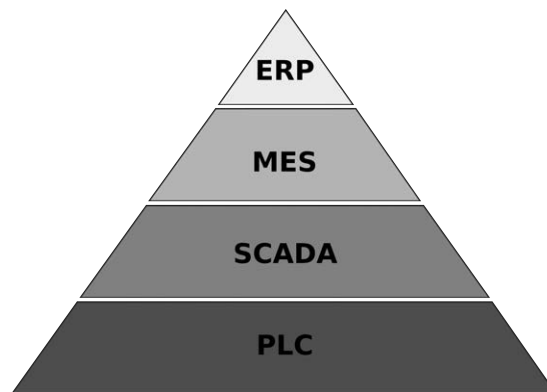


Figura 1.4: Integración dentro de un sistema productivo.

automoción.

1.2.3. Nivel de producción

Como se ha señalado en la Sección 1.2.2, un conjunto de máquinas inteligentes en red, que comparten datos e información, constituye lo que se define como un sistema de producción inteligente. Además, esta interconexión de la máquina proporciona un análisis sobre toda la flota de activos denominada eficiencia global del equipo (OEE) y definida por disponibilidad, productividad, eficiencia energética y calidad de fabricación.

El nivel de producción se define como un sistema de fabricación inteligente. Este sistema inteligente es capaz de integrar máquinas inteligentes e inferir conocimiento de datos provenientes de otros dominios, como el comportamiento de las materias primas, el medio ambiente, la energía, los negocios, el flujo de materiales y otros indicadores clave de rendimiento (KPIs). Esta integración proporciona una vista de alto nivel del entorno de fabricación, en el que los datos podrían extraer información de valor añadido que puede ayudar a aumentar la eficiencia del sistema.

Por lo tanto, como se muestra en la Figura 1.4, una fábrica inteligente puede aprovechar los sistemas de integración bien establecidos como la planificación de recursos empresariales (ERP), que cuenta con datos comerciales sincronizados en tiempo real: compras, ventas, distribución, finanzas, fabricación, entre otros. A partir de estos datos, una fábrica inteligente debe tomar decisiones basadas en negocio para aumentar su competitividad. Además, los sistemas de ejecución de la fabricación (MES) son una fuente de datos útiles para fábricas inteligentes. En este caso, un MES puede proporcionar datos relacionados con el sistema de producción, KPI de seguimiento, materias primas, stock, etc. Los controladores lógicos programables (PLC) y el control de supervisión y adquisición de datos (SCADA) son capas inteligentes sobre las máquinas con capacidad de proporcionar control directo y supervisión de los sistemas de producción.

El Capítulo 6 describe una aplicación orientada a la producción de cigüeñales de automoción.

1.2.4. Nivel de distribución

El siguiente nivel de abstracción es la distribución, en el que los productos se envían a los clientes u otras plantas productivas para terminar su fabricación. Este nivel se define como logística inteligente. En este punto, los datos agregados provenientes de diferentes sistemas de producción se mezclan con los datos de distribución para aumentar la eficiencia del sistema, es decir, entregar el producto en el momento adecuado.

Los sistemas de distribución son combinaciones complejas de infraestructura y de recursos con disponibilidad limitada, y con diferentes destinos de productos y tiempos de entrega requeridos. Como resultado, los procesos deben ser altamente eficientes para evitar cuellos de botella y reducir los tiempos de comercialización del producto (tiempo que se tarda en desarrollar un producto hasta su entrega al cliente) para evitar un impacto negativo en la competitividad de la empresa.

Por lo tanto, los sistemas de distribución inteligentes son administradores de recursos complejos que pueden realizar tres actividades diferentes: planificación, implementación y control automatizados. Sobre la base de los datos de producción, el sistema inteligente define un plan de entrega del producto, incluido su destino, la infraestructura y los recursos necesarios (por ejemplo, aeropuertos, aviones, etc.) y la planificación de contingencias. La planificación se realiza mediante la búsqueda de la ruta óptima y teniendo en cuenta otros factores tales como los diferentes productos que utilizan los mismos recursos para maximizar su uso. Además, el sistema inteligente implementa y/o ejecuta el plan, midiendo sus estados pasados, actuales y futuros con el fin de detectar posibles desviaciones y de generar información útil basada en el plan de contingencia. Si se detectan tales desviaciones, el sistema inteligente puede controlar la situación y tomar las medidas necesarias para garantizar la calidad del mismo.

1.3. Desafíos del aprendizaje automático en la 4IR

Según algunos estudios, se espera que el IIoT aumente la eficiencia industrial en un 3,3 % por año con ahorros de alrededor del 2,6 % con respecto a la reducción de costos¹⁰. Estas cifras serán el resultado de un aumento general de la eficiencia, lo que llevará a una mayor producción con un menor uso de materia prima y energía. Como se describe en la Sección 1.2, los desarrollos se están moviendo hacia la *smartización*, donde la inteligencia artificial tiene una gran relevancia.

Asimismo, la inversión en aplicaciones de IIoT se mide en miles de millones según la región. Por ejemplo, la inversión esperada en Europa es de en torno a 140 mil millones de euros por año. Esto significa que los productos basados en inteligencia artificial orientados a la industria recibirán un fuerte respaldo que aumentará la tasa de adopción. El aprendizaje automático es una tecnología de inteligencia artificial con aplicaciones prometedoras que permiten sistemas de fabricación inteligentes dentro de la 4IR. Tiene un enorme potencial

¹⁰<https://www.pwc.nl/en/assets/documents/pwc-industrie-4-0.pdf>

en todos los niveles: negocios, tecnología y personas. Sin embargo, existen algunos desafíos y oportunidades relacionados con cada nivel.

Para comprender los desafíos y las oportunidades para el aprendizaje automático en la industria, se debe analizar cómo encaja en una arquitectura 4IR. Para hacer esto, se usa la arquitectura de referencia para IIoT (Lin et al., 2017). Esta arquitectura define tres niveles diferentes de la siguiente manera:

- **Nivel *Edge***: recopila los datos provenientes de diferentes niveles industriales: componente, máquina, línea de producción o logística (Sección 1.2).
- **Nivel *Platform***: procesa los datos desde el nivel *Edge* y proporciona una primera capa de servicios y comentarios en la que el tiempo es una variable crítica por razones de seguridad e integridad.
- **Nivel *Enterprise***: recopila información del nivel de *Platform* y despliega un segundo nivel de servicios que brinda soporte para la toma de decisiones de alto nivel.

La Figura 1.5 ilustra una implantación de esta arquitectura. Este ejemplo aborda el mantenimiento predictivo de activos en las que se recopilan, transmiten y sincronizan diferentes datos de componentes en el nivel *Edge*. Posteriormente, todos los datos se envían al nivel de *Platform* donde una capa de aprendizaje automático extrae información útil crítica que se puede usar para detener la máquina en caso de una emergencia o para respaldar la toma de decisiones por parte del operario de la máquina en caso de una alta probabilidad de fallo. Parte de esta información se transmite al siguiente nivel, donde otra capa de aprendizaje automático extrae información operativa y/o orientada al negocio. En esta capa, las decisiones de negocios se toman en base a los conocimientos suministrados, como la planificación de producción o la disponibilidad general de la fábrica.

Una arquitectura inteligente de la industria podría asignarse a diferentes niveles de impacto analizando las principales implicaciones del aprendizaje automático. Por lo tanto, el impacto más importante del aprendizaje automático utilizado en el nivel empresarial está relacionado con las personas y los negocios. El aprendizaje automático aplicado en el nivel de *Platform* tendrá menos impacto en los negocios y otorgará más importancia a la tecnología. En el nivel *Edge*, el aprendizaje automático tendrá un impacto directo como habilitador principal de cada elemento inteligente dentro de este nivel. Las siguientes secciones ofrecen una idea general del impacto esperado en cada nivel con sus desafíos y oportunidades relacionados.

1.3.1. Impacto en los negocios

Como se ha descrito anteriormente, el aprendizaje automático será un habilitador clave para las industrias inteligentes, incrementando los niveles de OEE y teniendo un impacto positivo en la competitividad de las empresas. Un aumento en la competitividad significa que los bienes producidos por industrias inteligentes tienen un tiempo de comercialización

¹¹<http://www.iiconsortium.org/smart-factory-machine-learning.htm>



Figura 1.5: El papel de aprendizaje automático en el mantenimiento predictivo de las fabricas inteligentes¹¹.

relativamente más corto que los productos de la competencia, es decir, el suministro del producto en el momento y al precio adecuado.

Conceptos como la personalización masiva o la *servitización* han surgido para lograr este nivel de mejora. La personalización masiva está relacionada con la capacidad de los sistemas de producción para proporcionar grandes volúmenes de productos y servicios personalizados gracias a una gran flexibilidad. Para lograrla, el sistema de producción debe conocer las condiciones pasadas y presentes para monitorizar la disponibilidad real, así como las condiciones futuras para predecir los cambios requeridos en el sistema de producción, mayoritariamente con la personalización de un nuevo producto.

De esta manera, la *smartización* de activos productivos tiene un impacto disruptivo en los negocios, llamada *servitización*. Como lo describen [Kamp et al. \(2017\)](#), la *smartización* aprovechará nuevos modelos de negocio como el mantenimiento, el control de calidad, la eficiencia de la planta, etc., que aprovechan los resultados de análisis predictivos realizados con aprendizaje automático. Por otro lado, una mayor disponibilidad permitirá vender tiempo de operación en lugar de máquinas. Estos modelos de negocio se aplican a sectores como la industria aeronáutica, donde las turbinas de los aviones se venden por su tiempo en vuelo.

Existen oportunidades para que el aprendizaje automático influya y aumente la competitividad de las empresas pero hay algunos desafíos que se deben superar. El desafío más importante es comprender que el aprendizaje automático no es una tecnología de bajo coste: se requiere una estrategia de implantación elaborada para valorar cómo obtener el retorno de la inversión más rápido y lograr un mayor impacto en la empresa. De esta manera, se están realizando esfuerzos desde muchos frentes para reducir la necesidad de recursos requeridos por el aprendizaje automático, como el almacenamiento de datos y el tiempo de entrenamiento del algoritmo. Algunos ejemplos son el *one-shot learning* capaz de aprender a partir del primer ejemplo ([Fei-Fei et al., 2006](#)), algoritmos de aprendizaje que aprenden de los *data*

streams y no de las bases de datos (Silva et al., 2013) o algoritmos *novelty detection*, capaces de realizar aprendizaje en tiempo real de situaciones desconocidas (Faria et al., 2016). Este tipo de enfoque se presenta en la Parte III del presente documento.

1.3.2. Impacto en la tecnología

El principal impacto del aprendizaje automático es en la tecnología, ya que habilita la *smartización* de activos productivos dentro de las industrias inteligentes. De esta manera, un componente, máquina, línea de producción o fábrica es consciente de su condición, y tiene la capacidad de reaccionar al ser entrenado con un algoritmo de aprendizaje automático para este propósito.

Existe una gran cantidad de oportunidades para diferentes aplicaciones en donde se requiere el aprendizaje automático. Sin embargo, como se explica en la Sección 1.3.1, el aprendizaje automático no siempre es aplicable debido a su coste o viabilidad, y menos si para ello existen soluciones tradicionales basadas en ingeniería capaces de resolver el problema. Los algoritmos de aprendizaje automático deben aplicarse para mejorar las tecnologías siempre que cualquier otro enfoque de ingeniería tradicional no pueda proporcionar los resultados requeridos en cuanto a precisión o tiempo de respuesta y/o coste, causados por la complejidad del proceso o las incógnitas específicas del mismo.

Desde el punto de vista del desarrollo tecnológico se espera que el aprendizaje automático pueda mejorar el comportamiento de los activos productivos a través de una mayor disponibilidad, eficiencia y un menor consumo energético. Esto provocará un aumento general de la productividad que justifica la entrada de tecnologías de aprendizaje automático en fábricas inteligentes.

Al mismo tiempo, el aprendizaje automático está mejorando la transparencia dentro de la industria, donde el uso de algoritmos exploratorios y a la vez transparentes e interpretables se está aprovechando para una mejor comprensión de los productos y procesos. Esta retroalimentación resultará en mejores decisiones durante el diseño del producto o proceso e incluso en nuevos productos y procesos. En consecuencia en la presente tesis se trabaja fundamentalmente sobre algoritmos exploratorios no supervisados llamados *clustering* (Capítulo 4).

Desde el punto de vista del aprendizaje automático, el principal desafío relacionado con la tecnología es diseñar algoritmos rápidos, precisos, eficientes y robustos para satisfacer las necesidades de la industria. Por lo tanto, existe la necesidad de un enfoque que permita extrapolar los nuevos desarrollos del laboratorio hacia aplicaciones industriales más rápidamente. De esta manera, la 4IR está impulsando a las empresas industriales a proporcionar aplicaciones reales para desplegar algoritmos en la etapa de desarrollo, con el fin de reducir los tiempos de comercialización. Es por ello, que esta tesis busca desarrollar algoritmos, como los tratados en la Parte III, que puedan ser usados en la industria desde los puntos de vista de desarrollo e implementación en entornos industriales.

1.3.3. Impacto en las personas

Como se describe en la Sección 1.1.3, *Society 5.0* de Japón, es el enfoque que mejor ilustra el impacto del aprendizaje automático en las personas. Por lo tanto, hay muchas oportunidades para usar estas técnicas y reemplazar a los humanos en tareas tediosas, estresantes y repetitivas dentro de la industria.

Sin embargo, cuando se mencionan los términos inteligencia artificial, fabricación inteligente, sistemas de producción conscientes y máquinas autónomas, la primera preocupación que se plantea es que los sistemas de fabricación destruirán puestos de trabajo produciendo un conflicto *hombre contra máquina*. Este es el primer desafío que debe superar el aprendizaje automático en fábricas inteligentes: demostrar que es una tecnología de soporte y no una amenaza para el empleo.

Aunque las industrias inteligentes generan sentimientos negativos, es de hecho el caso contrario. El término clave es *inteligencia*: éste no debe utilizarse junto a la palabra *máquina*. Los seres humanos tienden a romper las reglas y aplican sus habilidades de improvisación para reaccionar ante las perturbaciones de las que no tienen conocimiento previo. Por esta razón, las personas pueden diseñar máquinas programadas para realizar tareas específicas.

Una máquina programada es buena para seguir las reglas y es capaz de reaccionar a las perturbaciones de acuerdo con su propio entrenamiento. Esto no significa que sea inteligente, ya que sin un entrenamiento específico, no podrá realizar su trabajo. Por ejemplo, si la máquina tiene capacidades de auto-mantenimiento que le permiten realizar una predicción relacionada con la degradación de los rodamientos, estas mismas capacidades serán totalmente inútiles para predecir la degradación de los husillos de bolas de un eje lineal, sin importar qué tan similares sean sus componentes.

En este escenario, las personas son la parte más importante de las industrias inteligentes ya que los empleados diseñan, programan, implementan y monitorizan las directrices que mejor convienen para la competitividad de la empresa. Por lo tanto, el impacto en las personas será un cambio principalmente educativo. Los empleados de la 4IR serán formados para satisfacer las necesidades de la industria inteligente, donde las capacidades intelectuales son más importantes que las capacidades físicas. Como resultado, la 4IR proporcionará trabajos de mejor calidad que involucren tareas de alto nivel más gratificantes, que influyen en el desarrollo profesional y personal de los empleados. Por lo tanto, el bajo valor añadido y las tareas repetitivas que comprometen la ergonomía, se dejarán para las máquinas.

Las anteriores secciones han sido publicadas en el Capítulo 1 del libro ([Larrañaga et al., 2019](#)).

1.4. Motivación de este trabajo de investigación

Tal y como se ha descrito anteriormente, la 4IR es la principal fuerza impulsora de la fusión de diferentes tecnologías disponibles que conducen a un cambio industrial que afectará a la sociedad en diferentes niveles. Existen diferentes enfoques en todo el mundo, todos con un objetivo común: impulsar la competitividad del sector industrial de su país. Aunque

estas políticas persiguen intereses diferentes, los datos se definen en todos los casos como el habilitador de la convergencia necesaria entre IT y OT, siendo el enlace principal para compartir información valiosa entre componentes, máquinas, sistemas de producción e industrias, aumentando así la competitividad.

El aprendizaje automático es la técnica más común para extraer información útil e implementar capacidades inteligentes. En este campo, el aprendizaje automático como rama de la inteligencia artificial, es una de las tecnologías líderes de la 4IR. Por lo tanto, ésta es una clara oportunidad para que esta tecnología con una larga historia (Minsky, 1961; Turing, 1950), tome la iniciativa en el desarrollo para las industrias. También, existe la posibilidad de aprovechar los nuevos desarrollos de la tecnología *Edge* o *Cloud* para implementar algoritmos potentes y extraer información valiosa de los datos.

Las fábricas inteligentes son el objetivo de la 4IR dentro del sector industrial y son el resultado de muchos años de investigación y desarrollo en diferentes campos de aplicación, que hasta ahora están migrando del laboratorio hacia la fábrica. Es por ello que se requiere más investigación relacionada con la integración de diferentes tecnologías en todos los niveles, permitiendo esta migración.

Se están dando avances importantes y por ello este trabajo de tesis busca dar un paso más para desarrollar técnicas de aprendizaje automático que puedan ser desplegadas en la industria, permitiendo así potenciar herramientas que puedan resolver los problemas que no pueden ser resueltos por medios convencionales de ingeniería. De esta manera, existe una motivación clara para desarrollar y aplicar algoritmos de aprendizaje automático que tengan en cuenta las necesidades industriales de los medios productivos, siendo éstos altamente dinámicos y por otro lado, con un número mínimo de fallos. Se debe resaltar su importancia, ya que desde el punto de vista industrial es necesario desarrollar máquinas que fallen menos y sean altamente flexibles soportadas por algoritmos de aprendizaje automático. Esto puede ser complejo a la hora de desarrollar algoritmos, ya que éstos pueden no adaptarse a condiciones con datos altamente desbalanceados y cambiantes en el tiempo.

1.5. Hipótesis y objetivos

De acuerdo con lo descrito en la Sección 1.4, la hipótesis planteada en este trabajo es:

La degradación de los elementos de máquina, sujetos a comportamientos dinámicos, es detectable de una manera más rápida mediante el estudio de los patrones que forman sus principales variables de funcionamiento a lo largo del tiempo.

De acuerdo con esta hipótesis, los objetivos que se persiguen en este trabajo de tesis son:

- Investigar y desarrollar de algoritmos de aprendizaje automático en el campo industrial, específicamente en el sector de la máquina-herramienta y los procesos de tratamiento térmico láser.

- Investigar patrones de funcionamiento mediante la aplicación de algoritmos de *clustering* en datos estáticos de elementos de maquinaria.
- Desarrollar sistemas de *clustering* probabilístico dinámico para monitorizar *online* la degradación de elementos de máquina.
- Desarrollar e implementar algoritmos de selección de variables espacio-temporales para incrementar el rendimiento de algoritmos de aprendizaje automático y reducir las necesidades de almacenamiento de datos históricos.

1.6. Organización del documento

En la Parte I se presentan los antecedentes necesarios (Capítulos 2 – 5). En la Parte II (Capítulos 6 – 8) se presenta todo el trabajo previo realizado para definir qué tipo de algoritmos son apropiados para inducir patrones de funcionamiento de equipamiento industrial. De esta manera, el Capítulo 6 recoge el trabajo realizado con datos reales de un proceso de tratamiento térmico láser, en donde se estudian las diferentes necesidades de tecnologías de aprendizaje automático. En el Capítulo 7 se valoran los diferentes algoritmos de *clustering* para su aplicación en la industria en relación a la generación de patrones de comportamiento de servomotores industriales. En el Capítulo 8 se verifican las capacidades de los algoritmos seleccionados para extraer nuevo conocimiento de un proceso de mecanizado de cigüeñales de automoción.

En la Parte III (Capítulos 9 – 10), se presentan las contribuciones al aprendizaje automático no supervisado desarrolladas en términos de *clustering* probabilístico dinámico. Así, en el Capítulo 9 define en líneas generales el algoritmo de *clustering* probabilístico dinámico. En el Capítulo 10 se desarrollan las herramientas para que este algoritmo pueda trabajar en entornos industriales desconocidos.

En la Parte IV (Capítulo 11), se presentan las conclusiones del trabajo, los logros obtenidos en forma de publicaciones en revistas, congresos, patentes y libro, y el trabajo a futuro que surge de estas tareas de investigación.

Parte I

Antecedentes

Notación y terminología

A continuación se listan las diferentes notaciones y terminología usada a lo largo de los siguientes capítulos para el aprendizaje automático y la industria.

2.1. Aprendizaje automático

Los vectores se denotan con letras en negrita, por ejemplo, \mathbf{x} . Una instancia se caracteriza por valores específicos de las variables asociadas, definidos como datos, con la misma marca de tiempo. Un conjunto de instancias n -dimensionales se denota $\mathcal{D} = \{\mathbf{x}^1, \dots, \mathbf{x}^N\}$, en donde $\mathbf{x}^i = (x_1^i, \dots, x_n^i)$ denota las n coordenadas de la instancia i -ésima del conjunto de datos original.

Si estas instancias se analizan en el mismo momento en el que se producen, se denomina análisis *online*; si se analizan en otro momento posterior, se denomina análisis *offline*.

Cuando las instancias se producen de manera constante e ilimitada, se denomina *data stream*. Si existe un cambio en el comportamiento del *data stream* que ocasiona una desviación en el modelo que representa los datos esto se denomina *concept drift*. En este caso, los algoritmos son entrenados inicialmente con un subconjunto de instancias N y una vez el *concept drift* es detectado, se vuelve a entrenar con una ventana móvil de tamaño N' .

El proceso de generar diferentes grupos de instancias que comparten características similares se denomina *clustering*. Cada grupo es un clúster. Es decir, las instancias se agrupan en K clústers $\{Cl_1, \dots, Cl_K\}$ en donde Cl_k es el k -ésimo clúster y $\mathbf{c}^k = (\mathbf{c}_1^k, \dots, \mathbf{c}_n^k)$ es su centroide correspondiente. Estos clústers se pueden denominar componentes en el caso del clustering probabilístico.

2.2. Industria

De manera resumida, se puede definir una máquina como un conjunto de elementos o componentes que juntos generan una acción específica. Estos elementos se dividen en sensores y actuadores. Los primeros se encargan de recoger información del entorno y los segundos

de transmitir el movimiento a los elementos. Tanto los sensores como los actuadores están centralizados en los sistemas de control de la máquina, denominados controlador lógico programable (PLC) y control numérico por computador (CNC). El PLC es un sistema robusto de tiempo real para el control de los sistemas de acuerdo a señales de entrada produciendo señales de salida. El CNC permite la automatización de las máquinas basado en código computacional. En algunos casos, existen máquinas que llevan ambos sistemas. En tal caso se habla de un sistema PLC/CNC. Si físicamente ambos elementos se encuentran integrados en la misma unidad se refiere a éstos como unidad de control numérico (NCU).

Un sistema de producción es un conjunto de líneas productivas que no necesariamente fabrican el mismo producto. Cada línea productiva es una sucesión de procesos necesarios para fabricar el producto. Estos procesos de fabricación están realizados por máquinas.

Una máquina-herramienta es un tipo de máquina destinada a eliminar material (procesos sustractivos) o añadir material (procesos aditivos), ambos orientados a dar forma a diferentes objetos que pueden ser parte de un producto o un producto completo.

En un proceso de tratamiento superficial por láser se imprime energía en forma de calor sobre una superficie para cambiar las características mecánicas de la misma. La energía es depositada sobre la superficie por el haz láser, que dibujando un patrón sobre la misma genera el perfil térmico necesario para asegurar el tratamiento térmico. Este tipo de procesos son únicamente válidos sobre elementos metálicos, como el acero y sus diferentes variedades.

Aprendizaje automático

Los modelos de desempeño para equipamiento de uso general, como bombas, servomotores o husillos a bolas, se desarrollan mediante modelos teóricos, la experiencia o el estudio de su funcionamiento en condiciones de laboratorio. Es importante resaltar que para todos estos componentes no hay conocimiento preciso sobre su función final, es decir, dónde se instalarán y cuáles serán sus condiciones operativas específicas. Es por ello, que este tipo de equipamiento está diseñado para trabajar dentro de una amplia gama de condiciones de operación, en donde el rendimiento se define como un comportamiento nominal.

Sin embargo, a nivel de máquina, la 4IR descrita en el Capítulo 1, prevé el uso de datos de los diferentes medios productivos para buscar nuevo conocimiento. Este se puede usar para estimar el desempeño de equipos similares, por ejemplo, el efecto de una condición de operación en el rendimiento general o en la degradación. Además, este conocimiento basado en variables se puede extrapolar de las máquinas a otros niveles, como los componentes de máquina, en un nivel inferior, y las plantas de fabricación, en un nivel superior. Esto es importante, ya que también se puede aplicar a procesos productivos, logrando así extraer información útil para la monitorización y control de los mismos. [Lee et al. \(2014\)](#) explican que existen similitudes entre las máquinas que realizan las mismas tareas con el mismo nivel de mantenimiento, donde las condiciones de salud y el rendimiento pueden ser similares, lo que lleva a un patrón potencialmente útil.

Adicionalmente, se ha abierto una amplia gama de aplicaciones, en la que los datos de sensores y otros datos contextuales, combinados con modelos computacionales permiten producir nueva información útil. Esto puede ser utilizado en sistemas de control o monitorización, como se ha explicado antes, e incluso como nuevos servicios ([Kopetz, 2011](#)). Por lo tanto, las tecnologías IIoT basadas en aprendizaje automático son capaces de dar el salto de los modelos teóricos a modelos más cercanos a la realidad, es decir, aquellos basados en datos. Los conocimientos adquiridos con este enfoque podrían ser útiles para comparar información sobre cualquier elemento, desde el rendimiento de la máquina hasta el mantenimiento, lo que tiene un impacto directo en la utilización general de la planta, resultando en una mayor productividad.

De esta manera, las técnicas de la inteligencia artificial, y en particular del aprendizaje

automático, resultan ser la herramienta idónea para su aplicación en este tipo de problemáticas. El aprendizaje automático es un proceso de búsqueda de patrones estructurales en los datos que permiten ser examinados, razonados y utilizados para la toma de decisiones en el futuro. Este proceso puede ser automático o semi-automático permitiendo clasificar, predecir y extraer nuevo conocimiento de los datos.

Los diferentes tipos de aprendizaje automático se pueden clasificar en supervisado, no supervisado y semi-supervisado. El aprendizaje supervisado trata de predecir una variable clase C a partir de sus variables predictoras x^1, \dots, x^N . Cuando C es una variable continua, el método supervisado se denomina regresión. Cuando C es una variable discreta se le denomina clasificación. Por otro lado, el aprendizaje no supervisado no cuenta con una variable clase. En este caso, las variables se agrupan en clústers con gran variabilidad entre ellos y baja variabilidad dentro de ellos. El aprendizaje semi-automático no cuenta con todas las clases.

Es decir, estas tecnologías aplicadas a la industria tienen la capacidad de generar nuevas oportunidades para reducir el tiempo de inactividad de las máquinas y aumentar su disponibilidad, principalmente canalizadas a través de un mantenimiento predictivo, donde se deben predecir los fallos lo antes posible (Mobley, 2002). El aprendizaje automático es muy utilizado en este contexto, ya que los algoritmos son altamente aplicables en la resolución de problemas específicos de mantenimiento predictivo, como la vida útil remanente (RUL) de un activo específico, la diagnosis y la prognosis de fallos, la optimización de procesos, la inspección visual y control de calidad, entre otros. Es decir, se pueden obtener modelos que pueden predecir desviaciones de calidad o fallos durante los procesos industriales.

Sin embargo, la aplicación del aprendizaje automático en la industria presenta varios retos que se deben resolver para lograr una correcta aplicación de estas tecnologías al mundo industrial. Algunas de estas problemáticas son abordadas a lo largo de este trabajo y están relacionadas en gran medida con el ciclo de vida de los datos presentado en la Figura 1.2. Es decir, la realidad industrial hace necesario lo siguiente:

- La imputación de datos. Debido a la existencia de todo tipo de sensores y fuentes de datos que trabajan con diferentes tiempos de muestreo, se generan espacios vacíos que no pueden ser procesados por algunos algoritmos, como los de *clustering* (Sección 3.1).
- La selección de variables. Las restricciones en la transmisión de datos de un lugar a otro hacen necesario la alta eficiencia a la hora de transmitir únicamente aquellas variables que se necesiten para el análisis. Por otro lado, esta selección permite además incrementar la precisión de los algoritmos (Sección 3.2).
- La reducción dimensional. Herramienta que permite analizar e inspeccionar de manera apropiada conjuntos de datos con alta dimensionalidad (Sección 3.3).
- La evaluación del rendimiento de los algoritmos mediante técnicas que permitan establecer el comportamiento de los mismos en relación a problemas como la detección o la generación de falsas alarmas (Sección 3.4).
- El uso de algoritmos no supervisados según la aplicación requerida (Capítulo 4).

- El trabajo con *data stream*, ya que la industria es un entorno dinámico que produce datos dependientes de muchos factores, como la degradación de los medios productivos, para lo que es necesario adaptar también los algoritmos a estos cambios (Capítulo 5).

3.1. Imputación de datos

La imputación de datos es una herramienta de gran importancia que permite rellenar los campos vacíos de una base de datos. Estos campos pueden generarse debido a fallos en la comunicación desde el sensor al sistema de adquisición, pero más comúnmente debido a las diferencias en las velocidades de muestreo de diferentes variables. Por ejemplo, en los datos industriales es normal tener diferentes tipos de variables, como la temperatura, la cual puede ser adquirida cada segundo o un dato del PLC/CNC de la máquina que puede ser adquirido cada 100 *ms*. Si ambas variables se van a usar en un mismo conjunto de datos van a quedar espacios en los que la temperatura no tendrá valores. En este caso, la solución puede ser muy sencilla. Sin embargo, cuando aumenta el número de variables con diferentes tiempos de adquisición entre ellas, la solución va a necesitar una respuesta mucho más elaborada.

A día de hoy existen diferentes técnicas de imputación, desde la más sencilla, que consiste en imputar como el último valor conocido, pasando por el uso de herramientas estadísticas, hasta algunas basadas en aprendizaje automático. [García-Laencina et al. \(2010\)](#) listan estas técnicas, que se resumen a continuación, de acuerdo a su aplicación en este trabajo:

1. Métodos de imputación basados en herramientas estadísticas:

- Valor medio, en donde los campos vacíos son completados con el valor medio de la variable para variables continuas o la moda para variables discretas.
- Regresión, en donde los campos vacíos son completados con el valor predicho por el análisis de regresión.
- *Hot y cold deck*, en donde los valores son reemplazados por el vector completo más parecido (*hot deck*) o por datos que existen en otro conjunto de datos (*cold deck*).
- Múltiple, en donde en lugar de imputar variables de una en una, como los métodos anteriores, se realizan varias posibles imputaciones para todas las variables, generando un nuevo conjunto de datos que puede ser luego analizado.

2. Métodos de imputación basados en herramientas de aprendizaje automático:

- *k*-NN, es un método *hot deck* en donde los valores son imputados según el vector de valores más parecido (o cercano) al vector que contiene valores vacíos. Esta medida de similitud puede ser de varios tipos: correlación de Pearson, distancia euclídea o varianza ([Keerin et al., 2012](#)). En el Capítulo 7 se ha imputado usando la distancia euclídea,

$$dist(\mathbf{x}^i, \mathbf{x}^j) = \sqrt{\sum_{p=1}^N (\mathbf{x}_p^i - \mathbf{x}_p^j)^2} \quad (3.1)$$

en donde \mathbf{x}_p^i y \mathbf{x}_p^j son las instancias y N es el número de instancias total del conjunto de datos. De esta manera, los datos imputados en la instancia \mathbf{x}^i con campos vacíos serán aquellos que pertenezcan a la instancia \mathbf{x}^j que minimice $dist(\mathbf{x}^i, \mathbf{x}^j)$, $j = 1, 2, \dots, N, j \neq i$, \mathbf{x}^j sin valores faltantes. Esto corresponde al 1-NN, pero en general podrán cogerse los k vectores más próximos e imputar con, por ejemplo, su valor medio.

Es importante resaltar que el uso de estas técnicas depende del tipo de tarea que esté resolviendo. Es decir, como se verá en la Sección 5.1, el trabajo en tiempo real, hará imposible el uso de gran parte de estas técnicas, al ser la mayoría de ellas para el uso principalmente *offline*. De esta manera, los conjuntos de datos usados en los Capítulos 8, 9 y 10 han sido imputados en origen directamente por el PLC/CNC usando el último valor conocido, para así poder trabajar con tiempos de muestreo de 100 *ms*.

3.2. Selección de variables

En cuanto a la selección de variables, es importante resaltar que a un nivel amplio de abstracción existen dos formas de seleccionar variables: mediante la experiencia y el conocimiento fundamental de la ingeniería, tal y como se aplica en el Capítulo 7, 8, 9 y 10 o con técnicas de aprendizaje automático, tal y como se describe en el Capítulo 6.

La selección de variables tiene dos aplicaciones fundamentales a nivel industrial. La primera está relacionada con la reducción del tráfico de datos, debido a las restricciones que existen a día de hoy en gran parte de las plantas productivas en cuanto a la infraestructura de comunicaciones, una realidad industrial presente incluso en las plantas más avanzadas del mundo. Esta situación tiene una gran componente económica, ya que el desarrollo de infraestructuras dedicadas a la comunicación en superficies de gran tamaño (muchas plantas productivas tienen la extensión de pequeños pueblos) tiene un coste elevado. Adicionalmente, el uso de tecnologías inalámbricas no está permitido debido a sus vulnerabilidades en cuestión de seguridad e interferencias con el proceso. Debido a esto, es importante reducir al máximo el transporte de datos y limitarlo única y exclusivamente a datos de gran utilidad y valor. Por todo esto, la selección de variables es una técnica apropiada para ello (Saeys et al., 2007).

La segunda, está relacionada con la mejora de la precisión de los algoritmos al reducir de manera controlada la dimensionalidad del problema. Otros beneficios descritos pueden ser facilitar la visualización de los datos para comprenderlos mejor, reducir las necesidades de almacenamiento y tiempos de entrenamiento, y mejorar de la disponibilidad del algoritmo (es decir, el algoritmo podrá procesar más instancias) (Guyon y Elisseeff, 2003).

Las técnicas de selección de variables usadas en el Capítulo 6 se enmarcan dentro del aprendizaje supervisado, donde existe una variable clase categórica C que se ha de predecir. Hay dos tipos de técnicas:

- *Filter*: su función es evaluar la relevancia de las variables. En el Capítulo 6 se han utilizado dos tipos de filtros. El de tipo univariante, llamado *information gain* entre

dos variables X y C definido como:

$$f(X) = \mathbb{I}(\mathbf{X}, C) = - \sum_{\mathbf{x}=1}^r \sum_{c=1}^{r_0} p(\mathbf{x}, c) \log_2 p(\mathbf{x}, c)$$

en donde se mide la reducción en la incertidumbre de uno de ellos (por ejemplo, C) una vez que el valor de la otra variable (por ejemplo X_j) es conocida. Aquí se asume que los posibles valores de \mathbf{X} son $\{1, 2, \dots, r\}$ y de C son $\{1, 2, \dots, r_0\}$. Una desventaja de este tipo de filtro es que no tiene en cuenta las correlaciones ni la redundancia entre las variables seleccionadas.

El otro tipo de filtro es multivariante y tiene en cuenta estas dependencias y la redundancia de las variables. Este tipo de filtro se llama *correlation-based feature subset selection* (CFS) (Hall, 1999). El filtro CFS busca el mejor subconjunto de variables, que sean altamente correlacionadas con la clase, pero sin correlación entre ellas. Formalmente, al denotar con \mathcal{S} un subconjunto de variables predictoras del conjunto de todas las variables \mathbf{X} , el CFS busca

$$\mathcal{S}^* = \arg \max_{\mathcal{S} \subseteq \mathbf{X}} r(\mathcal{S}, C),$$

donde

$$r(\mathcal{S}, C) = \frac{\sum_{X_i \in \mathcal{S}} r(X_i, C)}{\sqrt{k + (k - 1) \sum_{X_i, X_j \in \mathcal{S}} r(X_i, X_j)}}$$

mide la correlación entre cada variable seleccionada \mathcal{S} y la variable clase C (numerador) y entre cada pareja de variables seleccionadas (denominador). k es el número de variables seleccionadas en \mathcal{S} .

- *Wrapper* (Kohavi y John, 1997): este tipo de técnica evalúa cada posible subconjunto de variables con un criterio basado en el rendimiento estimado del clasificador creado con este subconjunto de variables. Una de las principales desventajas de este tipo de técnica es que depende del tipo de clasificador usado, lo que no permite su aplicación a problemas de gran escala (Larrañaga et al., 2019).

3.3. Reducción dimensional

Según el ciclo de vida descrito en la Figura 1.2, el último paso después de la transformación de los datos en información, generalmente con técnicas de aprendizaje automático, está enfocado en la forma de entregar la información. Teniendo en cuenta que muchos resultados deben ser analizados en todas las dimensiones disponibles, suele ser un verdadero problema la interpretación del mismo por parte del usuario, más si carece del conocimiento necesario sobre los detalles de los algoritmos utilizados. Por otro lado, también podría ser necesaria una reducción dimensional como entrada a un algoritmo de aprendizaje automático para facilitar su labor.

Tabla 3.1: Matriz de confusión

		Verdad	
		+	-
Predicción	+	a	b
	-	c	d

Por esta razón, una solución para facilitar la interpretación de los resultados y ayudar a la exploración de los mismos para sacar más y mejores conclusiones es la reducción dimensional. De acuerdo a [Fodor \(2002\)](#), existen varias técnicas de reducción dimensional, de las cuales se ha trabajado especialmente con métodos no lineales y extensiones. Dentro de este tipo de métodos se encuentra el escalado multidimensional ([Groenen y Velden, 2014](#)). Esta técnica es utilizada en el Capítulo 7 para explorar el resultado del *clustering* y se fundamenta en la conservación de las distancias entre los datos originales pero sobre un espacio de dimensión inferior. En el caso del Capítulo 7, estas proyecciones se han hecho en un espacio bi-dimensional. De esta manera, partiendo de la distancia euclídea definida en la Ecuación (3.1) el objetivo es representar la disimilitud entre dos puntos δ_{ij} en un espacio de menor dimensión, en este caso 2-D, en donde la distancia entre esos puntos $dist(\mathbf{x}^i, \mathbf{x}^j)$ es lo más parecida posible a esta disimilitud ([Groenen y Velden, 2014](#)).

$$\sum_{i=2}^N \sum_{j=1}^{i-1} w_{ij} (\delta_{ij} - dist(\mathbf{x}^i, \mathbf{x}^j))^2$$

donde w_{ij} es el peso específico no negativo definido por el usuario.

3.4. Evaluación del rendimiento de un algoritmo de clasificación supervisada

Evaluar el rendimiento de los algoritmos en términos de su capacidad de clasificar nuevas instancias es necesario a la hora de valorar la viabilidad de los mismos para una aplicación industrial. Según la experiencia propia recogida, los algoritmos desplegados en planta deben ser estudiados en cuanto a su capacidad de detectar fenómenos de manera precisa y, también a la problemática asociada con la generación de falsos positivos. Es decir, dependiendo de la criticidad de la aplicación, generar falsas alarmas podría crear un problema similar al de no poder detectar eventos (falso negativo), ya que en base a estas se pueden estar tomando decisiones erróneas en relación a la gestión de la máquina, de la línea o de la planta. Es por esta razón que a lo largo del Capítulo 9 y 10 se intensifica el estudio del rendimiento del algoritmo en estos términos.

3.4.1. Matriz de confusión y valores de mérito

A partir de la matriz de confusión mostrada en la Tabla 3.1 es posible calcular valores de mérito como:

- Precisión: ratio de instancias correctamente clasificadas,

$$\text{Precisión} = \frac{a + d}{a + b + c + d}$$

- Error: ratio de instancias clasificadas incorrectamente,

$$\text{Error} = \frac{b + c}{a + b + c + d}$$

- Sensibilidad o *recall*: ratio de verdaderos positivos,

$$\text{Recall} = \frac{a}{a + c}$$

- Especificidad: ratio de verdaderos negativos,

$$\text{Especificidad} = \frac{d}{b + d}$$

- Exactitud: ratio de verdaderas predicciones,

$$\text{Exactitud} = \frac{a}{a + b}$$

- *F-score*: es la media armónica entre el *recall* y la exactitud,

$$\text{F-score} = 2 \cdot \frac{\text{Recall} \times \text{Exactitud}}{\text{Recall} + \text{Exactitud}}$$

3.4.2. Brier score

Para evaluar la certeza de una asignación, por ejemplo, a diferentes clústers, tal y como se describe en el Capítulo 9, se utiliza el *Brier score* (Brier, 1950). Este permite medir la desviación entre las probabilidades *a posteriori* y las reales. Por lo tanto,

$$\text{Brier} = \frac{1}{N} \sum_{j=1}^K \sum_{i=1}^N (P_{ij} - e_{ij})^2, \quad (3.2)$$

donde P_{ij} es la probabilidad *a posteriori* de que la instancia \mathbf{x}^i se asigne el valor j (ya sea una etiqueta j o un clúster de un total de K posibles), estimado por el modelo, y

$$e_{ij} = \begin{cases} 1 & \text{si } c^i = j \\ 0 & \text{en caso contrario} \end{cases}$$

Por ejemplo, si los P_{ij} de una instancia \mathbf{x}^i para 3 posibles valores forman el vector $(0, 33, 0, 34, 0, 33)$, y e_{ij} es previamente conocido como $(0, 1, 0)$ (es decir, la clase real para \mathbf{x}^i es el valor 2). En consecuencia, utilizando la Ecuación (3.2),

$$Brier = \frac{1}{1} [(0, 33 - 0)^2 + (0, 34 - 1)^2 + (0, 33 - 0)^2] = 0,65.$$

Esta puntuación es mejor cuanto más bajos son los valores, es decir, la probabilidad de asignación está más concentrada en el valor real de la clase y menos distribuida entre los posibles valores.

3.5. Criterio de información Bayesiano

De acuerdo con [McLachlan y Peel \(2004\)](#), se puede obtener un criterio de información para la selección de modelos. Así, esta herramienta permite encontrar de manera iterativa los mejores modelos de acuerdo con la minimización del criterio. Esto resulta útil para la selección automática del número de componentes de las mixturas de Gaussianas descrita en la Sección 4.5 y poder ser aplicada en la versión dinámica del algoritmo GDPC descrito en el Capítulo 10.

La selección de modelos puede realizarse en función de la log-verosimilitud ($\log L$) corregida por el sesgo dada por:

$$\log L(\hat{\Psi}) - b, \quad (3.3)$$

donde $\hat{\Psi}$ es la estimación de los parámetros desconocidos Ψ obtenida mediante el uso de la estimación de máxima verosimilitud o el algoritmo EM. El término b es el sesgo.

El criterio de información para la Ecuación (3.3) generalmente se expresa como:

$$-2 \log L(\hat{\Psi}) + 2W, \quad (3.4)$$

donde el primer término mide la falta de ajuste y $W > 0$ es una penalización para tener en cuenta la complejidad del modelo. La elección de un modelo depende de la minimización de la Ecuación (3.4). Para el criterio de información Bayesiano (BIC), la Ecuación (3.4) se expresa como:

$$-2 \log L(\hat{\Psi}) + \dim(\hat{\Psi}) \log N, \quad (3.5)$$

donde N es el tamaño de la muestra y $\dim(\hat{\Psi})$ es el número de parámetros. Así, en el caso de usar mixturas de Gaussianas para *clustering* probabilístico el número de componentes de la mixtura es el valor de K que minimiza la Ecuación (3.5) para un valor constante de N .

Clustering

El *clustering* es una técnica de aprendizaje automático no supervisada que tiene como objetivo segmentar los conjuntos de datos o instancias en subconjuntos en donde sus coordenadas comparan características similares, dependiendo de la métrica utilizada. Estos subconjuntos se denominan clústers, o también componentes del modelo de mixturas, para el caso del *clustering* probabilístico definido más adelante. De esta manera, existen dos tipos de algoritmos de *clustering* (Larrañaga et al., 2019):

- Probabilísticos (Sección 4.5), en donde una instancia puede pertenecer a varios componentes al mismo tiempo con diferente probabilidad de pertenencia. El tipo de *clustering* proporcionado por estos algoritmos se denomina *soft clustering*. El algoritmo probabilístico es la base fundamental de este trabajo, ya que, como se detalla en el Capítulo 7 y 8, es una herramienta muy útil para la segmentación de datos provenientes de procesos productivos. De esta manera, en estos dos capítulos se desarrollan los aportes metodológicos y de aplicación al mundo industrial relacionados con el *clustering* probabilístico.
- No probabilísticos, en donde una instancia solo pertenece a un clúster específico. A este tipo de *clustering* se le denomina *hard clustering*. Los algoritmos tratados en este trabajo son: jerárquicos (Sección 4.1), K -medias (Sección 4.2), espectral (Sección 4.3) y propagación de afinidad (Sección 4.4). Estos algoritmos son aplicados en el Capítulos 6 (jerárquico), 7 (jerárquico, K -medias, espectral, y propagación de afinidad) y 8 (jerárquico y K -medias).

A continuación, se resumen las características principales de estos algoritmos.

4.1. Clustering jerárquico

Estos algoritmos (Gordon, 1987) representan los datos de manera jerárquica en una estructura llamada dendrograma, tal y como se muestra en la Figura 4.1. En el nivel cero del dendrograma se encuentran todas las instancias que se van uniendo entre sí a medida que

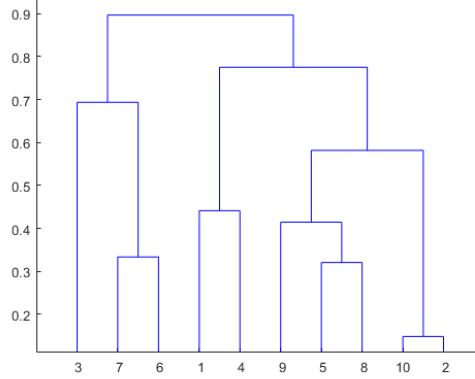


Figura 4.1: Ejemplo de dendrograma.

se va subiendo de nivel. El resultado de este tipo de *clustering* se puede obtener cortando el dendrograma en diferentes niveles dependiendo del número K de clústers deseados.

La vertiente de este tipo de técnica utilizada en el presente trabajo es el *clustering* jerárquico aglomerativo, el cual parte del supuesto de que cada instancia es un clúster, que se va fusionando en pares de clusters dependiendo del tipo de vinculación definida a medida que asciende en la jerarquía del dendrograma. De esta manera, los parámetros utilizados por el *clustering* jerárquico aglomerativo son:

- Método de vinculación: el criterio de vinculación se refiere a la distancia entre dos grupos utilizados por el algoritmo para encontrar la fusión óptima. Se puede utilizar el método de Ward que calcula la diferencia entre dos clústers, Cl_i y Cl_j , como la diferencia entre la suma de las instancias al cuadrado entre cada objeto y su centroide dentro del clúster resultado de la unión de Cl_i y Cl_j y la suma del cuadrado de las distancias al centroide dentro de estos dos clústers por separado. La métrica usada se expresa como:

$$d_{Ward}(Cl_i, Cl_j) = \sum_{\mathbf{x}^k \in Cl_i \cup Cl_j} d(\mathbf{x}^k, \mathbf{c}^{ij}) - \left[\sum_{\mathbf{x}^i \in Cl_i} d(\mathbf{x}^i, \mathbf{c}^i) + \sum_{\mathbf{x}^j \in Cl_j} d(\mathbf{x}^j, \mathbf{c}^j) \right],$$

donde d es la distancia euclídea al cuadrado y \mathbf{c}^{ij} , \mathbf{c}^i y \mathbf{c}^j son los centroides de los clústers $Cl_i \cup Cl_j$, Cl_i y Cl_j respectivamente.

- Número de clusters K : para seleccionar este número no existe un criterio matemático perfecto que permita hacerlo (Tibshirani et al., 2001), por lo tanto debe ser seleccionado de acuerdo a la opinión de expertos en el dominio. Este valor, tal y como se ha comentado antes, define el nivel en el que el dendrograma es cortado.

4.2. *K-medias*

K-medias divide un conjunto de puntos n -dimensionales $\mathcal{D} = \{\mathbf{x}^1, \dots, \mathbf{x}^N\}$ en K clústers Cl_1, \dots, Cl_K basado en la minimización del criterio de error cuadrático definido por [MacQueen \(1967\)](#):

$$F_{K\text{-means}}(\{Cl_1, \dots, Cl_K\}) = \sum_{k=1}^K \sum_{\mathbf{x}^i \in Cl_k} \|\mathbf{x}^i - \mathbf{c}^k\|^2,$$

en donde K es el número de clústers, $\mathbf{x}^i = (x_1^i, \dots, x_n^i)$ denota los n componentes de la instancia i -ésima del conjunto de datos original. Cl_k es el k -ésimo clúster y $\mathbf{c}^k = (c_1^k, \dots, c_n^k)$ es su centroide correspondiente.

De esta manera, los parámetros utilizados para el algoritmo de *K-medias* son:

- Número de clústers K : al igual que en el clústering jerárquico (Sección 4.1), este valor debe ser seleccionado en función de las opiniones de expertos apoyados en varias iteraciones en donde los resultados de clusterización sean significativos.
- Inicialización del clúster: como *K-medias* puede converger a mínimos locales, el algoritmo es sensible a la inicialización. La estrategia más común para la inicialización del *clustering* es elegir K instancias al azar y luego ir buscando el mínimo error cuadrático. Sin embargo, esta técnica no garantiza evitar los mínimos locales, comprometiendo además la velocidad de procesamiento e grandes conjuntos de datos.

La solución para esto es utilizar una variante llamada *K-medias++* definida por [Arthur y Vassilvitskii \(2007\)](#) que además, permite acelerar la convergencia. La idea principal de esta inicialización es tomar un centroide \mathbf{c}^1 , elegido de manera uniforme dentro de \mathcal{D} y asignarle a cada instancia un peso significativo, según la distancia al cuadrado al centroide (mayor distancia da lugar a mayor peso). Posteriormente, se seleccionan secuencialmente $K-1$ nuevos centroides, basados en el peso específico más elevado fijado anteriormente. Esta estrategia garantiza que los centroides estén bien repartidos en el espacio de los datos, reduciendo los errores debidos a los óptimos locales y disminuyendo los cálculos adicionales de particionado.

- Distancia: la distancia utilizada en este trabajo es la euclídea. Esto significa que el algoritmo *K-medias* buscará grupos con forma esférica en el espacio de los datos ([Jain, 2010](#)).

4.3. *Clustering* espectral

El objetivo del *clustering* espectral ([Barnard et al., 1995](#)) es encontrar una partición de un grafo de similitud donde las aristas entre diferentes grupos tienen pesos muy bajos y las aristas dentro de un grupo tienen pesos muy altos. El *clustering* espectral trabaja principalmente con el algoritmo Shi y Malik (SM) ([Shi y Malik, 2000](#)), conocido como un algoritmo de

biparticionado, ya que divide los puntos en dos grupos hasta que se forman las particiones de K . Los parámetros del algoritmo SM son:

1. Grafo de similitud (matriz de afinidad): se construye utilizando las similitudes entre pares o distancias entre pares de instancias en \mathcal{D} . Existen tres grafos de similitud diferentes de los cuales se elige uno:
 - El grafo de vecindad η , en donde las instancias son conectadas únicamente si su valor de similitud es mayor a η .
 - El grafo de k vecinos más cercanos que calcula el gráfico de k -NN para cada $\mathbf{x} \in \mathcal{D}$. De esta manera, el vértice \mathbf{x}^i está conectado con el vértice \mathbf{x}^j si \mathbf{x}^j es el k -vecino más cercano de \mathbf{x}^i .
 - El grafo completamente conectado, en donde todos los puntos con similitud positiva son conectados a todos los vértices. Existen diferentes funciones de similitud que pueden ser aplicadas como una función Gaussiana definida por:

$$S_{ij} = \exp\left(-\frac{\|\mathbf{x}^i - \mathbf{x}^j\|^2}{2\sigma^2}\right),$$

o radial, definida por:

$$S_{ij} = \exp\left(-\frac{\|\mathbf{x}^i - \mathbf{x}^j\|^2}{\sigma^2}\right).$$

2. Una vez se transforma este grafo de similitud se obtiene la matriz de adyacencia ponderada \mathbf{W} a partir de la cual se extraen los grados de las aristas resultantes de la suma de los pesos en todas las aristas adyacentes.
3. Se construye la matriz de grados \mathbf{D} la cual tiene la suma de los pesos en su diagonal.
4. Se genera el grafo Laplaciano $\mathbf{L} = \mathbf{D} - \mathbf{W}$ siendo una representación simplificada de los objetos originales.
5. Luego, se extraen los valores propios más bajos $\mathbf{c}^1, \dots, \mathbf{c}^K$ en donde sus filas contienen la transformación de \mathbf{x}^i con $i = 1, \dots, N$.
6. Estos puntos \mathbf{x}^i pueden ser agrupados en clústers Cl_1, \dots, Cl_K usando K -medias.

4.4. *Clustering* por propagación de afinidad

En el *clustering* por propagación de afinidad (AP) (Frey y Dueck, 2007) la afinidad entre dos puntos se cuantifica mediante dos tipos de mensajes:

- Responsabilidad $r(\mathbf{x}^i, \mathbf{x}^k)$, tiene en cuenta la evidencia acumulada de lo bien que se adapta a \mathbf{x}^k para servir como ejemplar (centroide o representante) \mathbf{x}^i con respecto a otros posibles ejemplares. De esta manera, la responsabilidad se actualiza de la siguiente manera:

$$r(\mathbf{x}^i, \mathbf{x}^k) \leftarrow s(\mathbf{x}^i, \mathbf{x}^k) - \max_{k' \neq k} \{a(\mathbf{x}^i, \mathbf{x}^{k'}) + s(\mathbf{x}^i, \mathbf{x}^{k'})\}.$$

en donde $s(\mathbf{x}^i, \mathbf{x}^k)$ es la similitud entre la instancia \mathbf{x}^i y el posible ejemplar \mathbf{x}^k . $a(\mathbf{x}^i, \mathbf{x}^{k'})$ es la disponibilidad que se define a continuación.

- Disponibilidad $a(\mathbf{x}^i, \mathbf{x}^k)$, tiene en cuenta la evidencia acumulada de lo apropiado que sería para \mathbf{x}^i elegir \mathbf{x}^k como su ejemplar. La disponibilidad se actualiza de la siguiente manera:

$$a(\mathbf{x}^i, \mathbf{x}^k) \leftarrow \min \left\{ 0, r(\mathbf{x}^k, \mathbf{x}^k) + \sum_{i' \notin \{i, k\}} \max\{0, r(\mathbf{x}^{i'}, \mathbf{x}^k)\} \right\}$$

$$a(\mathbf{x}^k, \mathbf{x}^k) \leftarrow \sum_{i' \neq k} \max\{0, r(\mathbf{x}^{i'}, \mathbf{x}^k)\}.$$

De acuerdo con esto, la parada del algoritmo se obtiene cuando para el objeto \mathbf{x}^i y el objeto \mathbf{x}^j que maximiza $a(\mathbf{x}^i, \mathbf{x}^j) + r(\mathbf{x}^i, \mathbf{x}^j)$ identifica su ejemplar.

Los principales parámetros para el algoritmo AP son la preferencia y la amortiguación:

- Preferencia, expresada como $s(\mathbf{x}^i, \mathbf{x}^i)$, es la idoneidad *a priori* de un punto \mathbf{x}^i para servir de ejemplar. Por lo tanto, la preferencia controla el número de ejemplares (clústers) que se encuentran. Un valor de preferencia alto puede producir una gran cantidad de ejemplares, mientras que los valores bajos pueden producir una pequeña cantidad de ejemplares. Las preferencias deben establecerse en un valor común, generalmente, en el mínimo o la mediana de las similitudes de entrada.
- La amortiguación es el factor $\xi \in (0, 1)$ que limita el alcance durante el cálculo de la responsabilidad y la disponibilidad, es decir, controla los cambios en los mensajes. Por lo tanto, una alta amortiguación produce cambios más pequeños en los mensajes. Sin embargo, el tiempo de procesamiento necesario para converger puede aumentar. Por el contrario, una baja amortiguación puede causar un alcance excesivo, penalizando así la convergencia.

4.5. *Clustering* basado en Mixturas Gaussianas

El *clustering* por modelos de mixturas de Gaussianas (GMM) (McLachlan y Peel, 2004) utiliza la distribución de probabilidad como una medida de similitud entre objetos desconocidos, por ejemplo, objetos con la misma media pero diferente varianza. El algoritmo GMM necesita buscar un modelo de mixtura de Gaussianas, cuyos componentes puedan tener una matriz de covarianza similar (Biernacki et al., 2000).

Suponiendo que la densidad del k -ésimo clúster viene dada por la densidad Gaussiana $f_k = (\mathbf{x}; \boldsymbol{\theta}_k)$, el modelo se define como:

$$f(\mathbf{x}; \boldsymbol{\Psi}) = \sum_{k=1}^K \pi_k f_k(\mathbf{x}; \boldsymbol{\theta}_k), \quad (4.1)$$

con parámetros

$$\boldsymbol{\Psi} = (\pi_1, \dots, \pi_K, \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K),$$

donde π_k son los pesos específicos de cada componente k dentro de la mezcla, $\boldsymbol{\theta}_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ con $k = 1, \dots, K$, y $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ son el vector de medias y la matriz de covarianzas de cada Gaussiana o componente k , respectivamente.

El parámetro desconocido $\boldsymbol{\Psi}$ se estima por el método de máxima verosimilitud (ML). Como no existe expresión cerrada, esta estimación se puede abordar utilizando el algoritmo EM (Dempster et al., 1977; McLachlan y Peel, 2004). El algoritmo EM es un método iterativo que tiene dos pasos: (1) esperanza o paso-E, donde se crea una función para estimar la verosimilitud; y (2) maximización o paso-M, donde los parámetros se calculan maximizando la verosimilitud esperada. Estos dos pasos se resumen a continuación.

4.5.1. Paso-E

Partiendo de la verosimilitud de $\boldsymbol{\Psi}$ definida como:

$$L(\boldsymbol{\Psi}) = \prod_{i=1}^N \left(\sum_{k=1}^K \pi_k f_k(\mathbf{x}^i; \boldsymbol{\theta}_k) \right)$$

entonces, la log-verosimilitud se define como:

$$\begin{aligned} \log L(\boldsymbol{\Psi}) &= \sum_{i=1}^N \log \left(\sum_{k=1}^K \pi_k f_k(\mathbf{x}^i; \boldsymbol{\theta}_k) \right) \\ &= \sum_{k=1}^K \sum_{i=1}^N z_{ki} \{ \log \pi_k + \log f_k(\mathbf{x}^i; \boldsymbol{\theta}_k) \}, \end{aligned} \quad (4.2)$$

donde z_{ki} contiene 0 o 1, dependiendo de la asignación (no/si) de cada instancia entrante i a cada componente. Su variable aleatoria correspondiente Z_{ki} se define como:

$$Z_{ki} = \begin{cases} 1 & \text{Si } \mathbf{x}^i \text{ es asignado al componente } k \\ 0 & \text{En otro caso} \end{cases}$$

El valor de Z_{ki} es estimado por el paso-E en cada iteración. Las iteraciones posteriores t para el paso-E solo requieren encontrar la probabilidad posterior de que la instancia i -ésima esté asignada al k -ésimo componente, es decir, $\tau_k^{(t)}(\mathbf{x}^i; \boldsymbol{\Psi}^{(t)})$. Por lo tanto,

$$\begin{aligned}\mathbb{E}_{\Psi^{(t)}} \{Z_{ki}|\mathcal{D}\} &= P_{\Psi^{(t)}} \{Z_{ki} = 1|\mathcal{D}\} \\ &= \tau_k^{(t)}(\mathbf{x}^i; \Psi^{(t)}),\end{aligned}$$

donde

$$\tau_k^{(t)}(\mathbf{x}^i; \Psi^{(t)}) = \frac{\pi_k^{(t)} f_k(\mathbf{x}^i; \boldsymbol{\theta}_k^{(t)})}{f(\mathbf{x}^i; \Psi^{(t)})}. \quad (4.3)$$

Para un GMM, la Ecuación (4.3) es:

$$\tau_k^{(t)}(\mathbf{x}^i; \Psi^{(t)}) = \frac{\pi_k^{(t)} \mathcal{N}(\mathbf{x}^i | \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)})}{\sum_{j=1}^K \pi_j^{(t)} \mathcal{N}(\mathbf{x}^i | \boldsymbol{\mu}_j^{(t)}, \boldsymbol{\Sigma}_j^{(t)})}$$

De esta manera, al usar la Ecuación (4.2) y la Ecuación (4.3) es posible definir la esperanza condicional de $\log L(\Psi)$ dada \mathcal{D} como:

$$\begin{aligned}\mathbb{E}_{\Psi^{(t)}} \{\log L(\Psi)|\mathcal{D}\} &= Q(\Psi; \Psi^{(t)}) \\ &= \sum_{k=1}^K \sum_{i=1}^N \tau_k^{(t)}(\mathbf{x}^i; \Psi^{(t)}) \{\log \pi_k + \log f_k(\mathbf{x}^i; \boldsymbol{\theta}_k)\}.\end{aligned}$$

Una vez encontrada la esperanza en la iteración t , se aplica el paso-M.

4.5.2. Paso-M

La iteración $t + 1$ para paso-M busca maximizar $Q(\Psi; \Psi^{(t)})$ con respecto a Ψ para actualizar el valor de $\Psi^{(t+1)}$. Los pesos específicos estimados, $\pi_k^{(t+1)}$, se encuentran de manera independiente al vector de parámetros desconocidos $\boldsymbol{\theta}^{(t+1)}$ mediante

$$\pi_k^{(t+1)} = \frac{1}{N} \sum_{i=1}^N \tau_k^{(t)}(\mathbf{x}^i; \Psi^{(t)}) \quad (k = 1, \dots, K).$$

Para obtener $\boldsymbol{\theta}^{(t+1)}$ se resuelve el siguiente sistema de ecuaciones

$$\sum_{k=1}^K \sum_{i=1}^N \tau_k^{(t)}(\mathbf{x}^i; \Psi^{(t)}) \frac{\partial \log f_k(\mathbf{x}^i; \boldsymbol{\theta}_k)}{\partial \boldsymbol{\theta}_k} = 0. \quad (4.4)$$

Para un GMM,

$$\begin{aligned}f_k(\mathbf{x}^i; \boldsymbol{\theta}_k) &= f_k(\mathbf{x}^i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ &= (2\pi)^{-\frac{n}{2}} |\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (\mathbf{x}^i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}^i - \boldsymbol{\mu}_k) \right\},\end{aligned} \quad (4.5)$$

donde los parámetros $\boldsymbol{\mu}_k^{(t+1)}$ y $\boldsymbol{\Sigma}_k^{(t+1)}$ ($k = 1, \dots, K$) son estimados utilizando la Ecuación

ción (4.4), resultando

$$\boldsymbol{\mu}_k^{(t+1)} = \frac{\sum_{i=1}^N \tau_k^{(t)}(\mathbf{x}^i; \boldsymbol{\Psi}_k^{(t)}) \mathbf{x}^i}{\sum_{i=1}^N \tau_k^{(t)}(\mathbf{x}^i; \boldsymbol{\Psi}_k^{(t)})} \quad (k = 1, \dots, K),$$

y

$$\boldsymbol{\Sigma}_k^{(t+1)} = \frac{\sum_{i=1}^N \tau_k^{(t)}(\mathbf{x}^i; \boldsymbol{\Psi}_k^{(t)}) (\mathbf{x}^i - \boldsymbol{\mu}_k^{(t+1)}) (\mathbf{x}^i - \boldsymbol{\mu}_k^{(t+1)})^T}{\sum_{i=1}^N \tau_k^{(t)}(\mathbf{x}^i; \boldsymbol{\Psi}_k^{(t)})} \quad (k = 1, \dots, K).$$

A medida que la verosimilitud aumenta monótonicamente con las iteraciones del algoritmo EM, un criterio estándar de parada se produce cuando

$$L(\boldsymbol{\Psi}^{(t+1)}) - L(\boldsymbol{\Psi}^{(t)})$$

es lo suficientemente pequeño como para indicar la convergencia de la secuencia de parámetros $\{\boldsymbol{\Psi}^{(t)}\}$.

De esta manera, los parámetros del algoritmo GMM son los siguientes:

- Tipo de covarianza: se refiere al tipo matriz que debe ser encontrada. Puede ser completo, en donde cada componente tiene su propia matriz de covarianza $\boldsymbol{\Sigma}_k$; enlazado, donde todos los componentes comparten la misma matriz de covarianza ($\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma} \forall k$); diagonal, donde cada componente tiene su propia matriz diagonal ($\boldsymbol{\Sigma}_k = \boldsymbol{\lambda}_k \mathbb{I}$); o esférica, donde los componentes tienen la misma matriz diagonal ($\boldsymbol{\Sigma}_k = \boldsymbol{\lambda} \mathbb{I}$).
- Inicialización: al igual que con K -medias, el algoritmo EM es sensible a la inicialización y puede converger a mínimos locales con resultados diferentes. Por lo tanto, existen dos tipos de inicialización:
 - Aleatorio.
 - K -medias, donde la salida de este algoritmo se usa para obtener componentes iniciales.
- Número de componentes: elegidos como se describe para el algoritmo K -medias o de manera automática mediante el BIC, descrito en la Sección 3.5.

Capítulo 5

Clustering de data streams

La mayor parte de los desarrollos de aprendizaje automático suponen modos de comportamiento estables en los sistemas analizados. Haciendo especial énfasis en las aplicaciones industriales, esta suposición se aplicaría a lo largo de todo un proceso industrial (por ejemplo, mecanizado, pintura, soldadura, etc.), significando esto que las condiciones de contorno (es decir, degradación de los elementos, materia prima, herramientas, temperatura, etc.) del problema específico son invariantes en el tiempo. La realidad es que este supuesto no se mantiene en la mayoría de las aplicaciones industriales, en donde las variaciones de las condiciones de operación son comunes entre los activos productivos.

Adicionalmente, casi todos estos algoritmos se basan en el análisis del conjunto total de los datos, lo cual genera la necesidad de sistemas de almacenamiento de gran tamaño y complejidad, así como estrategias de gestión de los mismos altamente complejas. Es importante resaltar que las problemáticas industriales necesitan una respuesta apropiada en el momento requerido, por ejemplo, para evitar daños adicionales cuando se produce un fallo en una máquina (Isaacs et al., 2017). Sin embargo, las capacidades de los algoritmos están limitadas por su propia complejidad y también por la potencia de cálculo.

Debido a este comportamiento dinámico presente en este tipo de aplicaciones industriales, como el análisis de degradación de los componentes de una máquina o la monitorización de su condición de operación, estas aplicaciones requieren un análisis *online*, en donde los algoritmos deben poder procesar los datos en tiempo real sin tener necesidad de conocimiento previo de los patrones de degradación de los elementos de la máquina.

5.1. *Data streams*

Debido a las problemáticas descritas anteriormente, en aplicaciones industriales altamente dinámicas, es necesario trabajar con sistemas de aprendizaje automático continuo basado en los datos más actuales posibles. Así, es posible analizar los modos de operación del sistema. Este tipo de aplicaciones dinámicas requieren un flujo continuo de datos, denominado *data stream* (Faria et al., 2016).

Las principales características de un *data stream* son la continuidad ilimitada de los datos que fluyen a la misma velocidad de adquisición, y lo más importante: su distribución, que puede cambiar con el tiempo (Silva et al., 2013). Esta situación es definida por Gama et al. (2014) como un *concept drift*, siendo un cambio en el tiempo entre la relación de los datos de entrada y la variable objetivo. Este cambio podría ser provocado por variaciones específicas en el comportamiento, como la degradación, algo común en muchos entornos industriales y una herramienta de gran utilidad para la detección temprana de fallos. Adicionalmente, también resulta útil para detectar la degradación de los sensores (Frederickson et al., 2017).

Por otro lado, la heterogeneidad de los datos dificulta el análisis de *data streams* en tiempo real (Koložali et al., 2016). Con lo cual, habilita este tipo de análisis como una herramienta importante para detectar estas las variaciones en los patrones de datos heterogéneos y el posible modelo implícito en entornos dinámicos (Gama, 2010) y en tiempo real, necesitando la capacidad de los algoritmos de aprendizaje automático para adaptarse a esta configuración. Por lo tanto, serían necesarias técnicas que operen con tamaños de ventanas de datos adaptativas, que permitan reducir la necesidad de consultar todo el conjunto de datos cada vez que se realiza una actualización del modelo, planteando una posible solución para mejorar el rendimiento.

De acuerdo a esto, ya que el *data stream* y la dinámica del proceso son una consideración importante para la industria, el estudio y la detección de estas variaciones (*concept drift*) son las principales motivaciones de este trabajo de investigación. En especial, este trabajo se fundamenta en la hipótesis planteada descrita en la Sección 1.5, donde se propone el *clustering* como una herramienta válida para estudiar estos comportamientos de degradación a lo largo del tiempo.

A continuación, se describen los métodos de *clustering* aplicados al análisis de *data streams* encontrados en la literatura.

5.2. *Clustering* aplicado a *data streams*

En el escenario industrial descrito anteriormente, el aprendizaje no supervisado es más apropiado ya que es difícil disponer de instancias etiquetadas. Específicamente, las técnicas de *clustering* con capacidades de procesamiento de *data streams*. Sin embargo, en este campo el análisis de *data stream* aún está en desarrollo y se limita a tipos de algoritmos específicos basados en densidad y distancia (Silva et al., 2013; Puschmann et al., 2017).

La Tabla 5.1 resume los diferentes métodos que se pueden emplear para el *clustering* de *data streams* los cuales se agrupan de acuerdo con el método subyacente de la siguiente manera: *K*-medias, jerárquicos, *clustering* espacial basado en densidad de aplicaciones con ruido (DBSCAN) y versiones híbridadas de los anteriores.

Tabla 5.1: Algoritmos de *clustering* para *data streams*

Tipo	Nombre del algoritmo y referencia	Descripción
K-medias	BIRCH (Zhang et al., 1997).	Algoritmo incremental basado en una estructura de datos llamada <i>CF-Tree</i> que resume la forma de los grupos generados por <i>K-medias</i> .
	CluStream (Aggarwal et al., 2003).	Este algoritmo divide el proceso de <i>clustering</i> en dos pasos: <i>online</i> , para almacenar estadísticas que resumen los datos y <i>offline</i> , donde se utilizan las estadísticas de resumen para aumentar la eficiencia de procesamiento en la secuencia de datos, es decir, los procesos con alto costo computacional se ejecutan <i>offline</i> . En este algoritmo <i>K-medias</i> se utiliza en el paso <i>online</i> para inicializar los micro-clústers y los <i>K-medias</i> modificados se utilizan para crear un macro-clúster.
	DGClust (Gama et al., 2011).	Algoritmo distribuido que controla los cambios en los datos de los nodos de adquisición y los reporta a un servidor central, en donde se clasifican de acuerdo al que mayor número de cambios tenga. Allí, un <i>K-medias</i> adaptativo se usa para encontrar los clústers.
	StreamKM ++ (Ackermann et al., 2012).	Es similar al algoritmo BIRCH, pero realiza un muestreo no uniforme aplicando el algoritmo <i>K-medias</i> ++ para generar los clústers. Se utiliza una estructura de datos diferente en comparación con CluStream, que se basa en una técnica de combinación y reducción (Har-Peled y Mazumdar, 2004) para resumir el tamaño y la forma de los grupos.
	SWClustering (Zhou et al., 2008).	Similar al algoritmo BIRCH, pero la estructura de datos empleada se basa en histogramas para rastrear la evolución de los clústers.
Jerárquico	ODAC (Rodrigues et al., 2008).	El algoritmo mantiene una jerarquía de clústers en forma de árbol, que evoluciona con los datos. La división es dinámica y el proceso de fusión se activa por cambios en los diámetros de los clústers.
DBSCAN	D-Stream (Chen y Tu, 2007).	Similar a CluStream, en donde el algoritmo divide el proceso de agrupación en dos pasos: <i>online</i> , para asignar cada instancia en una cuadrícula en el espacio de datos y <i>offline</i> , mediante el uso de DBSCAN para calcular la densidad de la cuadrícula y los clústers en las diferentes cuadrículas.
	DenStream (Cao et al., 2006).	Dos estructuras llamadas <i>core-micro-cluster</i> y <i>outlier-micro-cluster</i> se crean usando DBSCAN y se emplean para resumir los clústers con formas arbitrarias y valores atípicos, respectivamente.
Híbridos	ClusTree (Kranen et al., 2011).	Este algoritmo utiliza una mezcla de <i>clustering</i> jerárquico y <i>K-medias</i> , donde controla la antigüedad de los datos y asigna más importancia a los datos más recientes. Por lo tanto, el resumen del <i>data stream</i> se mantiene mediante una estructura de clasificación adaptativa.

De acuerdo a esto, existe una clara limitación en la variedad de métodos existentes, ya que no existen herramientas de *clustering* dinámico con algoritmos subyacentes como propagación de la afinidad, espectrales o GMM. Es por ello, que en la Parte III se propone un algoritmo para procesamiento de *data streams* basado en GMM.

Parte II

Clustering aplicado a procesos industriales

Tratamiento térmico por láser. Adquisición y tratamiento de datos industriales

La adquisición y el tratamiento de los datos industriales han sido unas de las principales motivaciones para comenzar a pensar en las técnicas de *clustering* como herramientas que podrían llegar a resolver problemas reales, encontrados a día de hoy en las plantas industriales. Aunque el presente capítulo no habla de *clustering* para resolver un problema específico, sí que representa un punto de partida fundamental que ha permitido desarrollar esta temática.

De hecho, esta primera aproximación a un problema real ha servido para el desarrollo y el soporte de otro tipo de técnicas de aprendizaje automático que han permitido reflexionar en mejores formas de resolver algunos puntos importantes del mismo como por ejemplo, la reducción dimensional basado en *clustering* como entrada a otros algoritmos.

De esta manera, a continuación se presenta la problemática resuelta y al final de la misma se describe esta reflexión a partir de la cual el *clustering* define el camino a seguir por el presente trabajo.

Este capítulo aparece publicado en [Diaz et al. \(2016\)](#).

6.1. Proceso térmico del láser

Hoy en día, las aplicaciones de láser están creciendo en muchos sectores como la medicina, la metrología, las telecomunicaciones y las aplicaciones industriales ([Pérez y López, 2013](#)). Sin embargo, las aplicaciones industriales representan cerca del 63 % del mercado global (1671 millones de dólares americanos en 2015), donde los láseres de alta potencia utilizados para el macro-procesamiento son los más interesantes debido a su potencial de aplicación en la industria manufacturera. Cerca del 75 % de estas aplicaciones están relacionadas con el corte de chapa metálica y el 25 % restante está relacionado con otros procesos como la perforación

o los tratamientos térmicos¹.

Como resultado de esto, las tecnologías láser han evolucionado para permitir su aplicabilidad en el sector de fabricación, donde la robustez, la repetibilidad y la reproducibilidad son factores clave. En consecuencia, se han desarrollado tecnologías de monitorización de procesos para permitir el control de calidad dentro de los mismos.

Al nivel del control de proceso del tratamiento térmico de superficies con láser, el objetivo principal se basa en regular la cantidad de energía depositada sobre la superficie tratada por la fuente del haz láser (Pérez y López, 2013). El objetivo de esta deposición energética es aumentar la temperatura de la superficie y modificar las propiedades mecánicas de la misma, donde los fenómenos físicos son esencialmente térmicos (Pérez et al., 2007; Badkar et al., 2011; Perez et al., 2007).

Básicamente, el enfriamiento repentino de la superficie que genera el tratamiento térmico ocurre cuando un haz láser de alta densidad se enfoca en un área pequeña donde causa un calentamiento localizado. En el caso de los aceros, esto implica alcanzar la temperatura de austenización (es decir, dependiendo del acero, una temperatura entre 800°C y 925°C). Cuando se mueve el haz láser, el área sufre un tratamiento de temple inmediato debido al enfriamiento repentino causado por la conducción de calor hacia el metal circundante (Groover, 2007).

Debido a que fundamentalmente se trata de un fenómeno térmico, los sistemas de monitorización de procesos láser se basan principalmente en la lectura de temperatura, donde a menudo se utilizan pirómetros sin contacto y cámaras termográficas. Ya que los sistemas de monitorización buscan garantizar la calidad del proceso, con un sistema de control de baja latencia, los pirómetros son la opción más utilizada (Qian et al., 2014).

Sin embargo, en un tratamiento térmico superficial, por lo general el área de proceso es más grande que el área cubierta por el pirómetro, lo que aumenta el riesgo de perder información crítica para controlar efectivamente el proceso. Debido a esto, para lograr esta calibración se necesitan costosos procesos de análisis para estimar el punto de medición óptimo. Particularmente, en el proceso al que se hace referencia en este capítulo, el área tratada es más de quince veces el rango de cobertura superficial del pirómetro, por lo que no es viable cubrir el área completa de trabajo.

Como resultado del problema mencionado anteriormente, los nuevos desarrollos se están orientando hacia el uso de cámaras termográficas y herramientas de procesamiento de imágenes para poder cubrir áreas más amplias, tal y como se describe en la literatura (Thombansen et al., 2014). Básicamente, estos desarrollos se basan en el procesamiento térmico de imágenes para obtener una matriz de temperaturas sobre la superficie tratada, que se puede analizar y utilizar como una matriz de pseudo-pirómetros.

No obstante, el control del proceso basado en esta técnica tiene un inconveniente importante con respecto a la latencia del sistema. Debido a la gran cantidad de información analizada en cada ciclo del proceso, la potencia de cálculo puede comprometer la latencia del sistema, que está directamente relacionada con el tamaño de la imagen y el tiempo de mues-

¹<http://goo.gl/z2nXkH>

treo. Esta alta latencia impide el uso de una estrategia de control apropiada para asegurar la calidad en el proceso.

Por lo tanto, el objetivo de este capítulo es aprovechar la capacidad de los CPSs para manejar grandes cantidades de información integrada con capacidades de procesamiento (Kagermann et al., 2013; Baheti y Gill, 2011; Babiceanu y Seker, 2015) de manera que reduzca la necesidad de potencia de cómputo para disminuir la latencia de un proceso de control de cámara termográfica. Además, el uso de las tecnologías de monitorización disponibles (en este caso, cámaras termográficas) cumple con algunas recomendaciones básicas de la *Industrie 4.0* descritas en el Capítulo 1, lo que explica que no se trata de poner más sensores dentro de los procesos (Diaz et al., 2015; Posada et al., 2015), sino de extraer información de los que están disponibles.

Como resultado de esta reducción de latencia, el sistema de monitorización tendrá capacidades de control de calidad en el proceso. En este caso, las técnicas de aprendizaje automático se utilizan para identificar patrones dentro de la imagen térmica que se pueden clasificar para detectar fluctuaciones durante el proceso. Este patrón generado por cada fluctuación se usará para desarrollar modelos predictivos que permitan controlar la potencia del láser sin la necesidad de un completo análisis de toda la imagen.

6.2. Metodología para el manejo de los datos

Debido a que la eficiencia en la deposición energética del haz láser está fuertemente relacionada con la rugosidad de la superficie, gracias a los cambios en la reflectividad de la misma, los parámetros del proceso deben ajustarse para trabajar con diferentes tipos de acabado superficial. Es decir, una menor rugosidad produce una mayor reflectividad, lo que significa que se absorbe menos energía del láser. Por lo tanto, uno de los propósitos principales del control del proceso de tratamiento térmico es poder proporcionar la cantidad de energía requerida independientemente de la rugosidad de la superficie.

Debido a esto, para la experimentación surgen dos escenarios: superficies de trabajo rectificadas y sin rectificar. En consecuencia, las técnicas de aprendizaje automático se utilizan para clasificar el tipo de superficie de la pieza dentro de un límite de tiempo aceptable. Este tiempo tendrá una fuerte influencia sobre el control del haz láser, ya que se debe tener el tiempo suficiente para modificar el comportamiento de la fuente láser: corregir las necesidades de energía y lograr la temperatura óptima del proceso.

Por lo tanto, la configuración experimental está orientada a recopilar datos reales de un proceso de tratamiento térmico superficial con láser y evaluar el rendimiento de las técnicas de aprendizaje automático para permitir su aplicación dentro de un dispositivo CPS. En consecuencia, el sistema de adquisición de datos y el modelo de aprendizaje automático se explican en las siguientes secciones.

Tabla 6.1: Tamaños y cantidades de muestras de cilindros.

Tipo de cilindro	Diámetro (<i>mm</i>)	Altura (<i>mm</i>)	Cilindros rectificados	Cilindros no rectificados	Cilindros totales
M	67,85	19,19	24	24	48
P	56,60	15,24	36	36	72

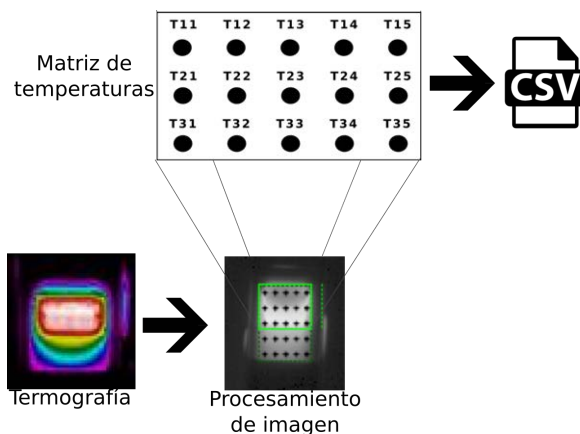


Figura 6.1: Proceso de adquisición de datos.

6.2.1. Adquisición de datos del tratamiento térmico con láser

Como se ha mencionado anteriormente, el parámetro para controlar el proceso de tratamiento térmico del láser es la cantidad de energía depositada sobre la superficie. En una estrategia de control real de un proceso láser el parámetro a controlar es la potencia necesaria de la fuente del haz láser para alcanzar la temperatura de tratamiento térmico, siendo la temperatura de la superficie la variable de entrada tratada durante el proceso.

Para obtener la temperatura de la superficie se han realizado experimentos en 12 cigüeñales de automoción (Figura 10.5) de los cuales 6 tenían su superficie rectificada y 6 sin rectificar. Estas piezas se pueden simplificar como un conjunto de cilindros con las dimensiones descritas en la Tabla 6.1. Cada juego contiene 4 cilindros tipo M (posiciones de apoyo del cigüeñal en el bloque del motor) y 6 cilindros tipo P (muñequillas de conexión con la biela del motor). Para obtener la temperatura del proceso, se ha utilizado una cámara termográfica FLIR A655sc 25^o con 640×480 píxeles a 25/12,5 Hz para registrar cada ciclo, es decir se producen 25/12,5 cuadros por segundo con 307200 píxeles. El tiempo del ciclo es de 20 a 25 s, dependiendo del tamaño del cilindro.

Para extraer los datos de temperatura de cada cuadro de 307200 píxeles se ha utilizado un software de procesamiento de imágenes que permite desplegar sobre la zona afectada térmicamente por láser una matriz de 3×5 pirómetros virtuales. El tamaño de la zona de tratamiento es de 20×10 mm. De esta manera, las medidas tomadas de la matriz de pirómetros virtuales se almacenan en un fichero CSV para su posterior manipulación. El proceso completo se muestra en la Figura 6.1.

6.2.2. Configuración experimental

Desde un punto de vista de aprendizaje automático, para cada tipo de cilindro descrito en la Tabla 6.1, un modelo de *naïve Bayes* clasificará su tipo de superficie según la información proporcionada por las variables, es decir, los valores de temperatura, contenidos en cada cuadro de termografía. Por lo tanto, el porcentaje de instancias clasificadas correcta e incorrectamente se utilizará como una medida cuantitativa para el desempeño del clasificador.

El clasificador *naïve Bayes* es el clasificador Bayesiano más simple. Está basado en el supuesto de independencia condicional de las variables predictivas dada la clase. Aunque, esta suposición no se cumple en numerosas ocasiones en situaciones reales, el clasificador tiene la capacidad de trabajar correctamente en muchos casos. De esta manera, la asignación más probable *a posteriori* de la variable clase se calcula como predicción. Formalmente, este clasificador se expresa como la búsqueda de:

$$c^* = \arg \max_c p(c|\mathbf{x}) = \arg \max_c p(c) \prod_{i=1}^N f(x^i|c)$$

en donde $f(x^i|c)$ es la probabilidad de x^i dado c , donde x^i ($i = 1, \dots, N$) son las variables predictoras, es decir, los píxeles de cada cuadro y c la variable clase, es decir, si es la superficie es rectificadora o no.

En consecuencia, para evaluar el tiempo mínimo requerido por el clasificador para detectar el tipo de superficie del cilindro, se entrenará el modelo con una cantidad incremental de fotogramas. Cada fotograma representa una cantidad de tiempo específica, dependiendo de la velocidad de la cámara termográfica, es decir, 12,5 *Hz* es equivalente a un fotograma cada 80 *ms* y 25 *Hz* es equivalente a un fotograma cada 40 *ms*. Por lo tanto, si el clasificador necesita 1 fotograma para alcanzar la precisión deseada, esto significará que el modelo puede dar un resultado en 80 *ms* a 12,5 *Hz*. Para ello, se han realizado experimentos para cilindros tipo M y P de 1 a 20 cuadros, lo que sitúa la longitud del vídeo entre 0,8 y 1,6 *s*, dependiendo de la velocidad de la cámara.

Después de realizar experimentos para cada tipo de cilindro, se han aplicado 3 tipos de técnicas de selección de variables (FS) diferentes, buscando mejorar la precisión del clasificador. Estas técnicas de FS ayudan a los modelos clasificadores a lidiar con grandes cantidades de información irrelevante (Saeys et al., 2007), mejorando así su rendimiento. De esta manera, las técnicas usadas en este capítulo son: *correlation-based feature selection* (CFS), *information gain* y *wrapper* y su principal objetivo es seleccionar las temperaturas dentro de la matriz definida. Estas técnicas se describen en profundidad en la Sección 3.2.

La experimentación completa para la clasificación se ha ejecutado en Weka, que es una colección de algoritmos de aprendizaje automático (Hall et al., 2009). Y se han verificado con validación cruzada *k-fold*.

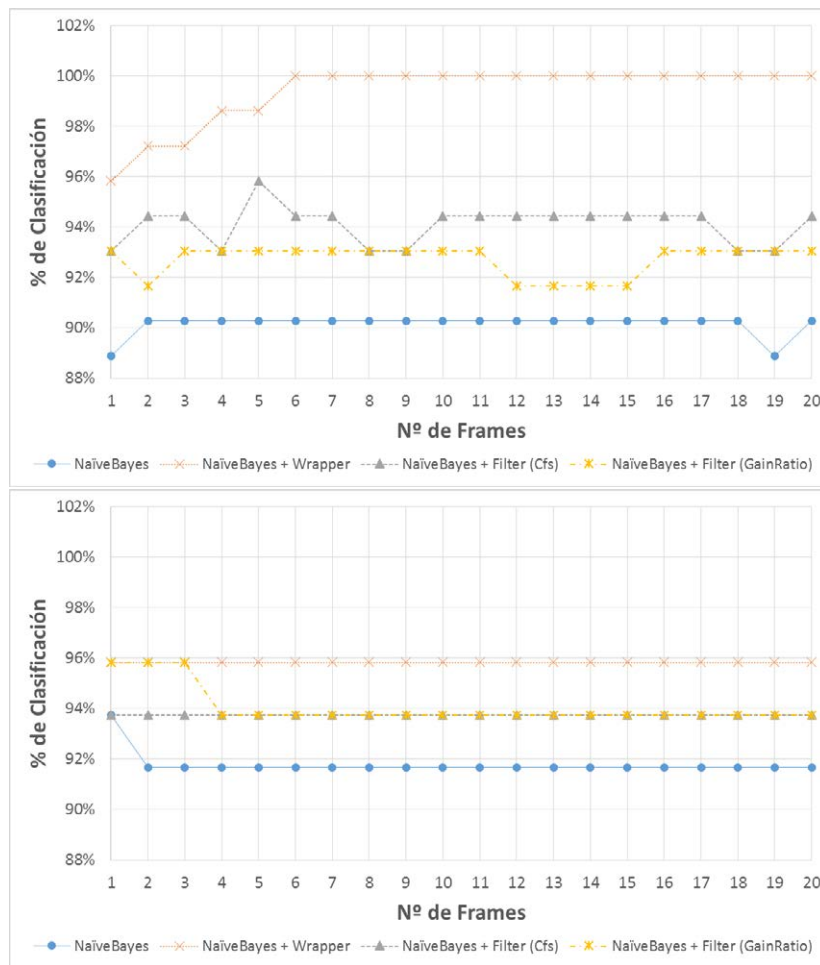


Figura 6.2: Precisión de los clasificadores para los Tipos P (arriba) y M (abajo).

6.3. Resultados

La precisión del clasificador para los cilindros tipo P se muestra en la Figura 6.2 (arriba), con y sin técnicas de selección de variables. Para los experimentos con el clasificador *naïve Bayes*, con un solo fotograma, el modelo es capaz de alcanzar cerca del 89% de precisión. Con dos fotogramas o más, el clasificador se estabiliza con una precisión del 90,2%. Esto significa que el clasificador podrá detectar superficies rectificadas y sin rectificar en 160 ms con una precisión del 90,2%.

Por otro lado, para los cilindros Tipo M que se muestran en la Figura 6.2 (abajo), el clasificador *naïve Bayes* solo alcanza el 93,8% para un fotograma, disminuyendo a 91,6% para dos fotogramas o más. Esta situación muestra un mejor comportamiento que los cilindros tipo P, obteniendo una mayor precisión en el mismo tiempo.

De esta manera, aplicando técnicas de FS, el rendimiento del clasificador aumenta para ambos tipos de piezas de prueba, excepto para el CFS utilizado en el Tipo M. En cuanto a los resultados para el Tipo P para el *naïve Bayes* junto con el filtro *information gain*, la precisión al usar un fotograma es del 93%, estabilizándose con más fotogramas, excepto para

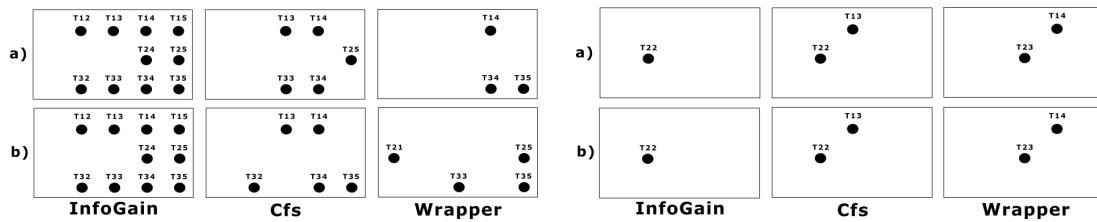


Figura 6.3: Temperaturas seleccionadas para a) un fotograma y b) dos fotogramas con cada método de FS. Tipo P (izquierda) y M (derecha).

dos y de 12 a 15 fotografías. El uso del *information gain* para el Tipo M hace aumentar la precisión hasta el 95,8% manteniéndose estable hasta tres fotografías.

Al utilizar CFS en el Tipo P, la precisión es similar a la obtenida con *information gain* para un fotograma. Sin embargo, hay una mejora hasta el 94,4% cuando se usan dos fotografías. En este caso, las variables seleccionadas con *information gain* se reducen a 5 en lugar de 10 para un fotograma y a 5 en lugar de 21 para dos fotografías.

Para las probetas Tipo M, con un fotograma se encuentra la misma precisión del clasificador aplicando y sin aplicar la selección de variables tipo CFS, sin embargo, permanece estable en un 93,75% de precisión, independientemente del número de fotografías dados. En consecuencia, el número de variables seleccionadas es dos, en comparación con uno utilizando *information gain* para uno y dos fotografías.

Por otro lado, al usar la técnica de *wrapper* en el Tipo P, la precisión es relativamente más alta que con las técnicas de filtrado, cercana al 96% para un fotograma y aumentando a 97% para dos fotografías. Es importante destacar que a partir de 6 fotografías, el clasificador puede alcanzar el 100% de precisión, lo que significa que a partir de esta cantidad, el problema es linealmente separable. En el Tipo M, la precisión es la misma que la obtenida con *information gain*, superando a otras técnicas de FS al usar tres fotografías.

La posición de las variables seleccionadas dentro de la matriz de pirómetros, para uno y dos fotografías se muestran en la Figura 6.3.

6.4. Discusión

6.4.1. Análisis de los resultados

Tal y como se ha explicado en la sección anterior, las técnicas de selección de variables ayudan a aumentar la precisión del clasificador en una media del 9% para el Tipo P y del 4% para el Tipo M. En general, el clasificador *naïve Bayes* con *wrapper* tiene el mejor comportamiento en cuanto a precisión para ambos escenarios. Comparando entre los diferentes modelos y la posición física dentro de la matriz de las variables seleccionadas, es posible encontrar más información sobre el comportamiento físico de la temperatura y su interacción con la superficie metálica. Esto es muy relevante de cara al desarrollo de un sistema de control con la precisión y rapidez apropiadas para ser aplicados a un proceso de tratamiento térmico por láser.

Analizando los resultados de la Figura 6.3, en donde se muestran los pirómetros virtuales dentro de la matriz de cuadros, las posiciones de medición de temperatura en el Tipo P son seleccionadas independientemente del número de fotogramas usados. De esta manera, las posiciones T14, T34 o T35 para Tipo P y T22 en Tipo M se entienden como zonas del proceso muy relevantes, al estar siempre seleccionadas.

Desde el punto de vista del proceso, en el escenario del Tipo P, los bordes de entrada (T14) y salida (T34 y T35) de la zona afectada por el láser son más sensibles a la variación de la rugosidad de la superficie, mostrando una variación de temperatura más clara, la cual es fácilmente detectable por el clasificador.

Por otro lado, para el Tipo M, el punto más sensible está cerca del centro de la matriz. Esta situación es interesante debido al hecho de que la única diferencia entre cada probeta es su tamaño, ya que los parámetros del proceso son exactamente los mismos para el Tipo P. Sin embargo, el número de posiciones de pirómetros virtuales seleccionados es significativamente menor.

6.4.2. Aplicabilidad de las técnicas de *clustering*

De acuerdo con lo comentado anteriormente, el tiempo de detección de las diferencias en los patrones de temperatura superficiales por los clasificadores, con una precisión adecuada, siempre se encuentra por debajo de los 160 *ms*. Sin embargo, este valor depende de la velocidad de la cámara termográfica, por lo que de manera paralela se comienza a trabajar con cámaras de alta velocidad, en concreto una cámara con una velocidad de 1000 *Hz* que permite capturar la temperatura con una resolución de 32×32 píxeles. La disposición de las diferentes cámaras se pueden ver en la Figura 6.4. El resultado completo de esta aproximación se puede encontrar en (Ogbechie et al., 2017; Larrañaga et al., 2019).

Es importante destacar que se espera que las variables seleccionadas sean las mejores descriptoras de las diferencias de temperatura entre las superficies rectificadas y no rectificadas. Asimismo, para aumentar la capacidad de procesamiento de datos del CPS y con ello poder analizar los fotogramas lo más rápido posible, es necesario reducir la cantidad de datos que se le proporcionan. Es por ello necesario que las mediciones de temperatura se realicen en zonas donde se encuentran los mejores descriptores. Además, es posible recopilar, transmitir y procesar solo los píxeles de interés dentro del fotograma reduciendo así las necesidades de potencia computacional.

Como parte de los trabajos descritos anteriormente, es aquí donde el *clustering* entra a jugar un papel fundamental, ya que precisamente esta reducción en la cantidad de los datos es algo que se puede lograr perfectamente con este tipo de técnicas. De esta manera, teniendo presente la espacialidad de los datos distribuidos en forma de píxeles sobre una superficie, los primeros trabajos de procesamiento de datos de estas imágenes de alta velocidad se llevan a cabo con un algoritmo de *clustering* jerárquico aglomerativo descrito en la Sección 4.1.

Este algoritmo se aplica a los diferentes fotogramas del proceso en donde se cuenta con 1024 píxeles con diferentes colores equivalentes a la temperatura del proceso. De esta manera, en lugar de plantear la matriz de pseudo-pirómetros mostrada en la Figura 6.1, se plantea

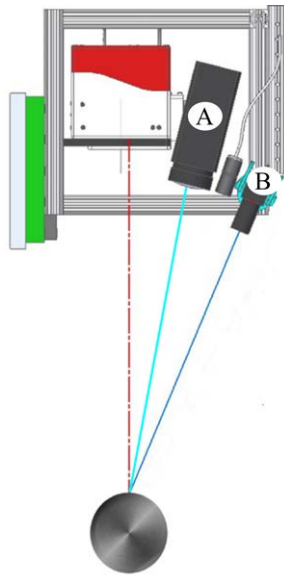


Figura 6.4: Disposición de cámaras dentro de la máquina de temple por láser: (A) cámara usada en este capítulo; (B) cámara de alta velocidad usada en trabajos posteriores.

buscar un número de regiones que pueda ser controlable desde el punto de vista de los datos, evitando en todo momento la pérdida de información, algo que sí sucedía en la matriz, al solo seleccionar áreas específicas sin ningún conocimiento del comportamiento de la temperatura.

De esta manera, utilizando el *clustering* jerárquico se obtienen 10 regiones, las cuales ha sido definidas tratando de mantener intactas las relaciones espaciales de los píxeles. Así, esta aplicación ha permitido apoyar la aplicación de otro tipo de técnicas de aprendizaje supervisado, reduciendo en gran medida el manejo de los píxeles y posibilitando el uso de sistemas CPS para el análisis del proceso. El resultado se puede ver en la Figura 6.5.

6.5. Conclusiones

La principal conclusión de este capítulo es que un CPS con capacidades de aprendizaje automático es un habilitador clave para la ayuda a la toma de decisiones. Gracias a estos dispositivos, estas decisiones se pueden implementar en el sistema de control, agregando capacidades adaptativas a un proceso de producción real. En este caso, los altos niveles de precisión obtenidos por los clasificadores, incluso sin aplicar técnicas de selección de variables, permiten detectar diferencias entre la rugosidad de las superficies, condición clave para los procesos fundamentalmente ópticos, como los tratamientos térmicos con láser.

Este trabajo ha proporcionado resultados para dos tipos de rugosidad de la superficie. Este es uno de los estados superficiales más difíciles de detectar por los operarios de máquina debido a las pequeñas diferencias visuales entre las temperaturas. En este caso, el modelo solo necesita información de al menos los dos primeros fotogramas de la termografía, que, como se ha explicado anteriormente, debería reducirse si la velocidad de la cámara termográfica fuera mayor. Esta situación abre nuevos espacios de aplicación los cuales permitirían integrar

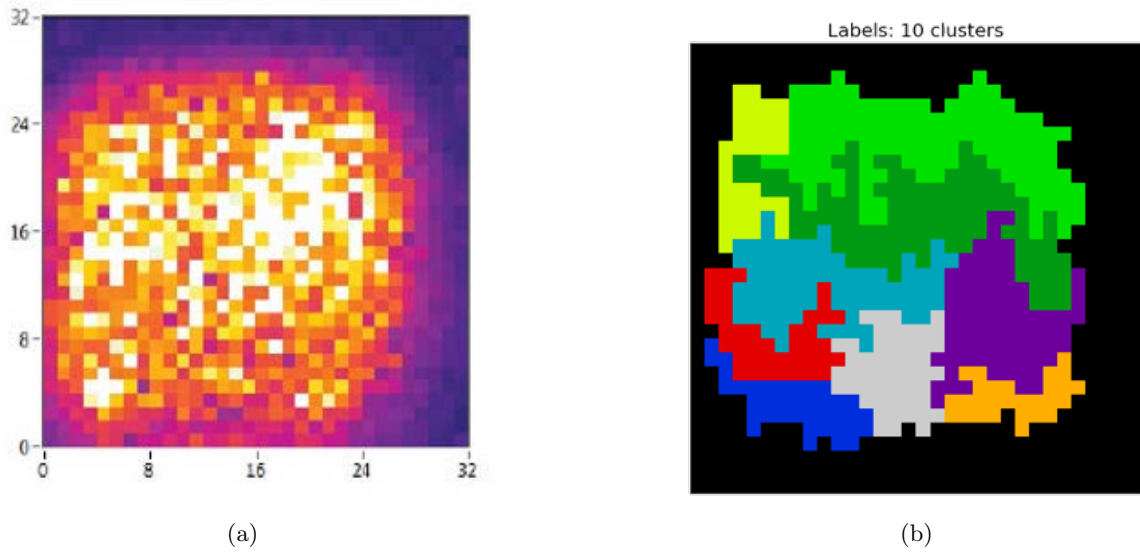


Figura 6.5: Aproximación al problema de pre-procesado de datos con técnicas de *clustering* con 10 zonas (a) Fotograma termografía de 32×32 píxeles, (b) resultado del *clustering* jerárquico aglomerativo.

modelos de aprendizaje automático para controlar el proceso de tratamiento superficial con láser.

Según los resultados obtenidos, es posible afirmar que el modelo de clasificador permite relacionar la reflectividad de la superficie (directamente relacionada con la rugosidad) y los valores de temperatura recopilados por termografía.

Por otro lado, la aplicación del *clustering* jerárquico sobre este problema, define un punto de partida para pensar en el *clustering* como una herramienta válida en varios niveles: pre-procesamiento y procesamiento en la industria, que será descrito en los siguientes capítulos.

Patrones de funcionamiento de motores industriales

De acuerdo con lo explicado en el Capítulo 6, se identifican también otro tipo de problemáticas que se podrían resolver en el mundo industrial. Esta vez, enfocándose en los elementos de máquina, tal y como se describe en la Figura 1.3 relacionada con los diferentes niveles de aplicación de la 4IR. En este caso, la aplicación se describe como la extracción de patrones de funcionamiento de los motores, primer paso para desarrollar la hipótesis de este trabajo definida en la Sección 1.5.

Para ilustrar esta conceptualización, el caso de estudio presentado en este capítulo está orientado a los servomotores que accionan los ejes de las máquinas-herramienta. Estos elementos sufren unos gradientes de aceleración muy elevados, debido a las necesidades de posicionamiento del sistema de control. Dependiendo de su uso, esta situación conduce a altos niveles de estrés que podrían producir una degradación prematura de los componentes internos. Por otro lado, los servomotores son máquinas rotativas, lo cual facilita la extrapolación de los resultados de este trabajo a otros componentes basados en la rotación, por ejemplo, husillos o bombas.

Este capítulo aparece publicado en el Capítulo 5 del libro [Larrañaga et al. \(2019\)](#).

7.1. Definición de los motores industriales como huella digital

Esta sección define las posibilidades de mejora existentes mediante el uso de huellas fundamentadas en los patrones que generan los datos de funcionamiento. Este tipo de huellas pueden ser identificadas con técnicas de *clustering*. Este proceso se describe a continuación.

7.1.1. Mejora de la fiabilidad mediante huellas digitales

Uno de los principales requisitos de un componente de máquina es poder soportar las condiciones de servicio. Normalmente, estas se definen, describen y cuantifican durante las primeras fases de diseño. La resistencia mecánica es una característica inherente que depende

de las propiedades (mecánicas, químicas, etc.) del material elegido. Por otro lado, las condiciones de servicio son diferentes cargas (tensión, compresión, corte, flexión, torsión) que afectan al rendimiento de los componentes en las máquinas (Shigley y Mischke, 1956).

Sin embargo, existen incertidumbres acerca de la resistencia de un material debido a que su composición química, forma de fabricación, anisotropía y las condiciones específicas del servicio pueden variar. Esto podría comprometer el rendimiento de los componentes de una máquina a lo largo de su tiempo de vida, afectando a su fiabilidad. Formalmente, la fiabilidad es la probabilidad de que un componente de máquina realice su función prevista durante un periodo de tiempo específico sin sufrir fallos.

La fiabilidad es un ejemplo de un modelo estadístico que genera un patrón de comportamiento para un componente (Shigley et al., 2004), describiendo el estado estable relacionado con el mismo. Sin embargo, los efectos sistemáticos y aleatorios se encuentran mezclados si la evidencia es recopilada mediante la medición. Por lo tanto, los modelos estadísticos separan dichos efectos y producen una clara aproximación del patrón, siendo equivalente a la fiabilidad del elemento.

Para modelar la fiabilidad de los componentes es necesario reunir evidencia de su comportamiento. Para ello, la manera tradicional es realizar pruebas destructivas, las cuales permiten medir la resistencia del material en condiciones de servicio. Sin embargo, estas pruebas se realizan bajo condiciones controladas en laboratorio, lo cual no permite estudiar las condiciones operativas específicas que pueden tener importantes efectos sobre el rendimiento. Por ejemplo, los servomotores se seleccionan de catálogos en función de las condiciones esperadas, generalmente obtenidas de la información recopilada durante la fase de diseño, en donde se definen las condiciones nominales y máximas.

En esta etapa es imposible estimar con precisión las condiciones reales de operación en circunstancias aleatorias. Por lo tanto, el servomotor puede tener que operar fuera de su zona de funcionamiento nominal descrita en los catálogos comerciales. De esta manera, si las condiciones de funcionamiento están fuera de los valores nominales durante largos períodos de tiempo, podría producirse una degradación prematura, reduciendo así la disponibilidad de la máquina.

Las pruebas destructivas son necesarias para modelar la fiabilidad de los componentes. Sin embargo, este proceso es costoso y, a veces, inviable (por ejemplo, plantas nucleares o equipos aeroespaciales). Por lo tanto, los ingenieros normalmente extrapolan los modelos de fiabilidad partiendo de aquellos pertenecientes a un grupo específico de componentes: con un mismo material y tolerancias de ensamblaje, definidos como un grupo de componentes idénticos. Con respecto al ejemplo de servomotor seleccionado, esto significa que los servomotores con la misma referencia de catálogo se definen como idénticos.

Usando el concepto de modelos basados en datos reales de operación, el cual se plantea bajo el paradigma de la 4IR y se describe en el Capítulo 1, es posible ajustar un modelo teórico utilizando datos reales de operación recolectados a partir de la sensórica de máquina. Lo que se obtendría es un modelo que tiene en cuenta el patrón de variación de un componente, midiendo los valores reales para los materiales en condiciones de operación. Este patrón será útil en la

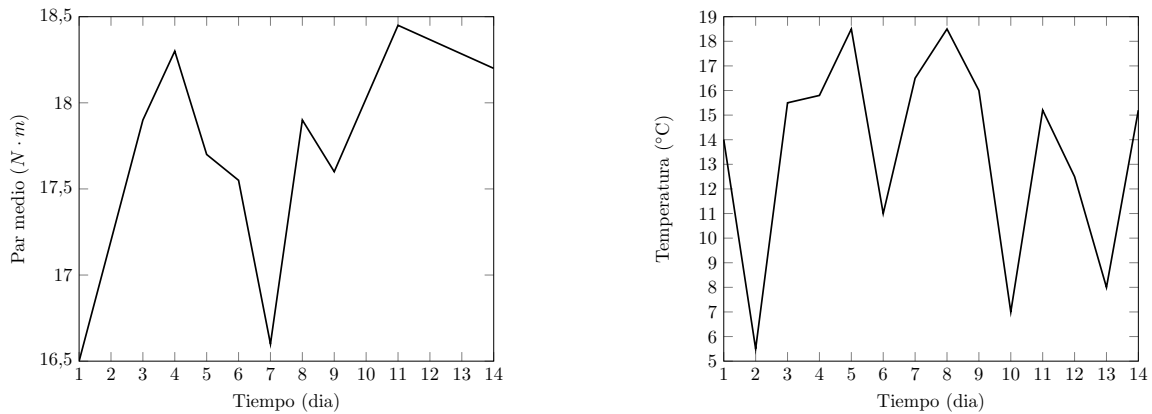


Figura 7.1: Valores para el servomotor medidos durante su operación con valor nominal de $27 N \cdot m$ a $100^{\circ}C$. (a) Par medio del servomotor. (b) Temperatura media del servomotor.

fase de diseño (para la selección del servomotor) o durante el control de calidad relacionado con la fabricación del elemento (rendimiento medio alrededor de un valor nominal).

Como ejemplo de una fase de diseño, se utiliza un servomotor para mover el eje de una máquina de fresado o fresadora. La Figura 7.1 muestra el par medio y la temperatura medidos durante la operación real de fresado para un eje responsable del movimiento transversal en X. Debido a que estos procesos utilizan el eje según las necesidades de mecanizado, las condiciones de operación no necesariamente estarán en su valor nominal. En este caso, el par medio y la temperatura siempre están muy por debajo del valor nominal, siendo este $27 N \cdot m$ y $100^{\circ}C$ respectivamente, lo que demuestra que un motor más pequeño y más barato sería una mejor opción. Por otro lado, la Figura 7.1 muestra cómo las condiciones de operación son estables en el tiempo. Esto hace que sea más difícil detectar un comportamiento específico útil que permita tomar una decisión de selección de un servomotor de menor coste que probablemente nunca experimente una degradación prematura.

Con respecto al control de calidad, es importante saber cómo de fiable es un componente antes de su instalación, es decir, la calidad del servomotor debe ser controlada antes del montaje de la máquina. Esto significa que no es factible obtener valores de rendimiento que estén ajustados al comportamiento de cada componente en condiciones de producción. Por lo tanto, es extremadamente útil tener un modelo como patrón de comportamiento que sirva de referencia para verificar eficientemente el rendimiento.

A lo largo de este capítulo el patrón de comportamiento se define como una huella digital del elemento, la cual permite comprobar su estado automáticamente según lo descrito por Lee et al. (2015). Es decir, se quiere permitir la comparación del rendimiento de un elemento activo con la huella digital del mismo. Este enfoque de huella digital se describe como un particionado de los diferentes estados por los que atraviesa el elemento en diferentes subconjuntos más pequeños, algo viable usando técnicas de *clustering*. Los clústers se definen de acuerdo a técnicas de particionado no supervisado que utilizan diferentes métricas para lograrlo. Diferentes algoritmos de *clustering* se encuentran resumidos en el Capítulo 4.

Es importante destacar que las tolerancias de fabricación aceptan desviaciones de rendi-

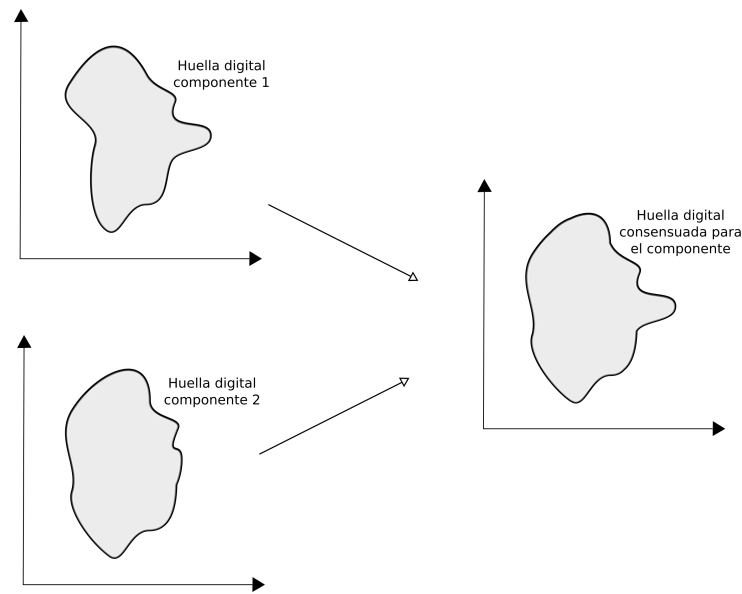


Figura 7.2: Generalización de la huella digital del componente.

miento normales debidas a la anisotropía de los componentes, es decir, tolerancias en torno al comportamiento nominal. Si la huella digital se creara utilizando datos de un solo componente estas anisotropías generarían una mayor tasa de rechazo, eliminando los componentes buenos que contengan estas pequeñas desviaciones. Por lo tanto, debe construirse una huella digital generalizada utilizando datos de muchos componentes con la misma referencia y aplicación, para tener en cuenta un mayor número de desviaciones de rendimiento normales.

Este proceso de generalización se denomina huella digital consensuada del componente, tal y como se refleja en la Figura 7.2. Por lo tanto, los diferentes patrones de funcionamiento del componente se pueden unir y crear un clúster, que consensuadamente tiene en cuenta la formas de funcionamiento del elemento.

7.1.2. *Clustering* aplicado a elementos del banco de pruebas

Como se ha mencionado anteriormente, esta aplicación se centra en la maquinaria rotativa, elementos muy comunes en la industria manufacturera. En concreto, se estudian los servomotores utilizados para posicionar los ejes de la máquina. Sin embargo, la metodología resultante se puede extrapolar fácilmente a otros sistemas de máquinas, como husillos o motores. Esto es posible porque el eje y los cojinetes son las partes mecánicas que más degradación sufren dentro de la maquinaria rotativa (Figura 7.3).

Uno de los problemas más comunes de los servomotores es su rendimiento cuando son nuevos. Es por ello que es importante conocer sus condiciones de partida para inferir si son lo suficientemente fiables para ser instalados. Por este motivo, como se describe en [Siddique et al. \(2005\)](#), el rendimiento de los componentes internos giratorios de los motores eléctricos es estudiado en condiciones de laboratorio para, de esta manera, proporcionar una imagen clara y sin interferencias contextuales de la integridad de los husillos de bolas, guías, soportes de

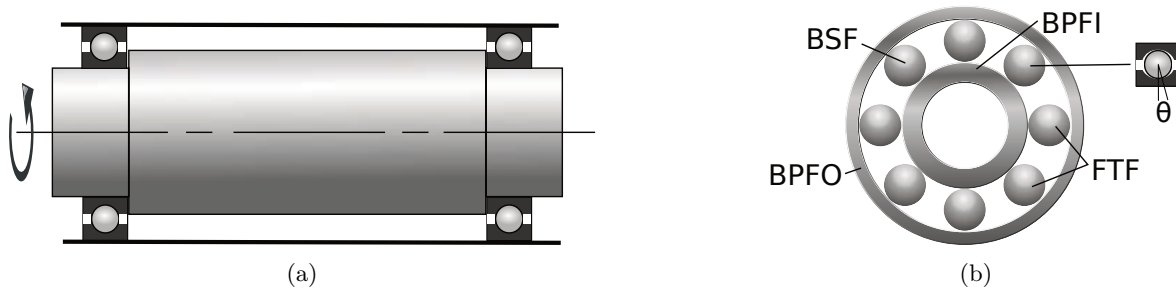


Figura 7.3: (a) Ensamblaje de rodamiento y eje. (b) Composición interna del rodamiento.

cables, etc.

Por lo general, las desviaciones en el comportamiento de un servomotor nuevo son causadas por los rodamientos internos. Los rodamientos son los componentes más débiles del servomotor y se seleccionan mediante catálogos que toman como referencia la operación nominal esperada. Sin embargo, dichos catálogos son una colección de resultados de comportamientos en pruebas destructivas realizadas en laboratorios. Desafortunadamente, los rodamientos pueden tener comportamientos no lineales que no están completamente modelados por pruebas de laboratorio y pueden ser muy perjudiciales para su vida útil.

De esta manera, la forma tradicional de mitigar estas incertidumbres producidas por comportamientos no lineales es mediante la aplicación de un factor de diseño (\bar{n}_d) durante la selección. El efecto principal de \bar{n}_d es alejarse de los valores límite, es decir, si el comportamiento del componente en condiciones específicas es bien conocido, entonces $\bar{n}_d = 1$. De lo contrario $\bar{n}_d > 1$. En general, este factor de seguridad se selecciona en función de las pruebas de laboratorio y del conocimiento basado en la experiencia. Los fabricantes de rodamientos tienen un amplio conocimiento para estar preparados, pero los factores externos causados por otras piezas dentro del ensamblaje pueden afectar a su rendimiento, haciendo que los elementos nuevos se comporten de forma errática o diferente a la esperada.

En este caso, se desarrollará una huella digital utilizando servomotores nuevos para, de esta manera, obtener un punto de referencia que permita comparar los servomotores antes de la instalación. Con el objetivo de aislar factores externos no deseados, las pruebas se realizarán en condiciones sin carga midiendo cuatro variables: velocidad angular del eje, consumo de energía, temperatura y vibración. Estas variables se utilizan normalmente para analizar el rendimiento de la maquinaria rotativa industrial, estando directamente relacionadas con los rodamientos y la resistencia del eje. Por lo tanto, sus valores pueden dar la siguiente información:

- Velocidad angular del eje, Ω , es decir, revoluciones del eje por minuto. La velocidad angular del eje es la variable controlada por el servomotor y la unidad de control numérico (NCU), y se utiliza como referencia para la prueba.
- El consumo de energía, P , es capaz de mostrar anomalías internas cuando existe una gran variabilidad en su valor en condiciones estables. Por ejemplo, si hay una parte

como un rodamiento defectuoso que interfiere con la rotación, el servomotor necesitará más potencia para mantener su velocidad de rotación.

- La temperatura, T , como el consumo de energía, captura anomalías internas. La temperatura se verá afectada si el comportamiento es fuera de lo común. Por ejemplo, un fallo en un rodamiento tiene un efecto negativo en su coeficiente de fricción, lo que lleva a un aumento de la energía disipada en forma de calor detectado por un termopar.
- La vibración es uno de los indicadores tradicionales que se utilizan para detectar degradaciones internas, ya que es capaz de mostrar comportamientos defectuosos de antemano. Sin embargo, la interpretación de la vibración no es directa, y se requiere más experiencia para comprender su comportamiento en un componente defectuoso.

De esta manera, se limita el estudio de las vibraciones solo a la relacionada con el eje y los rodamientos. Para este propósito, la variable de vibración se desglosa de la siguiente manera:

- Vibración relacionada con el eje: la frecuencia de giro del servomotor debe ser monitorizada para determinar el valor de la vibración del eje. Por ejemplo, si el servomotor está funcionando a 3000 RPM , la frecuencia de rotación (F_{shaft}) será de $3000 \text{ RPM} \times \frac{1 \text{ min}}{60 \text{ s}} = 50 \text{ Hz}$. Por lo tanto, el valor de vibración será el valor máximo o RMS a una frecuencia de 50 Hz (A_{shaft}). Un cambio en el valor de la amplitud de la vibración a esta frecuencia puede indicar anomalías relacionadas con el desequilibrio del eje o el pandeo del mismo.
- Vibración relacionada con el rodamiento: los valores de vibración de un rodamiento se desglosan de acuerdo con cuatro frecuencias de monitorización diferentes (Figura 7.3b), ya que es un componente mecánico ensamblado:

- Frecuencia fundamental del tren de bolas o FTF ,

$$FTF = \frac{1}{2} \frac{\Omega}{60} \left(1 - \frac{Bd}{Pd} \cos \theta \right), \quad (7.1)$$

- Frecuencia de paso de las bolas por el anillo interior o $BPFI$,

$$BPFI = \frac{Nb}{2} \frac{\Omega}{60} \left(1 + \frac{Bd}{Pd} \cos \theta \right), \quad (7.2)$$

- Frecuencia de paso de las bolas por el anillo exterior o $BPFO$,

$$BPFO = \frac{Nb}{2} \frac{\Omega}{60} \left(1 - \frac{Bd}{Pd} \cos \theta \right), \quad (7.3)$$

- Frecuencia de giro de las bolas o BSF ,

$$BSF = \frac{Pd}{2Bd} \frac{\Omega}{60} \left[1 - \left(\frac{Bd}{Pd} \right)^2 \cos \theta \right], \quad (7.4)$$

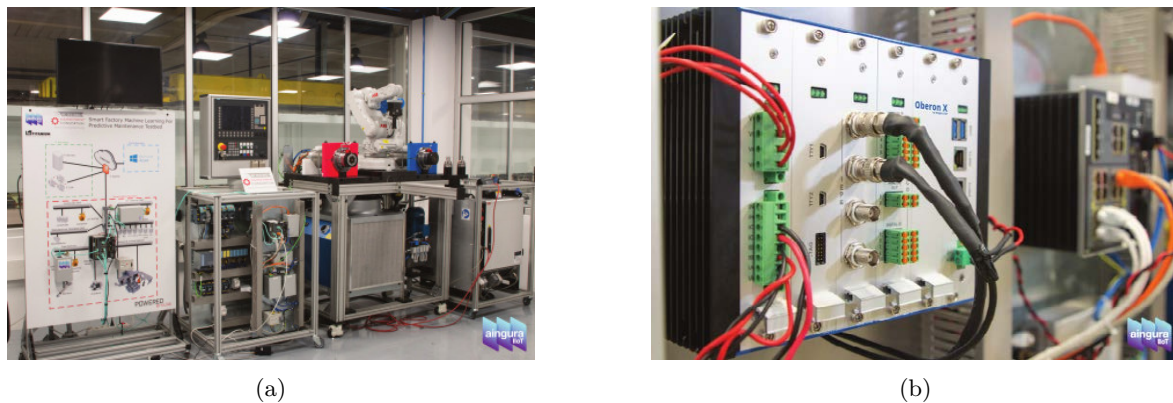


Figura 7.4: (a) Banco de pruebas del Industrial Internet Consortium. (b) Sistema de adquisición de datos usando un CPS.

donde Ω son las *RPM* del eje, Bd es el diámetro de la bola o rodillo, Nb es el número de bolas o rodillos, Pd es el diámetro de paso y θ es el ángulo de contacto de anillo exterior e interior con la bola como se ilustra en la Figura 7.3. Por lo tanto, los valores pico o RMS en estas frecuencias representan la vibración de cada parte del rodamiento, siendo posible durante el análisis de datos determinar qué parte del rodamiento no funciona correctamente.

Para demostrar el enfoque de la huella digital se ha utilizado un banco de pruebas de servomotor experimental. La Figura 7.4 muestra la infraestructura del banco de pruebas que permite reproducir el comportamiento real de una máquina y todos sus subsistemas desde la adquisición de datos hasta el análisis. Este banco de pruebas ha sido desarrollado dentro de las actividades de este trabajo de tesis y se ha convertido en el primero, abierto a diferentes empresas y relacionado directamente con aprendizaje automático para aplicaciones industriales dentro del IIC¹.

En este caso, se ha configurado el banco de pruebas para obtener variables sin carga en diferentes escenarios de velocidad de rotación. Los servomotores utilizados para las pruebas son tres SIEMENS 1FK7042-2AF71, con las características descritas en la Tabla 7.1. Estos servomotores están equipados con dos rodamientos del tipo 6204 con las características descritas en la Tabla 7.2.

Los acelerómetros utilizados que adquieren la señal de vibración del rodamiento y el eje tienen una sensibilidad nominal de 100 mV/g y un rango de frecuencia de $0,2 \text{ Hz}$ a 10000 Hz en $\pm 3 \text{ dB}$. Adicionalmente, los valores de potencia y temperatura se obtienen directamente de la NCU (SIEMENS SINUMERIK 840D). Esta NCU almacena valores en espacios de memoria variable donde son recopilados por el sistema de adquisición.

¹<http://www.iiconsortium.org/smart-factory-machine-learning.htm>

²Datos del catálogo SIEMENS SIMOTICS S-1FK7 para servomotores.

Tabla 7.1: Especificaciones de los servomotores²

Especificación	Valor
Velocidad nominal	3000 <i>RPM</i>
Par estático	3,0 <i>N · m</i>
Corriente máxima	2,2 <i>A</i>
Par nominal	2,6 <i>N · m</i>
Corriente nominal	2,0 <i>A</i>
Momento de inercia del rotor	2,9 <i>kg · m²</i>

Tabla 7.2: Especificaciones del rodamiento ref. 6204 con $Bd = 0,312 \text{ mm}$, $Nb = 8$ bolas, $Pd = 1,358 \text{ mm}$ y $\theta = 0^\circ$

Tipo de vibración	Valor (<i>Hz</i>)
<i>FTF</i>	0,39 <i>RPM/60</i>
<i>BPFI</i>	4,92 <i>RPM/60</i>
<i>BPFO</i>	3,08 <i>RPM/60</i>
<i>BSF</i>	2,06 <i>RPM/60</i>

7.1.3. Descripción del conjunto de datos extraído del banco de pruebas

Se ha programado un ciclo en la NCU del banco de pruebas específico que permite activar los servomotores en las mismas condiciones de operación con el fin de obtener un conjunto de datos con información suficiente sobre cada servomotor. Por lo tanto, se ha desarrollado un programa de código NC mostrado en la Tabla 7.3 que permite mover los servomotores a una posición específica y volver a su posición original a la velocidad máxima. Específicamente, tres servomotores se moverán a la posición 5000 *mm* a una velocidad de avance (F) de 83120 $\frac{\text{mm}}{\text{min}}$, equivalente a 2400 *RPM*, y luego regresarán a la posición 0 *mm*, reiniciando el ciclo. Este ciclo ilustra el comportamiento de cada servomotor operando a una velocidad cercana a la máxima durante una rotación en sentido horario y antihorario.

El conjunto de datos adquirido en el banco de pruebas se toma de la NCU utilizando el CPS que se muestra en la Figura 7.4b. El trabajo de este dispositivo es reunir los valores de velocidad de rotación, potencia y temperatura para enviarlos a una base de datos remota. Adicionalmente, el CPS adquiere señales de los acelerómetros ubicados en cada servomotor,

Tabla 7.3: Programa NC para los servomotores

NC-Code
INI
G01 X5000 Y5000 Z5000 F83120
G01 X0 Y0 Z0
GOTO INI
M30

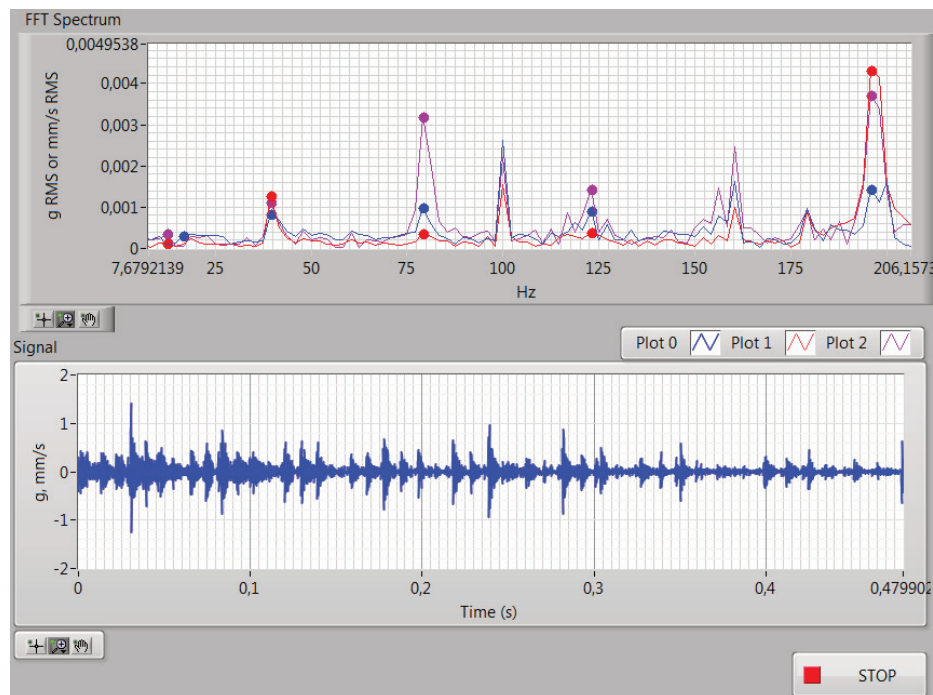


Figura 7.5: Visualización de la señal del acelerómetro: señal en el tiempo y sus componentes frecuenciales extraídas gracias a la FFT.

calculando internamente la transformada rápida de Fourier³ (FFT) de la señal. Después de haber definido las frecuencias de monitorización del eje y del rodamiento en términos de *RPM* del servomotor, el CPS usa la FFT para calcular la amplitud de aceleración de cada elemento (Figura 7.5). Estos valores también se envían a la base de datos remota y se sincronizan con los valores de NCU.

Dado que las variables son de diferente naturaleza el tiempo de adquisición se ha establecido en la variable más lenta, es decir, 480 ms , para así simplificar el conjunto de datos. Esta simplificación debería ayudar a evitar los pasos de pre-procesamiento antes de llevar a cabo el análisis de datos.

Para este estudio, se ha recopilado un conjunto de datos que cubre una operación de una semana, para así obtener un número representativo de instancias. Una vez recopilado, se ha reparado el conjunto de datos para eliminar espacios vacíos utilizando técnicas de imputación por *k*-NN, descritas en la Sección 3.1. El tamaño del conjunto de datos es de 1462585 instancias por un total de 39 variables, 13 por servomotor:

- Velocidad angular, Ω .
- Potencia, P .
- Temperatura, T .

³Una transformada rápida de Fourier es un algoritmo que muestrea una señal durante un período de tiempo (o espacio) y la divide en sus componentes frecuenciales

Tabla 7.4: Técnicas de *clustering* utilizadas

Método de <i>clustering</i>	Algoritmo
Jerárquico	Aglomerativo (Sección 4.1)
Particional	K -medias (Sección 4.2)
<i>clustering</i> espectral	Shi and Malik (SM) (Sección 4.3)
Propagación de la afinidad	Affinity Propagation (AP) (Sección 4.4)
Probabilístico	Gaussian mixture model (GMM) (Sección 4.5)

- Amplitud de vibración: A_{shaft} , A_{FTF} , A_{BPFI} , A_{BPFO} , A_{BSF} .
- Frecuencias de vibración: F_{shaft} , FTF , $BPFI$, $BPFO$, BSF .

7.2. Algoritmos de *clustering* para el desarrollo de huellas digitales

Los algoritmos descritos en el Capítulo 4 se utilizan para encontrar las huellas digitales de los servomotores, tal como se define en la Sección 7.1, en donde no hay una clase para predecir. En su lugar, las instancias se dividen en grupos naturales (Xu y Tian, 2015), que serán equivalentes al patrón de variación del servomotor. De esta manera, se aplican y comparan las cinco técnicas de *clustering* mostradas en la Tabla 7.4.

7.2.1. Detalles de la implementación

Utilizamos la librería `Scikit-Learn` para implementar cada uno de los algoritmos de *clustering* (Pedregosa et al., 2011). Ya que el propósito principal de este capítulo es mostrar cómo los algoritmos de *clustering* pueden encontrar la huella digital de elementos, la eficiencia del algoritmo no es una preocupación principal. Los experimentos se han realizado utilizando un subconjunto de 12000 instancias de datos extraídos al azar del conjunto de datos original y 36 variables, dejando de lado las variables de velocidad angular Ω de las variables enumeradas en la Sección 7.1.3 para los tres servomotores, debido a que es dicha velocidad angular constante a 2400 *RPM*.

Los parámetros utilizados en cada algoritmo se describen a continuación:

- *Clustering* jerárquico aglomerativo: el criterio de Ward de vinculación es seleccionado y la distancia euclídea es la métrica de distancia utilizada.
- K -medias: se emplean los parámetros predeterminados por la librería `Scikit-Learn` usando `K-medias++` para la inicialización del clúster y la distancia euclídea.
- *Clustering* espectral: se calcula la matriz de afinidad usando la función `kneighbors_graph`. Se utiliza la estrategia de descomposición de valores propios de Arpack, diseñada para resolver problemas de valores propios a gran escala de manera más eficiente.

- Propagación de afinidad: el valor de amortiguamiento (ξ) se ha establecido en 0,75. El valor de preferencia para este estudio de caso se ha establecido en cinco veces la similitud de entrada mínima para, de esta manera, encontrar un número adecuado de clústers. Las similitudes de entrada se han calculado previamente utilizando la función `euclidean_distances`.
- Modelo de mixtura de Gaussianas: el tipo de covarianza se ha configurado como completa, y para la inicialización se utiliza K -medias.

Para verificar el comportamiento de los algoritmos cuando cambia el número de clústers, los algoritmos se han ejecutado con el valor K establecido en 3, 5 y 7. Esta selección de valores de K no es aplicable para el algoritmo de propagación por afinidad, ya que éste se calcula de manera automática.

Para inspeccionar visualmente el comportamiento de los clústers de las 36 variables en un espacio 2-D y analizar el efecto de K , también se ha aplicado el escalado multidimensional (MDS) descrito en la Sección 3.3.

7.3. Resultados y discusión

Como se muestra en las Figuras 7.6 - 7.8, la representación 2-D de las 36 variables de los servomotores se muestra altamente concentrada en una región muy definida, casi sin valores atípicos. De manera intuitiva, este resultado es importante para la validación, ya que denota claramente el comportamiento de los nuevos servomotores de alta calidad, según los cuales se puede construir un patrón.

Las nubes de puntos que se muestran en las Figuras 7.6 - 7.8 se pueden identificar como la huella digital del servomotor, que representa cuánto de bueno es el rendimiento con respecto a cada una de sus variables. Las distancias entre puntos más grandes que el grupo o valores atípicos denotarán servomotores anómalos. El umbral de distancia se debe definir después de realizar pruebas suficientes con más servomotores de la misma referencia. Sin embargo, este enfoque está fuera del alcance de este capítulo.

Después de ejecutar los cinco algoritmos de *clustering*, se puede observar que las formas de clusterización son similares, independientemente del algoritmo. En particular, dentro de la región de datos más densos, hay tres clústers predominantes con una forma y distribución definidas que muestran tres comportamientos de servomotores diferentes.

Desde el punto de vista de la ingeniería estos tres comportamientos podrían estar directamente relacionados con los estados del servomotor durante la operación: inactivo, acelerando/desacelerando y velocidad constante. Por lo tanto, estos tres comportamientos se definen como estados de los servomotores.

Los algoritmos K -medias y aglomerativos muestran similitudes con algunos valores de K . El *clustering* espectral y el GMM cuentan con las posiciones de los centroides para $K = 3$ también similares. De esta manera, los centroides se concentran en el medio de los clústers.

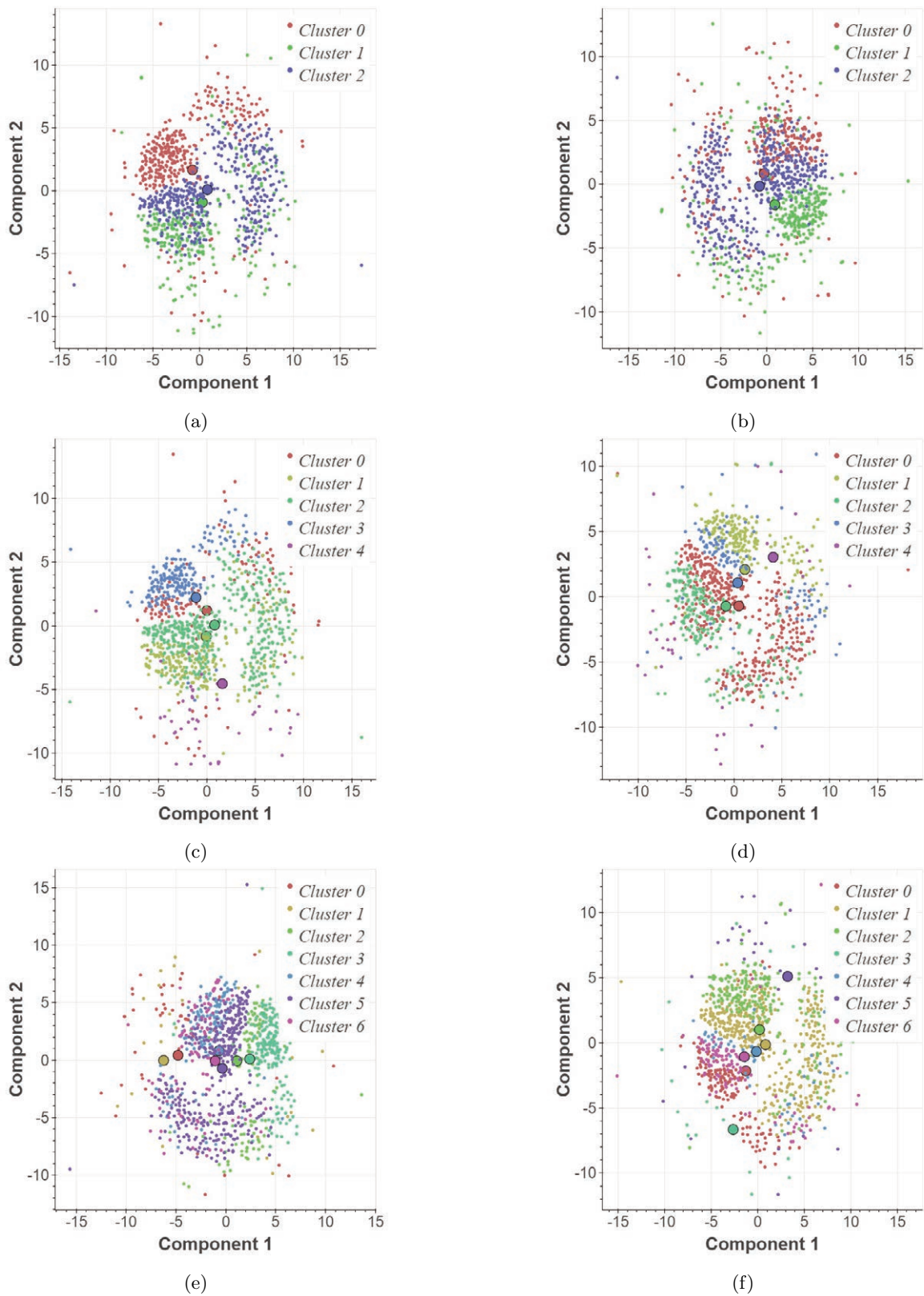


Figura 7.6: MDS jerárquico aglomerativo y el algoritmo K -medias con diferentes valores de K . (a) Aglomerativo con $K = 3$. (b) K -medias con $K = 3$. (c) Aglomerativo con $K = 5$. (d) K -medias con $K = 5$. (e) Aglomerativo con $K = 7$. (f) K -medias con $K = 7$.

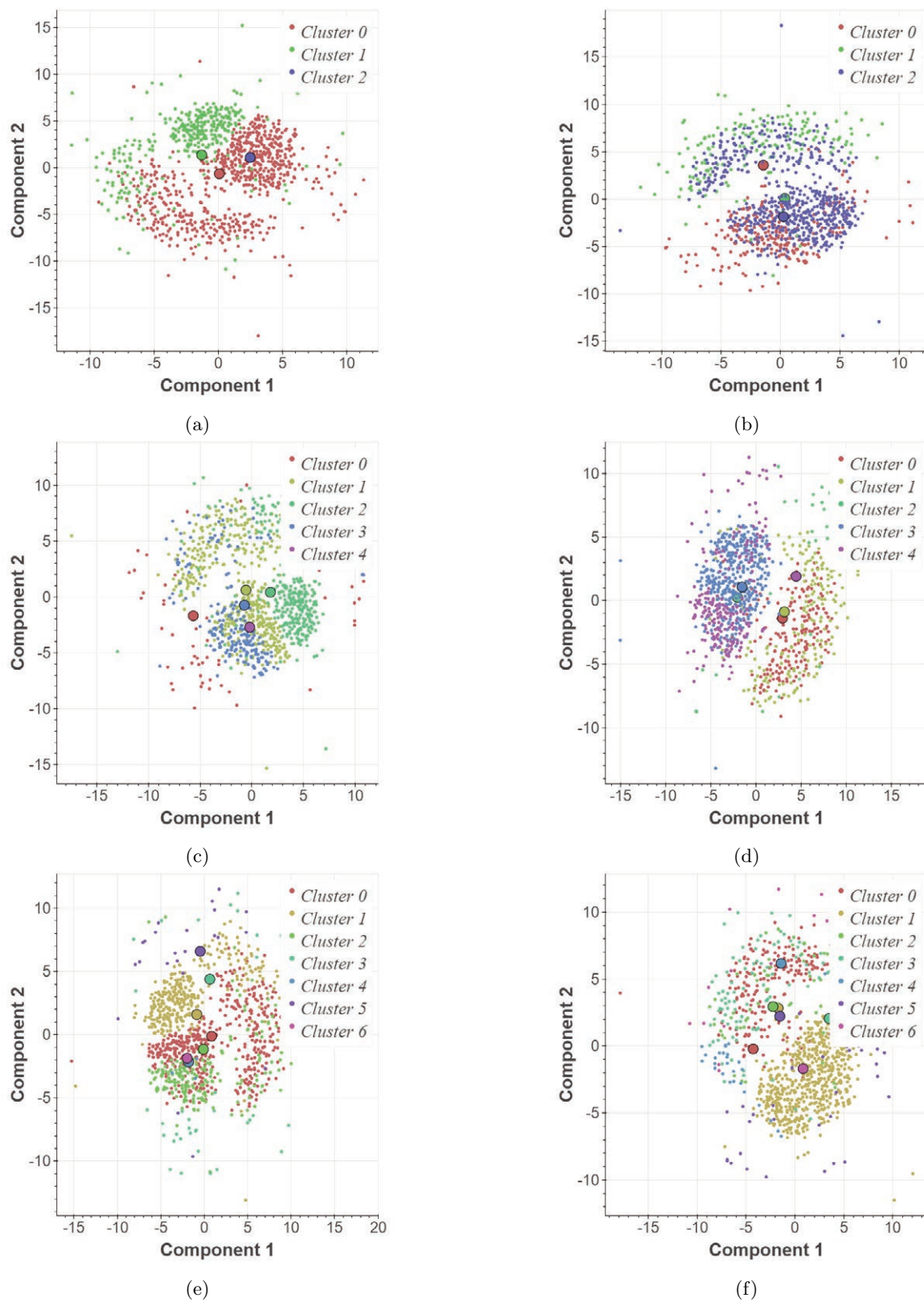


Figura 7.7: MDS para *clustering* espectral y GMM con diferentes valores de K . (a) *clustering* espectral con $K = 3$. (b) GMM con $K = 3$. (c) *clustering* espectral con $K = 5$. (d) GMM con $K = 5$. (e) *clustering* espectral con $K = 7$. (f) GMM con $K = 7$.

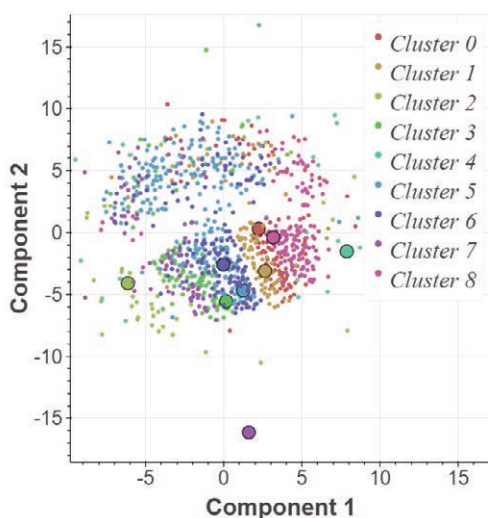


Figura 7.8: MDS para el algoritmo de propagación de afinidad.

Tabla 7.5: Centroides para el servomotor del eje X

Clúster	P (W)	T (°C)	$A_{FTF}(g)$	$A_{shaft}(g)$	$A_{BSF}(g)$	$A_{BPFO}(g)$	$A_{BPFI}(g)$
0	4,5	36,3	0,0003	0,0014	0,0009	0,0004	0,0005
1	23,3	35,6	0,0003	0,0014	0,0009	0,0005	0,0005
2	20,3	38,0	0,0004	0,0012	0,0009	0,0004	0,0005

Para el AP, el algoritmo detecta automáticamente nueve clústers diferentes utilizando los parámetros descritos en la Sección 7.2.1. Las formas y las posiciones de los centroides son similares. Sin embargo, este algoritmo es altamente sensible a los parámetros, donde pequeños cambios en los mismos pueden causar resultados radicalmente diferentes, especialmente cuando se modifica la preferencia.

Se selecciona $K = 3$ y el algoritmo jerárquico aglomerativo para ilustrar el análisis de los resultados del *clustering* con el fin de estudiar el comportamiento de cada servomotor. De esta manera, la potencia y la vibración del eje han sido las variables seleccionadas para este propósito. Estas variables tienen la capacidad de proporcionar información interesante sobre el rendimiento del motor. Sin embargo, otras combinaciones podrían seleccionarse según las necesidades.

Los resultados se muestran en la Figura 7.9, donde los clusters rojo (Cluster 0), verde (Cluster 1) y azul (Cluster 2) representan diferentes niveles de potencia, validando los tres clústers detectados utilizando el MDS para los servomotores.

Con el ánimo de realizar un análisis más detallado, las coordenadas de los centroides se muestran en las Tablas 7.5 - 7.7. De esta manera, analizando cada uno de los tres grupos se puede extraer lo siguiente:

- Clúster 0: representa niveles bajos de potencia, que van de 0 a 10 W con un centroide entre 4,4 y 5,4 W. Esto significa que este grupo está relacionado con la clase inactiva,

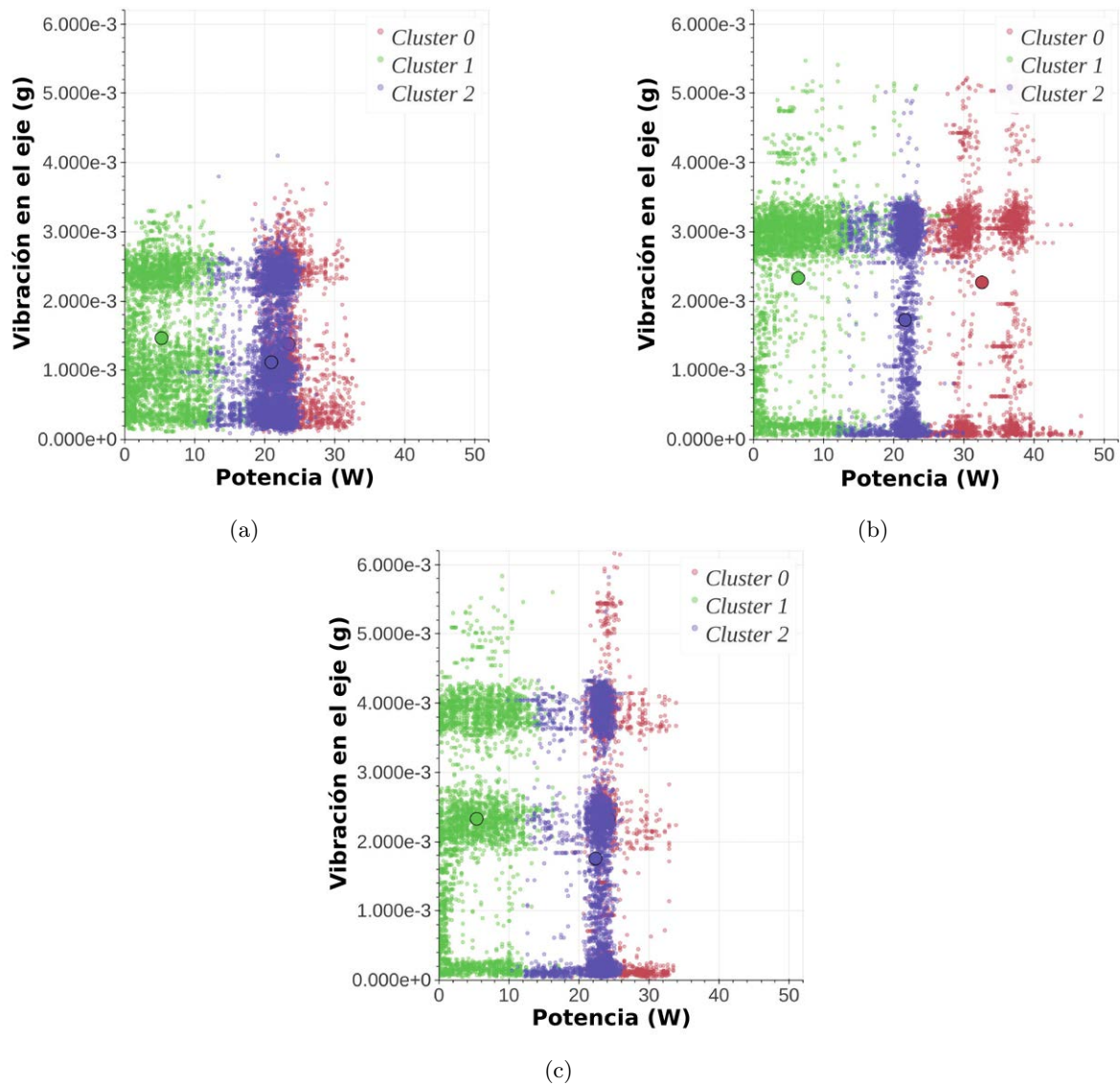


Figura 7.9: Potencia vs. vibración en el eje. (a) Servomotor eje X. (b) Servomotor eje Y. (c) Servomotor eje Z.

Tabla 7.6: Centroides para el servomotor del eje Y

Cluster	P (W)	T ($^{\circ}$ C)	$A_{FTF}(g)$	$A_{shaft}(g)$	$A_{BSF}(g)$	$A_{BPFO}(g)$	$A_{BPMFI}(g)$
0	5,41	36,6	0,0002	0,0022	0,0052	0,0004	0,0030
1	32,8	38,6	0,0002	0,0023	0,0050	0,0004	0,0029
2	21,1	36,9	0,0002	0,0018	0,0045	0,0003	0,0026

Tabla 7.7: Centroides para el servomotor del eje Z

Cluster	P (W)	T ($^{\circ}C$)	$A_{FTF}(g)$	$A_{shaft}(g)$	$A_{BSF}(g)$	$A_{BPFO}(g)$	$A_{BPFI}(g)$
0	4,6	33,4	0,0002	0,0022	0,0058	0,0008	0,0023
1	24,1	32,8	0,0002	0,0023	0,0057	0,0008	0,0022
2	21,6	34,7	0,0002	0,0018	0,0050	0,0007	0,0020

donde los motores se detienen y el consumo de energía es mínimo. La potencia no es 0 W como se esperaría, ya que los servomotores necesitan consumir energía para permanecer en la misma posición. Los valores de vibración son más elevados para el servomotor del eje X que para los otros servomotores. La vibración de este eje es mayor a la encontrada en los servos Y y Z con centroides en 22 mg.

- Clúster 1: representa altos niveles de potencia, siendo muy representativo para el servomotor del eje Y. Por lo tanto, este grupo está relacionado con la clase de velocidad constante, ya que necesita la máxima potencia para realizar la operación. El punto de partida es cerca de 25 W. El máximo valor para el servomotor del eje Y es 48 W, lo que significa que este necesita más potencia para realizar la misma operación. Los centroides para la variable potencia son completamente diferentes: 33 W para el eje Y y 23 - 24 W para los servomotores de los ejes X y Z respectivamente. Sin embargo, la vibración del eje es misma la misma para los servos Y y Z.
- Clúster 2: las instancias rondan los 20 W, momento en el cual este clúster es claramente distinto. De 10 a 15 W, los valores se confunden con el Clúster 0. Este clúster está relacionado con la clase de aceleración/desaceleración, teniendo potencia de rango medio con algunos picos esporádicos que son necesarios para cambiar del estado inactivo y superar la inercia de las partes internas del servomotor. Los centroides son similares, aunque existen diferencias en los valores de vibración para el servomotor del eje X. La vibración es escasa cuando la potencia es de alrededor de 20 W.

Es importante resaltar que en lo referente a cada uno de los componentes del rodamiento para los 3 clústers, se puede concluir que se están comportando de manera similar.

De esta manera, los clústers y sus centroides otorgan nueva información sobre el sistema que se está analizando, es decir, estos clústers definen la huella digital del servomotor en cada estado específico: inactivo, velocidad constante y aceleración/desaceleración. Por otro lado, los centroides indican los valores de operación medios dentro de cada clúster. Al estudiar las diferencias en términos de forma y/o posición del centroide se podría encontrar una anomalía o desviación que requiere inspección para encontrar la causa raíz.

Por ejemplo, realizando una comparativa entre las huellas de cada servomotor es posible concluir que el servomotor del eje X tiene menos niveles de vibración y consume menos energía que los demás. La vibración puede proporcionar información sobre el desequilibrio o la deformación del eje. Sin embargo, un aumento en el consumo de energía en la parte superior de la vibración del eje podría ser un signo de un desalineamiento entre el rodamiento y el

eje para los servomotores de los ejes Y y Z. Tal nivel de desalineamiento no es crítico en este caso, ya que los servomotores tienen el rendimiento nominal requerido. Sin embargo, esta diferencia podría ser crítica cuando el servomotor alcance el 80 % de su vida útil, ya que este podría evolucionar hacia una degradación prematura.

7.4. Conclusiones

Como se describe en los resultados experimentales de este capítulo, las técnicas de *clustering* son útiles para detectar patrones de comportamiento o huellas a nivel de componente. En este caso, se han analizado tres servomotores nuevos con la misma referencia, lo que demuestra que, aun siendo teóricamente idénticos, existen diferencias importantes entre los servos, las cuales aumentarán durante la operación debido a la degradación.

Encontrar patrones que puedan ayudar a desarrollar una huella digital general del elemento podría ser útil para evaluar el estado del componente antes de la instalación. En este caso, se han estudiado servomotores, pero este procedimiento podría aplicarse para analizar otras partes más críticas, por ejemplo, electromandrilos, bombas y muchos otros componentes como se mostrará en el Capítulo 8.

En este capítulo se ha encontrado que el algoritmo jerárquico aglomerativo ha sido el algoritmo más eficiente, proporcionando resultados más interpretables desde el punto de vista de la ingeniería. Sin embargo, hay otras opciones interesantes, como *K*-medias (el algoritmo más rápido en términos de tiempo de procesamiento) o GMM, que posiblemente podrían dar resultados más sólidos partiendo de la naturaleza Gaussiana de este tipo de elementos.

Los algoritmos de *clustering* son muy aplicables en este tipo de análisis cuando se requiere encontrar un patrón de comportamiento como punto de referencia. Además, no son difíciles de implementar. Sin embargo, requieren pruebas exhaustivas de antemano con el fin de ajustar el algoritmo, lo cual es complejo ya que algunos son sensibles a los cambios en los parámetros, por lo que es importante obtener un conocimiento profundo de su función.

Es conveniente resaltar que los algoritmos de *clustering* necesitan parámetros de configuración predefinidos, por ejemplo, número de clústers. En este caso, el conocimiento en relación al comportamiento de los componentes (por ejemplo, inactivo, aceleración/desaceleración, velocidad constante) debe conocerse con antelación, para lo que es necesaria la experiencia. Por lo tanto, las capacidades exploratorias de algoritmos de *clustering*, como el resto de técnicas de aprendizaje no supervisado, deben complementarse con opiniones de expertos para garantizar una correcta interpretación de los resultados.

Aplicación de *clustering* para detectar condiciones de trabajo en el proceso de mecanizado de alto rendimiento

Partiendo de la experiencia adquirida en el Capítulo 7, se plantea aplicar los algoritmos de *clustering* que mejor rendimiento han mostrado con el ánimo de extraer más información de uno de los elementos más críticos dentro de un sistema industrial de mecanizado de alto rendimiento en la industria de la automoción. Este elemento se llama electromandrino. La monitorización de este elemento es clave, ya que su avería podría ocasionar enormes costes a la producción, siendo el que mas influye en la disponibilidad de máquina. Es decir, el fallo del electromandrino puede parar la máquina, lo que generaría un efecto dominó parando el total de la línea productiva.

Por todo esto, en el presente capítulo se estudia la capacidad de los algoritmos de *clustering* para extraer esa información que permita a las personas de mantenimiento tener un conocimiento más profundo del elemento para así evitar sus fallos prematuros, incrementando de esta manera la disponibilidad de máquina. A su vez, se estudia la viabilidad de estos algoritmos para formar parte fundamental en las capacidades analíticas de un CPS.

Este capítulo aparece publicado en [Diaz-Rozo et al. \(2017\)](#)

8.1. *Clustering* en mecanizado

Tal y como se ha comentado en capítulos anteriores, los modelos de rendimiento en el sector de la máquina-herramienta se desarrollan principalmente en condiciones controladas, ya que los datos no están disponibles durante la operación real. Los datos son escasos porque actualmente existen sistemas robustos de adquisición de datos y, en otros casos, por razones de confidencialidad. En consecuencia, los fabricantes de máquinas-herramienta solo tienen

acceso a la información sobre sus máquinas cuando se produce un fallo, lo que les impide recuperar información previa que permita hacer un mejor diagnóstico del mismo.

Como se ha explicado en la Sección 1.1.1, el CPS se ha desarrollado bajo el paraguas del IIoT con el ánimo de integrar capacidades computacionales y físicas para proporcionar información útil para los operarios. Esta integración está basada en el hecho que los datos producidos por una máquina son útiles para comprender y ajustar los modelos teóricos (por ejemplo, análisis de elementos finitos). Adicionalmente, esto proporciona a los diseñadores o al personal de mantenimiento información sobre los fallos.

A nivel de máquina, un CPS permite el uso de datos provenientes de diferentes componentes con el ánimo de adquirir información útil. Sin embargo, los principales desarrollos dentro de los sistemas basados en datos se encuentran orientados a enfoques de aprendizaje supervisado. Por ejemplo, la predicción del RUL (variable clase):

- de un rodamiento de bolas usando como variables predictoras las mediciones de energía de banda mediante técnicas como filtros de Kalman (Wang et al., 2016),
- de herramientas de mecanizado usando variables del proceso (potencia, par, temperatura, etc.) como predictoras aplicando redes Bayesianas dinámicas (Tobon-Mejia et al., 2012), o
- de elementos periféricos, como puertas de ascensores, con datos de utilización (horas, ciclos, etc) como variables predictoras usando modelos de regresión logística (Yan et al., 2004).

Con respecto al aprendizaje no supervisado, las aplicaciones más comunes están en el campo de la logística, la cadena de suministro y la producción (Choudhary et al., 2008). Por ejemplo:

- Detección de factores de disminución del rendimiento en la producción basados en mapas auto-organizados y reglas de inducción (Gardner y Bieker, 2000).
- Mejora del rendimiento en la fabricación de semiconductores utilizando K -medias (Chien et al., 2007).
- Mejora del proceso catalítico en refinerías usando una variación del K -medias (Sebzalli y Wang, 2001).
- Identificación de los fallos de soldadura usando *clustering* difuso (Liao et al., 1999).
- Extraer conocimiento útil de los datos de diseño aplicando un algoritmo de *clustering* basado en K -medias (Jin y Ishino, 2006).

Este capítulo propone el desarrollo de una huella digital (Capítulo 7) o un patrón multidimensional de comportamiento de componentes utilizando técnicas de aprendizaje automático,



Figura 8.1: (a) Máquina para fabricación de cigüeñales usada en los experimentos. (b) Electromandrino instalado en la máquina.

específicamente, técnicas de *clustering*. Como se ha explicado en capítulos anteriores, los algoritmos de *clustering* pretenden dividir un conjunto de datos en clústers, interpretados como patrones de comportamiento (modos de operación o clases). Algunas propiedades de estos grupos, como la media, pueden dar una idea para su interpretación. Con el ánimo de ilustrar este enfoque, este capítulo trata sobre la aplicación de un dispositivo CPS para obtener el conocimiento del proceso sobre el electromandrino de la máquina durante su operación.

8.2. Metodología

Uno de los principales objetivos de este capítulo es mostrar cómo las tecnologías de IIoT han permitido el uso de datos reales para los sistemas de diagnóstico. Por lo tanto, los datos para este experimento se han tomado de una máquina-herramienta en producción real, similar a la que se muestra en la Figura 8.1. Esta máquina produce uno de los componentes del motor de combustión más importantes dentro de las plantas de fabricación de automóviles en todo el mundo, llamado cigüeñal. Esta máquina cuenta con dos electromandrilos para procesar dos componentes cada 120 s, con unos requisitos de calidad de menos de una parte producida defectuosa por millón. Específicamente, esta máquina está integrada en una línea de producción y está equipada con dos ejes que realizan la misma operación para procesar dos componentes y lograr el tiempo de ciclo de producción requerido por la línea de 60 s.

Debido a los requisitos de producción del fabricante de automóviles, la disponibilidad de la máquina debe ser alta, normalmente superior al 95 %. En este nivel, la degradación de los componentes de la máquina debe controlarse estrechamente para minimizar los fallos inesperados que pueden afectar la productividad general de la línea. Sobre esta base se requieren procedimientos de mantenimiento específicos para cuidar las partes críticas de la máquina como los electromandrilos (Figura 8.1b) y así evitar fallos inesperados durante la operación de mecanizado. Esto se conoce como mantenimiento preventivo, que en algunos casos puede resultar altamente costoso, ya que los componentes se reemplazan después de un tiempo específico de funcionamiento, independientemente de su condición.



Figura 8.2: (a) Integración del CPS en máquina con el detalle dentro del armario eléctrico. (b) CPS.

La función del electromandrino es entregar la potencia para que la herramienta pueda eliminar material de la pieza, en este caso, acero especial para motores de combustión. De esta manera, la potencia nominal de los electromandrilos equipados dentro de esta máquina es de 10 kW , sometiendo las partes mecánicas y eléctricas, como rodamientos y devanados eléctricos del motor, a altos niveles de estrés. Según la propia experiencia, el coste por el fallo de un electromandrino en este sector de la automoción puede ascender a 30000 dólares americanos para un repuesto de electromandrino y más de 40 horas de trabajo de mantenimiento para reemplazar el electromandrino. Como la línea de producción debe detenerse por completo, cerca de 200 operadores directos no pueden trabajar, lo que aumenta el impacto económico en la empresa. Esto, dependiendo de la planta productiva, podría estar cercano a un ratio de 50000 dólares americanos por hora.

Debido a las razones mencionadas anteriormente, el desarrollo de una herramienta más precisa para el diagnóstico de electromandrilos es un objetivo razonable. Por lo tanto, se ha desarrollado un CPS con electrónica embebida capaz de recopilar datos de ambos electromandrilos durante el proceso de mecanizado, para luego analizarlos mediante algoritmos de *clustering* que permitan obtener información relevante sobre los mismos.

8.2.1. Adquisición y pre-procesamiento de datos del electromandrino

La configuración experimental que se muestra en la Figura 8.2 se propone como implementación del CPS para recopilar datos de máquina. En este caso, los datos provenientes de la NCU a través de una interfaz Ethernet son recopilados por el CPS gracias a sus capacidades de comunicación bidireccional, pudiendo solicitar el estado de la máquina y registrar los datos. Una vez se han recopilado los datos y se les ha asignado una marca de tiempo la instancia se envía al dispositivo de disco duro local (HDD) para que se almacene en un formato CSV. La Figura 8.2a muestra la disposición física de la configuración y su integración en el armario eléctrico de la máquina.

El dispositivo que se muestra en la Figura 8.2b, es una primera versión del CPS diferente a la mostrada en la Figura 7.4b, el cual tiene la capacidad de establecer una comunicación con el sistema de control de la máquina-herramienta, es decir, con la NCU. Este CPS de grado industrial es un sistema SoC-e CPPS-Gate40 equipado con una FPGA Zynq® 7000-series™

y diferentes puertos de comunicación para obtener datos en tiempo real de la máquina. La FPGA puede proporcionar algunas funciones de pre-procesamiento, como filtrar o convertir señales al dominio de la frecuencia mediante una FFT.

La máquina está controlada por una NCU Sinumerik 840D de SIEMENS que utiliza el protocolo Step7 para enviar las variables internas a través de un puerto Ethernet al CPS. El procedimiento aplicable es almacenar las variables requeridas en una base de datos interna y enviar el paquete completo una vez que se tengan los valores de todas las variables. Esta técnica garantiza una comunicación de datos extremadamente rápida sin el uso de la memoria del sistema de control.

Dado que el objetivo en este capítulo es analizar los electromandrilos durante un proceso de mecanizado, el CPS necesita diferenciar cuándo la máquina realiza un ciclo con una pieza de trabajo. De esta manera, es posible pre-procesar los datos y descartar los datos innecesarios que podrían entorpecer el análisis. De acuerdo con esto, el CPS puede consultar variables en tiempo real para sincronizar los diferentes sistemas involucrados en la adquisición de datos. Una vez que el sistema de adquisición se ha integrado, el CPS está programado para recopilar las siguientes variables del CNC de la máquina: potencia en kilovatios (kW), velocidad angular en revoluciones por minuto (RPM), par en Newton metro ($N \cdot m$) y temperatura en grados Celsius ($^{\circ}C$). Estas variables de proceso son importantes para describir y diagnosticar el funcionamiento de motores eléctricos como se describe en [Nandi et al. \(2005\)](#). Además, se toman otras variables de la NCU, como se ha explicado anteriormente: marca de tiempo, estado de ciclo en vacío, estados que hacen referencia a cuando el proceso de mecanizado se está ejecutando sobre una pieza (no está trabajando en vacío). Estas variables se utilizan solo para fines de sincronización y pre-procesamiento.

En este caso, la máquina utilizada en el análisis puede procesar dos piezas al mismo tiempo para garantizar el tiempo de ciclo requerido (alrededor de 60 s por pieza). Para hacer esto, está equipada con dos ejes que realizan exactamente la misma operación de mecanizado en dos piezas, ya que toma alrededor de 120 s en completar el proceso de mecanizado para ambas piezas. El hecho de que ambos cabezales realicen exactamente la misma operación proporciona un marco de comparación perfecto. Para esta máquina, las variables se muestrean a 10 Hz para obtener transiciones durante 10 ciclos de mecanizado. Se producen 20 piezas durante los 10 ciclos de mecanizado, 10 piezas por cada electromandrino.

8.2.2. Desarrollo de patrones de comportamiento basados en el aprendizaje automático

Para encontrar los patrones de comportamiento del electromandrino es necesario utilizar técnicas de aprendizaje no supervisado debido a la falta de la variable clase que se pueda pre-decir. Los algoritmos de *clustering* que se van a usar son: K -medias (Sección 4.2), jerárquico aglomerativo (Sección 4.1), y GMM (Sección 4.5).

Con fines comparativos, se utiliza la implementación basada en Python de los anteriores algoritmos disponible en la librería de `Scikit-Learn` ([Pedregosa et al., 2011](#)). Los detalles

completos de la implementación están disponibles en la página web de `Scikit-Learn`¹.

De acuerdo con el objetivo de comparar estos algoritmos, es importante resaltar que las métricas existentes para esto, como la puntuación de homogeneidad, la puntuación de integridad, la medida V, el índice de Rand ajustado, la información mutua ajustada y el coeficiente de silueta, se pueden utilizar para evaluar el rendimiento de los algoritmos *clustering*. Sin embargo, tales métricas no son lo suficientemente fiables (Milligan y Cooper, 1985). Por lo tanto, los resultados se evaluarán siguiendo la opinión de un experto en mecanizado. Es importante tener en cuenta que los resultados son sensibles al método de *clustering* utilizado. Por ello, la evaluación de los expertos se basará en la contribución al nuevo conocimiento sobre el comportamiento del electromandrino.

8.3. Resultados de experimentación

Como se describe en la Sección 8.2, se han registrado diez ciclos de mecanizado con un tiempo de muestreo de 10 Hz, lo cual ha generado un total de 91297 instancias para analizar mediante *clustering*. Cada instancia contiene datos de potencia, par y temperatura para cada electromandrino. Por lo tanto, los datos para cada electromandrino se introducen en el algoritmo de *clustering* y luego las instancias son analizadas en términos de su asignación a cada clúster. Los resultados se presentan a continuación.

8.3.1. K-medias

El *K*-medias tiene tres parámetros principales: (a) número de clústers *K*, (b) tipo de inicialización y (c) distancia. Con respecto al número de clústers *K*, el valor seleccionado después de probar varios valores de *K* se basa en los estados operativos del cabezal: inactivo, aceleración/desaceleración y potencia total, es decir, $K = 3$. Con respecto a la inicialización *K*-medias++ es la opción seleccionada para este trabajo. Por otro lado, se utilizará la distancia euclídea para encontrar clústers esféricos de datos. Para el resto de parámetros utilizados se han dejado los valores predeterminados de la biblioteca `Scikit-Learn` donde están disponibles para consulta.

La Figura 8.3 muestra el par del electromandrino frente a la potencia. En este caso, el Clúster 0 está integrado por instancias con pares negativos de -33 a -8 $N \cdot m$ y con rango de potencia de -2 a 10 kW . El Clúster 1 está compuesto por instancias que tienen principalmente un par entre -8 y 8 $N \cdot m$ y un rango de potencia estrecho, de -1 a 3 kW . El Clúster 2 tiene instancias con un rango de par positivo entre 9 y 33 $N \cdot m$ con una potencia entre -10 y 2 kW . Las formas y los tamaños de cada grupo son similares para ambos ejes en estas dos dimensiones, con casi la misma cantidad de instancias agrupadas.

La relación entre temperatura y potencia se muestra en la Figura 8.4. En este caso, el Clúster 0 y el Clúster 2 tienen un rango de temperatura de 35,5 a 36,5°C. Para el Clúster 1, este rango de temperatura es de 33,8 a 36,5°C. En estas dimensiones, los tamaños y formas

¹<http://scikit-learn.org/>

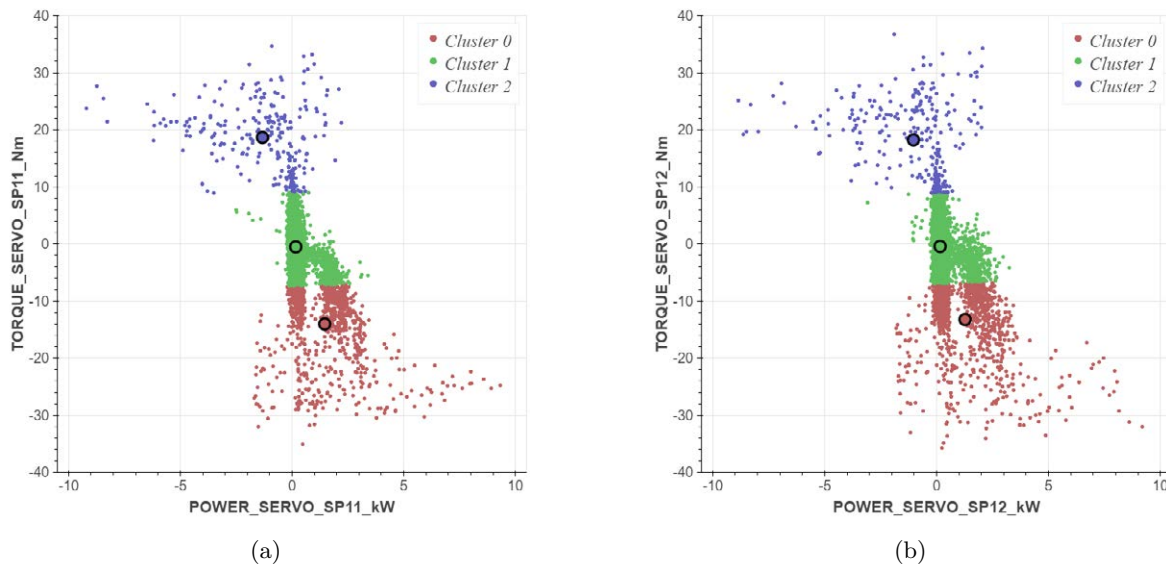


Figura 8.3: Resultados de K -medias (par vs. potencia) para (a) electromandrino 1 y (b) electromandrino 2.

son similares. La Tabla 8.1 resume las ubicaciones de los centroides en cada dimensión del clúster y el número de instancias agrupadas por cada electromandrino.

8.3.2. Jerárquico aglomerativo

Los parámetros de este algoritmo son: (a) método de vinculación, en donde se usa el método de Ward, (b) el número de clústers K que se elige como $K = 3$ y (c) la matriz de conectividad, la cual, al no tener datos espaciales, se utilizará exclusivamente para acelerar el algoritmo. Para ello se calculan los vecinos de cada instancia. Otros parámetros utilizados se han dejado en los valores por defecto que utiliza la librería `Scikit-Learn`.

La Figura 8.5 muestra el resultado del algoritmo representando el par del electromandrino en comparación con su consumo de energía. En este caso, los rangos del Clúster 0, Clúster 1 y Clúster 2 son muy similares a los encontrados por el algoritmo K -medias. En esta vista 2-D, las formas de clúster para cada electromandrino son similares con casi la misma cantidad de instancias agrupadas.

La relación entre temperatura y potencia se muestra en la Figura 8.6. En este caso, el Clúster 0 y el Clúster 2 tienen un rango de temperatura de $35,5$ a $36,5^{\circ}C$ y el Clúster 1 de $33,8$ a $36,5^{\circ}C$. En estas dimensiones, los tamaños y formas de los clústers también son similares. La Tabla 8.1 resume las ubicaciones de los centroides en cada dimensión del clúster y el número de instancias agrupadas por cada electromandrino. Es posible observar cómo los resultados del jerárquico aglomerativo tienen centroides con diferentes coordenadas a las de K -medias.

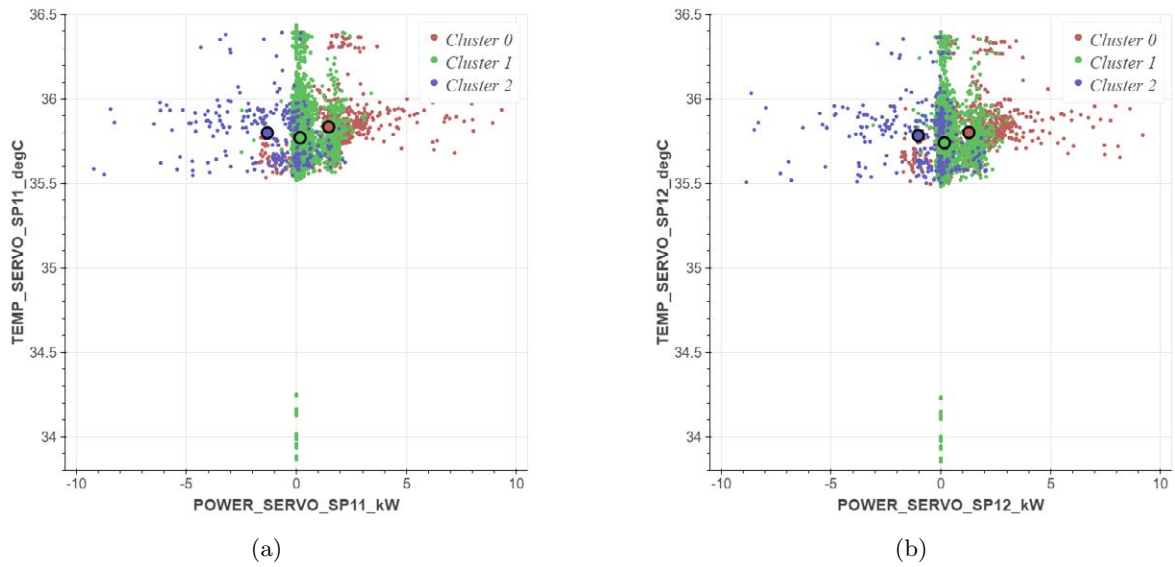


Figura 8.4: Resultados de K -medias (temperatura vs. potencia) para (a) electromandrino 1 y (b) electromandrino 2.

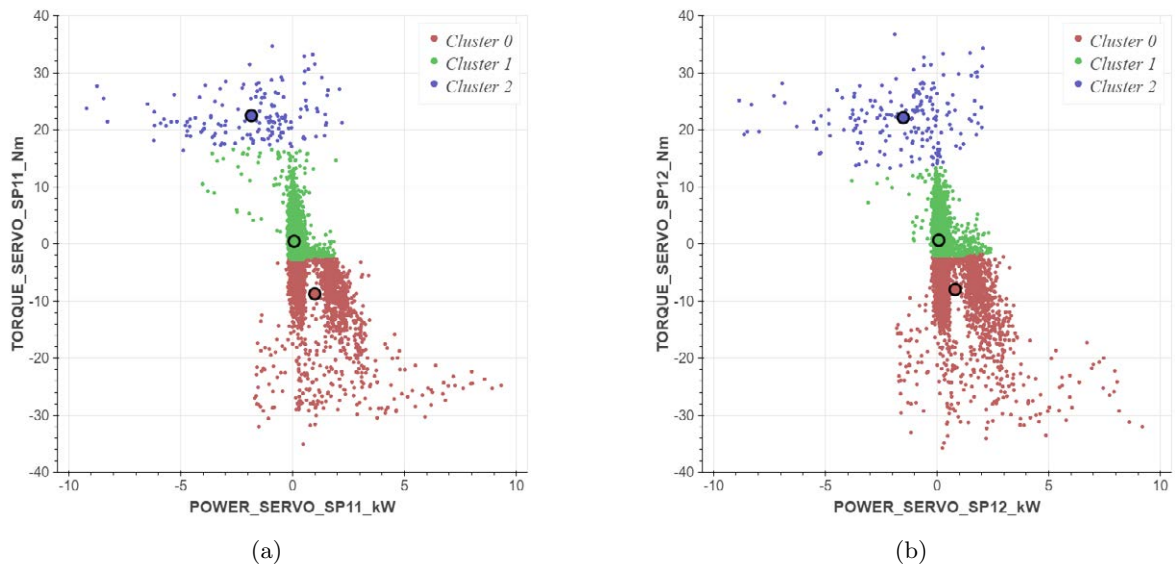


Figura 8.5: Resultados de jerárquico aglomerativo (par vs. potencia) para (a) electromandrino 1 y (b) electromandrino 2.

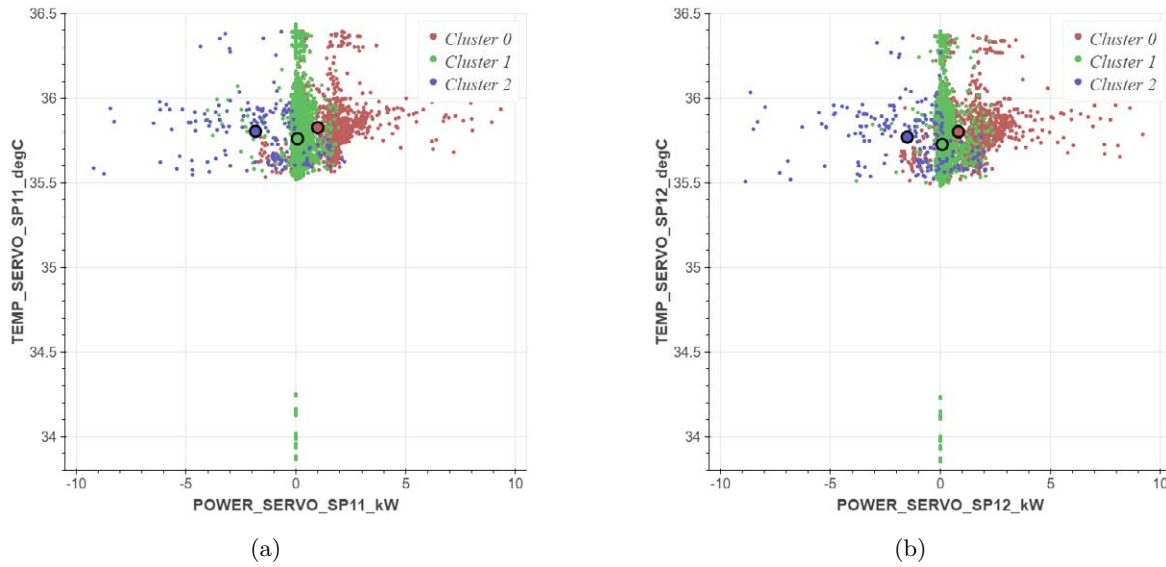


Figura 8.6: Resultados del jerárquico aglomerativo (temperatura vs. potencia) para (a) electromandrino 1 y (b) electromandrino 2.

8.3.3. Modelo de mixturas de Gaussianas

Este algoritmo tiene los siguientes parámetros: (a) el tipo de covarianza, donde se usa el tipo completo, es decir, cada grupo tiene su propia matriz de covarianza no restringida; (b) la inicialización, donde se usa K -medias para obtener los centroides del grupo inicial; y (c) número de clústers K , elegido como $K = 3$. Los otros parámetros utilizados se han dejado como valores por defecto de la librería `Scikit-Learn`.

La Figura 8.7 muestra el resultado de este algoritmo representando las variables del par del electromandrino en comparación con su consumo de potencia. En este caso, el Clúster 0 está bien distribuido en toda la zona y está integrado por instancias que tienen un par entre -33 y $33 N \cdot m$ y un consumo de energía de -10 a $10 kW$. El Clúster 1 está integrado por instancias que principalmente tienen un par entre -10 y $10 N \cdot m$ y un rango de consumo de energía muy estrecho de -1 a $1 kW$. El Clúster 2 está integrado por instancias con un par de -8 a $0 N \cdot m$ cuyo rango de potencia es de 0 a $2 kW$, y es completamente diferente al de los demás algoritmos de K -medias y jerárquico aglomerativo. En estas dos dimensiones las formas y los tamaños de cada clúster son similares para ambos ejes con casi las mismas instancias agrupadas. Sin embargo, el número de instancias por clúster es completamente diferente de los algoritmos K -medias y jerárquico aglomerativo.

La relación entre temperatura y potencia se muestra en la Figura 8.6. En este caso, el Clúster 2 tiene un rango de temperatura de $33,8$ a $36,5^{\circ}C$. El rango para Clúster 0 y Clúster 1 es de $35,5$ a $36,5^{\circ}C$. La Tabla 8.1 resume las ubicaciones de los centroides en cada dimensión del clúster y el número de instancias agrupadas por cada electromandrino.

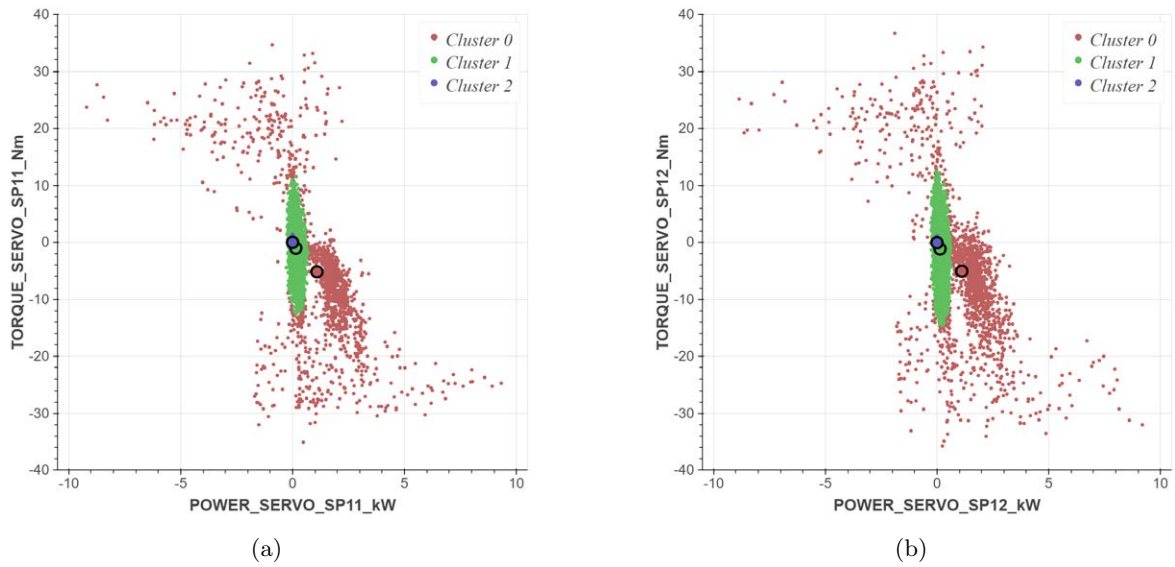


Figura 8.7: Resultados del *clustering* por modelos de mixturas de Gaussianas (par vs. potencia) para (a) electromandrino 1 y (b) electromandrino 2.

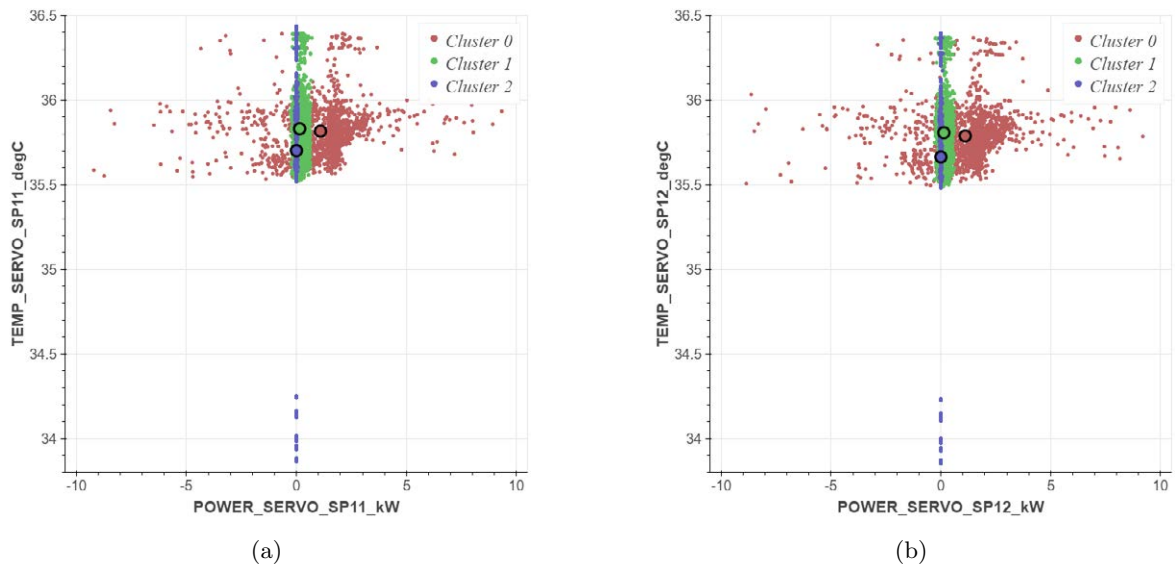


Figura 8.8: Resultados del *clustering* por modelos de mixturas de Gaussianas (temperatura vs. potencia) para (a) electromandrino 1 y (b) electromandrino 2.

Tabla 8.1: Centroides y número de instancias en cada grupo para cada electromandrino por cada algoritmo de *clustering*

Electromandrino No.	Cluster	<i>K</i> -medias				Jerárquico aglomerativo				Modelos de mixturas de Gaussianas			
		Potencia (<i>kW</i>)	Par (<i>N · m</i>)	Temperatura (°C)	No. Instancias	Potencia (<i>kW</i>)	Par (<i>N · m</i>)	Temperatura (°C)	No. Instancias	Potencia (<i>kW</i>)	Par (<i>N · m</i>)	Temperatura (°C)	No. Instancias
1	0	1,4	-14,0	35,8	9,66	1,0	-8,7	35,8	22145	1,1	-5,1	35,8	15,02
	1	0,1	-0,5	35,8	79003	0,1	0,5	35,8	67537	0,1	-1,0	35,8	38752
	2	-1,3	19,0	35,8	2634	-1,8	22,5	35,8	1615	0,0	0,0	35,7	37525
	0	1,3	-13,2	35,8	11518	0,8	-8,0	35,8	26882	1,1	-5,0	35,8	15005
2	1	0,1	-0,4	35,7	76909	0,1	0,6	35,7	62515	0,1	-1,1	35,8	39877
	2	-1,0	18,2	35,8	2,87	-1,5	22,0	35,8	1,9	0,0	0,0	35,6	36415

8.4. Discusión

El propósito de esta comparación es analizar el comportamiento del electromandrino desde diferentes puntos de vista para comprender cómo cada algoritmo describe el proceso de mecanizado y cómo se pueden utilizar para la búsqueda de conocimiento. De esta manera, los resultados se pueden analizar desde dos puntos de vista: algoritmos de *clustering* (Sección 8.4.1) y comportamiento interno del electromandrino (Sección 8.4.2).

8.4.1. Algoritmos de *clustering*

Desde el punto de vista del aprendizaje automático, el comportamiento de los algoritmos es muy similar. Sin embargo, analizando en detalle, al tener grandes cantidades de datos (más de 90000 instancias) se genera un verdadero desafío en términos de tiempo de procesamiento para los tres algoritmos. Para poder comparar efectivamente los algoritmos de acuerdo con el mismo estándar, las técnicas se han ejecutado en una máquina virtual Linux con un procesador de doble núcleo de 2,2 *GHz* y 8 *GB* de RAM.

En consecuencia, el algoritmo más lento ha sido el jerárquico aglomerativo, que ha tardado casi 35 *s* en completar la tarea. Se ha encontrado que es altamente dependiente de la matriz de conectividad, definida en la Sección 8.2.2, debido a las dificultades para encontrar vecinos, tal vez debido a la alta dinámica y a las no linealidades experimentadas durante el proceso, por ejemplo, cuando la herramienta toca el material. Como resultado, el algoritmo ha tenido problemas para encontrar clústers para fusionar. Los experimentos adicionales que utilizan diferentes bibliotecas de Python para construir la matriz de conectividad tal vez aumenten la eficiencia general. Sin embargo, esto se encuentra más allá del alcance de este capítulo. En el otro extremo de la escala, el algoritmo más rápido ha sido *K*-medias, el cual ha requerido menos de 0,8 *s* para completar la tarea. Este resultado posiciona el algoritmo *K*-medias como una herramienta útil para realizar una exploración preliminar del conjunto de datos.

Es interesante ver como los algoritmos *K*-medias y jerárquicos aglomerativos producen resultados similares. Viendo los resultados, se puede observar que los centroides del clúster de par alto (Clúster 2) solo difieren en alrededor de 3,6 *kW* para ambos electromandrilos, con la misma temperatura y casi la misma potencia. Esto significa que las instancias en cada grupo son comunes, es decir, las distancias euclídeas son muy similares. La Tabla 8.2 muestra el número de instancias coincidentes entre los clústers encontrados por cada algoritmo. En este caso, solo 13504 y 16335 instancias para electromandrino 1 y 2, respectivamente, no se comparten entre clústers. Esto significa que alrededor del 84 % de las instancias son iguales

Tabla 8.2: Porcentaje de distribución instancias entre clústers para cada K -medias y jerárquico aglomerativo e instancias compartidas entre ambos para cada electromandrino

Electromandrino No.	Clúster	K -medias	Jerárquico Aglomerativo	Instancias compartidas
1	0	9,66	22,14	9,66
	1	79,00	67,53	66,51
	2	2,63	1,61	1,61
2	0	11,51	26,88	11,51
	1	76,90	62,51	61,53
	2	2,87	1,90	1,90

para ambos algoritmos. Para el electromandrino 2, el número de instancias compartidas es mayor para Clúster 0 y Clúster 2.

Sin embargo, los resultados gráficos para K -medias y jerárquico aglomerativo difieren principalmente para el Clúster 1, mientras que el algoritmo jerárquico aglomerativo tiene un rango de par más amplio con menos instancias agrupadas. Solo se puede entender esto analizando el ciclo de mecanizado de manera detallada, donde el par y la potencia son bajos, pero un signo positivo puede corresponder al estado de la herramienta trabajando en vacío.

De manera opuesta, el algoritmo K -medias divide cada clúster usando el signo de la velocidad angular, el cual se infiere de la potencia y el par, por ejemplo, positivo (herramienta girando en sentido antihorario) y negativo (herramienta girando en sentido horario). Esto no proporciona el nivel de información novedosa encontrado por el jerárquico aglomerativo, pero sí podría ser útil para analizar el comportamiento del electromandrino en todos los niveles de velocidad angular: los elementos giratorios pueden funcionar de manera diferente cuando se cambia la velocidad angular.

Por el contrario, el algoritmo de mixturas de Gaussianas funciona de manera completamente diferente a los otros dos. Ya que esta técnica de clustering utiliza un modelo de mixturas de distribuciones de probabilidad en lugar de distancias euclídeas, es capaz de encontrar diferentes puntos de vista, significando esto nueva información. De hecho, éste es el único algoritmo capaz de reconocer dos grupos principales: eliminación de material (Clúster 0) y trabajo en vacío (Clúster 1). El Clúster 2 corresponde a la máquina inactiva, cuando los electromandrilos están detenidos con un consumo bajo de energía.

En este caso, el algoritmo de mixturas Gaussinas es útil para analizar el proceso de mecanizado independientemente de la dirección de rotación de la herramienta y la fase del ciclo. Esto es útil si se requiere una visión general del rendimiento de la máquina. Sin embargo, se debe tener en cuenta que probablemente un aumento en el número de componentes K proporcionaría más detalles e incluso se podría incluir la dirección de rotación.

Por otro lado, es interesante comprobar que los algoritmos de *clustering* abarcan todo el periodo temporal, es decir, son independientes del tiempo y solo dependen únicamente de los ciclos de mecanizado. Esto se ha comprobado formalmente mediante pruebas de uniformidad χ^2 , tomando la marca de tiempo de cada instancia y su clúster correspondiente, arrojando un p -valor $> 0,99$ en todos los casos. Esto significa que los tres clústers están representados

en todos los instantes de tiempo, rechazando la hipótesis nula que plantea la influencia del tiempo en el *clustering*.

8.4.2. Comportamiento interno del electromandrino

Desde el punto de vista de la ingeniería, los resultados obtenidos por el CPS basado en el aprendizaje automático muestran el comportamiento de elementos altamente precisos que trabajan en el mundo real. Una comparación de los patrones de cada electromandrino permite comprender claramente que ambos elementos realizan la misma operación. Las técnicas de *clustering* capturan esta situación, mostrando patrones similares independientemente del algoritmo utilizado.

De esta manera, los clústers del electromandrino 2 están más dispersos, lo que significa que este se ha desviado ligeramente de su punto de operación óptimo. Este es un resultado interesante, ya que se requiere verificar que los elementos mecánicos críticos sean robustos, donde la repetitividad sea uno de los indicadores más importantes. Es decir, las relaciones entre las variables deben ser constantes. Estas relaciones son difíciles de mantener en el mundo real, ya que hay otros factores externos, como el ruido, que causan desviaciones en los análisis. Sin embargo, se puede detectar que el rango de valores de par para el electromandrino 2 es mayor para el mismo valor de potencia. Esta situación no se aplica a la temperatura, ya que ésta es una variable más lenta, que necesita margen de tiempo para cambiar. Este fenómeno se conoce como inercia térmica.

La variable de temperatura muestra la efectividad de los sistemas de enfriamiento, que están trabajando para mantener los electromandrilos entre $35,5$ y $36,5^{\circ}C$. Este rango es de solo de $1^{\circ}C$, lo que es notable en un sistema con diferentes estados operativos en cada ciclo. Los tres algoritmos han encontrado una temperatura media de alrededor de $35,8^{\circ}C$ dependiendo del clúster, permitiendo así monitorizar el comportamiento del sistema de enfriamiento. Además, la medición de la temperatura tiene otros usos, como un indicador de mal funcionamiento interno. Por ejemplo, un aumento de la temperatura puede detectar un fallo en un rodamiento de bolas interno.

Otra consideración encontrada en los resultados muestra que se utiliza una pequeña cantidad de potencia de aproximadamente 2 kW para cambiar la dirección de rotación durante el mecanizado (modelos de GMM – Clúster 0). Los altos niveles de potencia y par muestran que los electromandrilos están trabajando a su capacidad nominal máxima, lo que significa que están correctamente seleccionados para realizar esta operación. De lo contrario, los niveles de potencia para electromandrilos más pequeños se concentrarán alrededor de valores altos y para electromandrilos más grandes a valores bajos.

Con respecto a la densidad de los clústers, el comportamiento del Clúster 2 en GMM es muy interesante porque su área es casi insignificante, solo visible en la Figura 8.8. Sin embargo, tiene alrededor de 37000 instancias muy agrupadas dentro de sus límites. En términos del proceso de mecanizado y la relación de este clúster con el estado inactivo, la pequeña dispersión de los datos en condiciones de operación parece ser lógica, ya que el electromandrino se detiene sin ninguna interferencia externa debido a la rotación de la herramienta o

al mismo proceso de extracción de viruta. De manera similar, el Clúster 1 con casi la misma cantidad de casos, se extiende sobre un área pequeña definida por límites claros de potencia, par y temperatura. En este caso, también tiene sentido, ya que las variables del electromandrino solo se ven afectadas por la inercia de rotación de la herramienta, es decir, tiene que compensar los desequilibrios relacionados con la herramienta y el portaherramientas. Por lo contrario, el Clúster 0 tiene una pequeña cantidad de instancias repartidas por todo el área. En este caso, el grupo muestra una dispersión variable debido al proceso de mecanizado, en el que se detectan discontinuidades en el material y la trayectoria.

Para los algoritmos jerárquicos aglomerativos y K -medias, este comportamiento es completamente diferente ya que estos algoritmos de *clustering* se rigen por el signo de la velocidad angular. En otras palabras, el Clúster 1, el más grande, está directamente relacionado con un electromandrino detenido, donde los niveles de par y potencia aumentan en términos de la necesidad de frenar para mantener la posición. El número de instancias es de aproximadamente 77500 (K -medias) y 65000 (jerárquico aglomerativo) para el Clúster 1, lo que significa que los electromandrilos están detenidos una gran parte del tiempo durante los ciclos de mecanizado. Sin embargo, el Clúster 0 es el segundo grupo más poblado, donde se obtiene un área de alta densidad de potencia positiva, cerca del centro de la gráfica. Desde el punto de vista del mecanizado, las áreas de alta densidad significan que el comportamiento de la punta de la herramienta es estable durante la operación después de su primer contacto con el material. El Clúster 2 es el grupo menos poblado, que está directamente relacionado con el Clúster 1 de la fase de operación de mecanizado siendo diferente donde la velocidad angular es la opuesta.

8.5. Conclusiones

Una conclusión general de este capítulo es que los algoritmos de aprendizaje automático no supervisados integrados en CPSs son los habilitadores clave para trabajar en herramientas de diagnóstico altamente precisas. Estas aplicaciones de búsqueda de conocimiento podrían ser el primer paso hacia el diagnóstico en tiempo real y luego hacia las herramientas de prognosis, que serían altamente beneficiosas para las nuevas aplicaciones de mantenimiento predictivo basadas en la detección.

En términos de técnicas de *clustering*, el modelo de mixtura de Gaussianas es capaz de encontrar clústers que aportan más información en relación a las condiciones de operación. A pesar de tener una implementación compleja, se ha encontrado que los resultados son óptimos en términos de interpretación por parte de expertos en máquinas-herramienta. Esta conclusión es el fundamento principal para los temas tratados en la Parte III.

Por otro lado, el algoritmo jerárquico aglomerativo es útil para analizar en detalle las fases del ciclo. Sin embargo, está muy influenciado por variables inherentes como la velocidad angular, lo que puede interferir en la interpretación o la búsqueda de nuevo conocimiento. El K -medias tiene las mismas características que el algoritmo jerárquico aglomerativo en términos de variables inherentes. Sin embargo, es considerablemente más rápido, resultando

ser una herramienta muy potente para analizar rápidamente conjuntos de datos.

En términos de desarrollo de algoritmos, el objetivo es buscar adaptar un análisis de *clustering* estático a un proceso principalmente dinámico, lo cual es tratado en los Capítulos 9 y 10. De esta manera, la búsqueda de conocimiento podrá ejecutarse en tiempo real, ayudando a comprender la evolución del clúster en términos de forma (es decir, ver cómo las características de mecanizado identificadas cambian con el tiempo) y la cantidad de clústers (es decir, identificar nuevas características de mecanizado).

Parte III

Contribuciones al *clustering* de *data streams*

Clustering probabilístico de *data streams* con modelos dinámicos de mixturas de Gaussianas

De acuerdo con las conclusiones extraídas en la Parte II y de su aplicación en procesos industriales, se entiende que el *clustering* basado en GMM es una buena herramienta para poder definir los patrones de comportamiento de los elementos industriales. Sin embargo, este algoritmo carece de capacidades de trabajo en entornos de *data streams* debido a que usa el algoritmo EM definido en la Sección 4.5 que resulta principalmente, en este tipo de aplicaciones, lento, exigente computacionalmente y limitado a la hora de trabajar con grandes cantidades de datos.

Por esta razón, en este capítulo se propone un algoritmo probabilístico basado en GMM que permita trabajar con *data streams*. Este capítulo se ha publicado en [Diaz-Rozo et al. \(2018\)](#).

9.1. Introducción

En el contexto de procesos de ingeniería reales, como aquellos mostrados en el Capítulo 8 y ([Diaz-Rozo et al., 2017](#)), los GMM tienen la capacidad de discriminar claramente lo que está sucediendo dentro de un proceso, mejorando lo obtenido por K -medias o por el *clustering* jerárquico. Estos algoritmos asumen una imagen estática del problema, donde no se tiene en cuenta la variación de los clústers (forma, número y posición) a lo largo del tiempo. En este caso, los algoritmos particionales, como K -medias, tienden a agrupar las instancias según las variables subyacentes que no se incluyen en el análisis, como la velocidad angular, lo que no aporta ningún conocimiento nuevo sobre el proceso analizado. El algoritmo jerárquico aglomerativo trabaja de manera similar.

El análisis de datos realizado con GMM tiene una capacidad comprobada para la búsqueda de conocimiento sobre los procesos de ingeniería, brindando más información que otros. Sin

embargo, esta ventaja de los algoritmos de *clustering* basados en modelos de probabilidad se ve reducida por su incapacidad, según nuestro conocimiento, para tratar con *data streams*. Por ejemplo, Bouguelia et al. (2018) muestran como los GMM se utilizan como un método simple para resumir los datos en mixturas de Gaussianas y con *data streams* en ausencia de cambios de estado o *concept drift* (Sección 5.1). Generalmente, los GMM se aplican para ayudar a otros modelos, como las redes Bayesianas evolutivas capaces de tratar con *data streams*, aproximando la distribución conjunta, compleja de aproximar por otros métodos (Wang et al., 2017). Sin embargo, los parámetros de distribución asociados se estiman *offline*, ejecutando el algoritmo EM únicamente sobre los datos históricos.

Como el *data stream* y la dinámica del proceso son una consideración importante para la industria, el objetivo de este capítulo está motivado por la necesidad de una técnica de *clustering* probabilístico basado en GMM, que tenga la capacidad de lidiar con la evolución dinámica y los cambios de los procesos industriales, proporcionando una nueva herramienta de aprendizaje automático para el IIoT.

De esta manera, con el ánimo de investigar las formas de detectar la degradación del proceso, se propone un algoritmo de *clustering* basado en GMM adaptado a entornos dinámicos denominado *clustering* probabilístico dinámico basado en mixturas de Gaussianas (GDPC). Adicionalmente, partiendo de un *data stream*, el GDPC proporciona las probabilidades de que cada instancia sea miembro de cada clúster, lo cual sirve como medida de la robustez de la asignación a cada clúster, detectando cuándo los datos no están bien representados por el modelo, es decir, cuando hay un *concept drift*.

9.2. Metodología

Siguiendo las líneas generales para los algoritmos de *data stream* marcadas por Sebastiao y Gama (2009) y Aggarwal et al. (2003), se definen los siguientes pasos:

1. Entrenamiento *offline* de GMM con un conjunto de datos inicial $\mathcal{D} = \{\mathbf{x}^1, \dots, \mathbf{x}^N\}$ de tamaño N , donde cada instancia tiene n componentes, $\mathbf{x}^i = (x_1^i, \dots, x_n^i)$ (Sección 9.2.1);
2. Test de ajuste de nuevas instancias entrantes del *data stream* al modelo actual, entrenado en el paso 1 o 7, medida mediante la log-verosimilitud (Sección 9.2.2);
3. Detección de valores atípicos (Sección 9.2.3);
4. Detección del *concept drift* mediante una métrica basada en el valor acumulado de valores atípicos encontrados (Sección 9.2.4);
5. Si se detecta un *concept drift*, ir al paso 7; de lo contrario, continuar con el paso 6 para realizar el *clustering*;
6. Si no se detecta un *concept drift*, asignar la instancia a cada componente del GMM dependiendo de su probabilidad *a posteriori* (Sección 9.2.5);

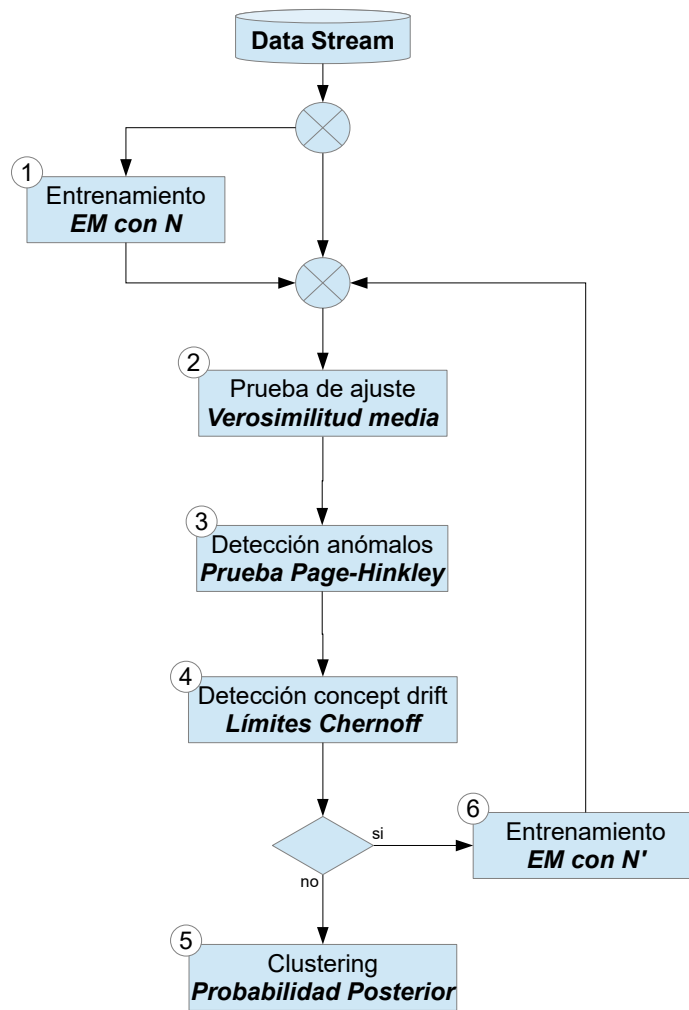


Figura 9.1: Esquema del algoritmo.

7. Si se encuentra un *concept drift*, el modelo se reajusta utilizando la última ventana de Chernoff $\mathcal{D}' = \{\mathbf{x}^1, \dots, \mathbf{x}^{N'}\}$ que trabaja como una ventana de datos móvil adaptativa (Sección 9.2.4);

La secuenciación de los pasos de este algoritmo se muestra en la Figura 9.1. En las siguientes secciones se da una descripción detallada de cada paso.

9.2.1. Entrenamiento *offline* del modelo GMM

Este paso se puede considerar *estático*, ya que el algoritmo debe entrenarse con una ventana de datos de tamaño N . La idea es crear un modelo de mixtura de Gaussianas inicial a partir de estos primeros datos para luego usarlo como referencia. La definición de GMM y

la forma de generar este modelo de mixturas se describió en la Sección 4.5.

Teniendo el modelo de referencia, inducido a partir del conjunto de datos inicial \mathcal{D} , el siguiente paso es realizar un test de ajuste con nuevas instancias provenientes del *data stream*.

9.2.2. Test de ajuste

El test de ajuste comprueba si la nueva instancia del *data stream* entrante está bien representada por el GMM actual. Para medir esto se utiliza el criterio de la log-verosimilitud media (Akaike, 1974), que se calcula a partir de la Ecuación (4.2) y (4.5)

$$\overline{\log L}(\Psi) = \frac{1}{N} \sum_{i=1}^N \log \left(\sum_{k=1}^K \pi_k f_k(\mathbf{x}^i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right), \quad (9.1)$$

y se compara con la log-verosimilitud $\log L_w(\Psi)$ de la nueva instancia \mathbf{x}^w . Una diferencia significativa en el valor de $\log L_w(\Psi)$ en comparación con $\overline{\log L}(\Psi)$ indica que los nuevos datos no se representan bien por el modelo de mixturas actual. A continuación, estos valores de $\overline{\log L}(\Psi)$ se utilizan para detectar lecturas atípicas, tal y como se describe en la siguiente sección.

9.2.3. Detección de valores atípicos

Una vez se encuentra el resultado del test de ajuste para el *data stream* (Ecuación (9.1)), es necesario conocer el valor de $\overline{\log L}(\Psi)$ para el que la instancia \mathbf{x}^w está representada correctamente por el modelo o no. Cuando una instancia no está representada por el GMM actual, se etiqueta como un valor atípico, y se utiliza como valor de entrada en el paso de detección de *concept drift* (Sección 5.1).

Para esto, Sebastiao y Gama (2009) y Borchani et al. (2016) aplicaron con éxito el test de Page–Hinkley (Hinkley, 1971), para análisis secuenciales de *data streams* que toma la varianza muestral de los datos analizados como métrica. Según describen Mouss et al. (2004), esta prueba asocia la varianza muestral con el comportamiento normal de la secuencia de datos. Por lo tanto, cualquier perturbación, como la degradación, se asocia con un cambio abrupto sobre la varianza pudiéndose asociar con un cambio en el comportamiento del *data streams*. En este caso, la hipótesis nula del test es que la instancia \mathbf{x}^w está correctamente representada por el GMM, es decir no existe *concept drift*.

Para realizar la prueba de Page–Hinkley, $\overline{\log L}(\Psi)$ se calcula a partir de la Ecuación (9.1) a medida que llegan nuevas instancias del *data streams*. Este valor se almacena en un vector definido por

$$\overline{\log L}(\Psi) = (\overline{\log L}_1(\Psi), \dots, \overline{\log L}_{N'}(\Psi)), \quad (9.2)$$

donde N' es explicado por Chernoff (1952), Watanabe (2000) y Barbará y Chen (2001) como

un tamaño de ventana adaptable limitada por

$$N' \leq \frac{3(1+\epsilon)}{(1-\epsilon)\epsilon^2 p} \ln\left(\frac{2}{\phi}\right), \quad (9.3)$$

donde $0 < \epsilon < 1$ es el límite de error aditivo, $0 < \phi < 2$ es una constante para controlar la probabilidad de que una instancia se asigne con éxito y p es la probabilidad de que la instancia \mathbf{x}^w no sea atípica, es decir, el número de instancias no atípicas sobre el total de instancias disponibles. El tamaño de la ventana N' cambia inversamente a p , lo que significa que si una nueva instancia está bien representada (p grande) o mal representada (p pequeña), el tamaño de la ventana es pequeño o grande, respectivamente. Esta ventana adaptativa ayuda a optimizar la cantidad de datos almacenados. Es decir, si una nueva instancia se ajusta al modelo no es necesario tener una gran cantidad de datos históricos. Sin embargo, si se comienzan a detectar instancias que no ajustan, el tamaño de ventana comienza a aumentar.

De esta manera, la diferencia acumulada entre la log-verosimilitud media y el valor medio acumulado (CUM_w) se define por

$$CUM_{N'} = \frac{1}{N'} \sum_{w=1}^{N'} (\overline{\log L_w(\Psi)} - mean_w) - \delta, \quad (9.4)$$

donde $mean_w = \frac{1}{w} \sum_{t=1}^w \overline{\log L_t(\Psi)}$ es el valor medio de los primeros componentes w del vector definido en la Ecuación (9.2), y δ es un parámetro aleatorio de tolerancia positiva para controlar la magnitud máxima de los cambios aceptados. La forma de seleccionar este valor se describe en la Sección 9.3. Adicionalmente, se define un valor MAX_w como:

$$MAX_w = \max \{CUM_w, \quad w = 1, \dots, N'\}$$

Por lo tanto, el umbral PH_w se define como $PH_w = MAX_w - CUM_w$ ($w = 1, \dots, N'$), y así,

$$reject_w = \begin{cases} 1 & \text{si } PH_w > \lambda \\ 0 & \text{en otro caso} \end{cases}$$

donde $\lambda > 0$ es el valor que controla la hipótesis nula y donde 1 significa que la instancia \mathbf{x}^w se etiqueta como un valor atípico (se rechaza la hipótesis). Por lo tanto, un vector $reject_w$ de tamaño N' almacena 1 o 0 si la instancia \mathbf{x}^w es un valor atípico o no, respectivamente.

9.2.4. Detección del *concept drift*

El objetivo de este paso es diferenciar entre los valores atípicos producidos por el ruido en el *data stream* (generado por las incertidumbres propias del sistema de adquisición de datos) y los relacionados con el modelo de mixtura de Gaussianas, donde un grupo de instancias, cuyo tamaño es mayor que un umbral definido, ya no se ajusta correctamente. Este caso se define como *concept drift* (Sección 5.1).

Para detectar un *concept drift*, se adapta el enfoque propuesto por [Barbará y Chen \(2001\)](#) para aplicar las cotas de Chernoff y definir el número mínimo de instancias s no atípicas (es decir, instancias bien representadas por el modelo) en una ventana definida, de acuerdo con

$$s = \frac{3(1 + \epsilon)}{\epsilon^2} \ln \left(\frac{2}{\phi} \right), \quad (9.5)$$

y el tamaño de las ventanas adaptables del *data stream* definida en la Ecuación (9.3), lo cual permite controlar la cantidad de instancias almacenadas para el entrenamiento si un *concept drift* es detectado. La idea principal de este enfoque es determinar la cantidad total de valores atípicos necesarios para determinar si el GMM actual ha caducado y se debe generar uno nuevo a partir de los datos actuales. De esta manera, r es la suma de valores no atípicos ($r = \sum_{i=1}^{N'} (1 - reject_i)$), en donde $r < s$ (los valores no atípicos son menores al valor permitido), significa que se ha detectado un *concept drift*.

En cuanto se detecta que el GMM actual está caducado se estima uno nuevo a partir de los datos de la ventana adaptativa utilizando el algoritmo descrito en la Sección 9.2.1, con N definido como el N' más bajo y determinado por la Ecuación (9.3). Asimismo, r se debe inicializar como 0, el cual aumenta su valor cada vez que un GMM representa correctamente una instancia. Adicionalmente, $p = \frac{r}{N'}$, siendo su valor inicial $p = 0$ y estando actualizado en cada ciclo.

9.2.5. Asignación de cada instancia a cada clúster

El siguiente paso en el algoritmo GDPC es asignar una instancia a cada componente a través de la probabilidad *a posteriori* de que la instancia \mathbf{x}^i pertenezca al componente k de la mezcla, como se refleja en la Ecuación (4.3). Es decir,

$$z_{ki} = \begin{cases} 1 & \text{si } k = \arg \max_h \tau_h(\mathbf{x}^i; \Psi) \\ 0 & \text{en otro caso} \end{cases}$$

9.2.6. Resumen del algoritmo GDPC

Recopilando el papel de cada uno de los módulos descritos en las Secciones 9.2.1 a 9.2.5, es posible proporcionar una visión clara del funcionamiento e interacciones que conforman la columna vertebral del algoritmo GDPC.

Una vez unidos los módulos el algoritmo está listo para recibir un *data stream* proveniente de un proceso específico e inicializar los parámetros relacionados con cada módulo: δ , λ , ϵ , ϕ y N . Estos parámetros se seleccionan siguiendo el enfoque presentado en la Sección 9.4. Usando estos valores el tamaño de ventana mínimo permitido, s , se calcula utilizando la Ecuación (9.5). Otros parámetros, como la probabilidad, p y el número de instancias sin valores atípicos, r , se deben inicializar a cero.

Después de la inicialización un conjunto de datos de entrenamiento \mathcal{D} se llena con el *data stream* hasta que el número de instancias sea igual al parámetro N . Posteriormente, el GMM

inicial puede entrenarse con \mathcal{D} usando el algoritmo EM para estimar los parámetros de la mixtura de Gaussianas. A partir de este paso el algoritmo comienza a analizar el *data stream* procesando una instancia cada vez y utilizando el GMM como el resumen estadístico que representa los datos analizados.

El análisis del *data stream* comienza encontrando la ventana adaptativa de tamaño N' utilizando las cotas de Chernoff que trabajan de manera similar a un búfer o ventana temporal. De esta manera, se almacenan las nuevas instancias N' que llegan de la secuencia de datos. Mientras tanto, cada instancia del *data stream* se compara con el GMM actual utilizando la log-verosimilitud media. Este valor medio es monitorizado por el test de Page–Hinkley en busca de valores atípicos. Si este test de hipótesis no es rechazado se entiende que la instancia está correctamente representada por el GMM, registrándose mediante la actualización de r .

El siguiente paso compara r y s . De esta manera, si el número de instancias correctamente representadas r es menor que s se detecta un punto de inflexión o *concept drift* en el *data stream*. En este caso, se ajusta un nuevo GMM utilizando los datos más recientes almacenados en la ventana adaptativa y el proceso continúa con la siguiente instancia del *data stream*. Si r permanece más pequeño que s (es decir, sin *concept drift*), la probabilidad, p se actualiza con r/N' y la instancia se asigna a un componente GMM utilizando su probabilidad *a posteriori*, continuando de esta manera con la siguiente instancia de la secuencia de datos.

El proceso completo se resume en el Algoritmo 9.1.

Algorithm 9.1 GDPC

Require: Data stream
Require: N
Require: δ y λ
Require: Inicializar s con $0 < \epsilon < 1$ y $0 < \phi < 2$
Require: Inicializar $p = 0$ y $r = 0$

while $i \leq N$ **do**
 Instancia es asignada a los datos de entrenamiento: $\mathcal{D} \leftarrow \mathbf{x}^i$

function ENTRENAMIENTO DEL GMM(\mathcal{D})
 GMM inicial entrenado con \mathcal{D} usando el algoritmo EM

while *data stream* **do**
 Inicializar el tamaño de ventana N' con Ecuación (9.3)

function TEST DE AJUSTE(\mathbf{x}^i)
 Medir $\log \bar{L}(\Psi)$

function DETECCIÓN DE VALORES ATÍPICOS($\log \bar{L}(\Psi)$)
 CUM_w es calculada usando Ecuación (9.4)
 Añadir el valor CUM_w al vector MAX_w
 Comenzar el test de Page–Hinkley con $PH_w = MAX_w - CUM_w$
 if $PH_w > \lambda$ **then**
 Un valor atípico es detectado
 $outliers+ = 1$
 $r \leftarrow (1 - outliers)$

function DETECCIÓN DE *concept drift*(r , $\mathcal{D} \leftarrow \{\mathbf{x}^1, \dots, \mathbf{x}^{N'}\}$)
 Calcular s con Ecuación (9.5)
 if $r < s$ **then**
 Un *concept drift* es detectado: entrenar nuevo GMM con \mathcal{D}
 else if $r > s$ **then**
 Actualizar probabilidad $p \leftarrow r/N'$
 Asignar nueva instancia

9.3. Evaluación del rendimiento del algoritmo GDPC

De acuerdo con el GDPC propuesto es necesario evaluar la influencia de los parámetros descritos en los resultados para entender y plantear una metodología de configuración del algoritmo. Estos parámetros están relacionados principalmente con los tests de hipótesis que se plantean, agrupándose de la siguiente manera:

1. Entrenamiento *offline* de GMM (Sección 9.2.1):
 - N : Número de instancias de entrenamiento iniciales.
2. Test de hipótesis de Page–Hinkley (Sección 9.2.3):
 - λ : Umbral positivo que define el valor a partir del cual se rechaza la hipótesis nula de una instancia \mathbf{x}^i para ser representada correctamente por el GMM.
 - δ : Tolerancia positiva a los cambios que se deben absorber, según la Ecuación (9.4).
3. Cotas de Chernoff (Sección 9.2.4):
 - ϵ : límite de error aditivo ($0 < \epsilon < 1$).
 - ϕ : Constante para controlar la probabilidad de que una instancia se asigne con éxito ($0 < \phi < 2$).

9.3.1. Datos experimentales

Para estudiar el efecto de los parámetros en la capacidad de detección del *concept drift*, se crean conjuntos de datos sintéticos simulando valores aleatorios de un GMM. Como el objetivo principal de estos experimentos es calibrar la capacidad del GDPC de detectar el *concept drift* es fundamental conocer de antemano la ubicación del mismo dentro del *data stream* y transformar este proceso de calibración en un proceso supervisado. De esta manera, se genera un *data stream* de referencia, con una variedad predefinida de *concept drifts*, útil para comparar los resultados del algoritmo en términos de detección y su retraso.

Por lo tanto, los *data streams* se crean concatenando diferentes subconjuntos de instancias cada una de las cuales se simula con un GMM de diferentes combinaciones de parámetros. De esta manera, estas variaciones permiten simular los *concept drifts*.

Se parte de un modelo sintético con $K = 3$ componentes para este escenario en particular,

$$\boldsymbol{\theta} = (\pi_1, \pi_2, \pi_3, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \boldsymbol{\Sigma}_3).$$

donde $\boldsymbol{\mu}_k$ ($k = 1, 2, 3$) es el vector de medias y $\boldsymbol{\Sigma}_k$ ($k = 1, 2, 3$) es la matriz de varianzas.

Diferentes valores de *offset* y ruido Gaussiano se seleccionan para generar artificialmente diferentes *data streams* sintéticos. Los cambios en la magnitud del *offset* o el ruido Gaussiano permiten simular *concept drifts*. De esta manera, los valores de *offset* son $\pm 5\%$, $\pm 10\%$, $\pm 25\%$ y para el ruido Gaussiano se define una variable aleatoria $noise_\sigma$, donde $\sigma = \{0, 05, 0, 10, 0, 25\}$

Tabla 9.1: Características de los subconjuntos con instancias simuladas.

Subset No.	l_{subset}	π_l	μ_l	Σ_l
0	15000	π	μ_0	Σ_0
1	7000	π	$0,75 \cdot \mu_0$	$0,75 \cdot \Sigma_0$
2	10000	π	$1,25 \cdot \mu_0$	$1,25 \cdot \Sigma_0$
3	5000	π	$noise_{\sigma=0,25} \cdot \mu_0$	$noise_{\sigma=0,25} \cdot \Sigma_0$
4	7000	π	$0,90 \cdot \mu_0$	$0,90 \cdot \Sigma_0$
5	10000	π	$1,10 \cdot \mu_0$	$1,10 \cdot \Sigma_0$
6	5000	π	$noise_{\sigma=0,10} \cdot \mu_0$	$noise_{\sigma=0,10} \cdot \Sigma_0$
7	7000	π	$0,95 \cdot \mu_0$	$0,95 \cdot \Sigma_0$
8	10000	π	$1,05 \cdot \mu_0$	$1,05 \cdot \Sigma_0$
9	5000	π	$noise_{\sigma=0,05} \cdot \mu_0$	$noise_{\sigma=0,05} \cdot \Sigma_0$

Tabla 9.2: *data streams* sintéticos.

<i>data stream</i> No.	Secuencia de subconjuntos (Tabla 9.1)	Ubicación del <i>concept drift</i> (CD_{set})
1	{0, 1, 2, 3}	{15000, 22000, 32000}
2	{0, 4, 5, 6}	{15000, 22000, 32000}
3	{0, 7, 8, 9}	{15000, 22000, 32000}

cuando su valor aleatorio se simula con distribuciones Gaussianas: $\mathcal{N}(0, 0, 05)$, $\mathcal{N}(0, 0, 10)$ y $\mathcal{N}(0, 0, 25)$, respectivamente.

Se genera un subconjunto original (sin variaciones) más otros nueve subconjuntos de instancias con diferentes longitudes l_{subset} y variaciones (de la forma $param_l = offset \cdot param_0$ o $param_l = noise \cdot param_0$) de sus parámetros. La descripción de la forma en que se modifican los parámetros para obtener diferentes subconjuntos se muestra en la Tabla 9.1.

De esta manera, la concatenación de los diferentes subconjuntos de instancias descritas en la Tabla 9.1 permite obtener diferentes *data streams* para la experimentación. Como resultado se obtienen tres *data streams* creados a partir de estas concatenaciones que se muestran en la Tabla 9.2. Los *data streams* tienen desplazamientos positivos, negativos y ruido Gaussiano, según la magnitud definida en la Tabla 9.1. Estas secuencias de datos tienen un número igual de instancias, es decir, $l_{datastream} = 37000$ con tres *concept drifts* en las posiciones dadas en el conjunto CD_{set} .

9.3.2. Diseño de experimentos

Para explorar de manera eficiente el efecto de los parámetros en el comportamiento del modelo es necesario un diseño de experimentos factorial y ortogonal en el que los factores (variables para análisis) y los niveles (rangos de valores de cada factor) estén equilibrados. Este

Tabla 9.3: Parámetros y niveles para el diseño de experimentos.

Parámetros	Nivel 1	Nivel 2	Nivel 3	Nivel 4
λ	0,10	1,00	10,0	100
δ	0,01	0,10	1,00	10,0
ϵ	0,10	0,25	0,75	0,90
ϕ	0,01	0,27	0,54	0,80
N	10	100	1000	10000

tipo de diseños se denominan métodos Taguchi (Taguchi y Wu, 1979). La clave de entrada para seleccionar el diseño es el número requerido de parámetros. En este caso, se están estudiando cinco parámetros para el GDPC: λ , δ , ϵ , ϕ y N ; cada uno con cuatro niveles, definidos de esta manera para abarcar un amplio espectro de valores. Por lo tanto, el diseño que mejor se adapta a este número de parámetros y niveles es el diseño ortogonal Taguchi L16 (tipo B) con cinco parámetros de cuatro niveles, es decir, un diseño de 4^5 con 16 observaciones. La Tabla 9.3 muestra los parámetros y niveles que se utilizarán en la experimentación.

Para estudiar la influencia de estos parámetros se determinan las siguientes métricas a extraer de los experimentos:

1. *Concept drift* detectados (n_{CD}): la cantidad de *concept drift* detectados dentro de un *data streams*.
2. Instancia marcada como punto de inflexión debida a un *concept drift* (CD_{det}): la primera instancia en la que el algoritmo detecta un *concept drift*, definida como un punto de inflexión.
3. Número de *concept drift* verdaderos (CD_{true}): la cantidad de *concept drift* verdaderos medidos por su ubicación CD_{det} en comparación con CD_{set} de la Tabla 9.2.
4. Tiempo de procesamiento en segundos (t): el tiempo requerido por el algoritmo para procesar el *data streams*.

A partir de las observaciones, el rendimiento se evalúa utilizando las herramientas descritas en la Sección 3.4.1 extrayendo los valores de precisión, *recall*, especificidad, y *F-score*. De esta manera, utilizando la matriz de confusión es posible comparar una clase relacionada con el *concept drift* real (c_t) y el *concept drift* detectado por el GDPC (c_m), es decir,

1. c_t es positivo y c_m es positivo: las instancias detectadas por el modelo son *concept drift* reales, es decir, CD_{true} .
2. c_t es negativo y c_m es positivo: no todas las instancias detectadas por el modelo son *concept drifts*, es decir, $CD_{det} - CD_{set}$.

Tabla 9.4: Resultados experimentales para el *data streams* 1.

Test	λ	δ	ϵ	ϕ	N	t	CD_{det}	CD_{true}	n_{CD}	Prec.	Recall	Espec.	F-score
1	0,1	0,01	0,10	0,01	10	17,27	15	1	{15546}	0,889	0,333	1,000	0,500
2	0,1	0,10	0,25	0,27	100	18,72	230	3	{15140, 22020, 32100}	0,994	1,000	0,994	0,997
3	0,1	1,00	0,75	0,54	1000	18,45	717	3	{15009, 22021, 32021}	0,981	1,000	0,981	0,990
4	0,1	10,00	0,90	0,80	10000	16,40	320	3	{15059, 22002, 32043}	0,992	1,000	0,991	0,995
5	1,0	0,01	0,25	0,54	10000	9,66	35	3	{15127, 22205, 32145}	0,999	1,000	0,999	0,999
6	1,0	0,10	0,10	0,80	1000	8,23	4	3	{15070, 22105, 32155}	1,000	1,000	1,000	1,000
7	1,0	1,00	0,90	0,01	100	8,03	3	3	{15352, 22477, 32773}	1,000	1,000	1,000	1,000
8	1,0	10,00	0,75	0,27	10	8,28	12	3	{15069, 22073, 32324}	1,000	1,000	1,000	1,000
9	10,0	0,01	0,75	0,80	100	8,17	3	3	{15026, 22030, 32061}	1,000	1,000	1,000	1,000
10	10,0	0,10	0,90	0,54	10	8,08	4	3	{15105, 22103, 32202}	1,000	1,000	1,000	1,000
11	10,0	1,00	0,10	0,27	10000	9,36	3	3	{15138, 22478, 32770}	1,000	1,000	1,000	1,000
12	10,0	10,00	0,25	0,01	1000	8,18	2	2	{15382, 22150}	0,800	0,667	1,000	0,800
13	100,0	0,01	0,90	0,27	1000	7,91	3	3	{15140, 22140, 32370}	1,000	1,000	1,000	1,000
14	100,0	0,10	0,75	0,01	10000	9,66	3	3	{15591, 22463, 32544}	1,000	1,000	1,000	1,000
15	100,0	1,00	0,25	0,80	10	9,29	3	3	{15048, 22056, 32057}	1,000	1,000	1,000	1,000
16	100,0	10,00	0,10	0,54	100	9,97	3	3	{15460, 22180, 32260}	1,000	1,000	1,000	1,000

- c_t es positivo y c_m es negativo: no se detectan todas las instancias con *concept drifts* verdaderos, es decir, $CD_{set} - CD_{true}$.
- c_t es negativo y c_m es negativo: la cantidad de instancias no detectadas como *concept drift*, es decir, $N - CD_{det}$.

De acuerdo con la Tabla 3.1, los valores de mérito se definen como:

- Precisión: $\frac{CD_{true} - n_{CD} + N}{N}$
- Recall (sensibilidad) - verdaderos positivos (TP): $\frac{CD_{true}}{n_{CD}}$
- Especificidad - verdaderos negativos (TN): $\frac{N - n_{CD}}{N - |CD_{set}|}$
- F-score: $\frac{2 \times Recall \times Precision}{Recall + Precision}$

9.3.3. Resultados experimentales

Los resultados de estos experimentos se muestran en las Tablas 9.4, 9.5 y 9.6.

En la Tabla 9.4 se muestran los resultados para el *data stream* 1. En este caso, el algoritmo es capaz de detectar el primer *concept drift* con retrasos no mayores a 591 instancias para el experimento 14 (punto de inflexión detectado en la instancia 15591) en el conjunto completo de experimentos. Los experimentos 1 y 12 son los únicos sin detección correcta del *concept drift* verdadero (CD_{true}), con solo una y dos detecciones, respectivamente. Por otro lado, en los experimentos 6 a 11 y 13 a 16 se logra el 100 % de precisión sin ningún falso positivo. Por lo tanto, el 87,5 % de los experimentos tienen combinaciones de parámetros que son capaces de detectar *concept drifts* verdaderos (100 % de *recall*), pero solo el 62,5 % de ellos con una precisión del 100 %.

Tabla 9.5: Resultados experimentales para el *data stream 2*.

Test	λ	δ	ϵ	ϕ	N	t	CD_{det}	CD_{true}	n_{CD}	Prec.	Recall	Espec.	F -score
1	0,1	0,01	0,10	0,01	10	24,34	19	1	{15546}	0,909	0,333	1,000	0,500
2	0,1	0,10	0,25	0,27	100	17,47	228	3	{15140, 22020, 32007}	0,994	1,000	0,994	0,997
3	0,1	1,00	0,75	0,54	1000	21,85	719	3	{15003, 22011, 32039}	0,981	1,000	0,981	0,990
4	0,1	10,00	0,90	0,80	10000	22,37	332	3	{15018, 22047, 32032}	0,991	1,000	0,991	0,995
5	1,0	0,01	0,25	0,54	10000	10,04	24	3	{15185, 22080, 32028}	0,999	1,000	0,999	0,999
6	1,0	0,10	0,10	0,80	1000	8,85	8	3	{15079, 22114, 32164}	1,000	1,000	1,000	1,000
7	1,0	1,00	0,90	0,01	100	9,39	10	2	{15909, 32839}	0,923	0,667	1,000	0,800
8	1,0	10,00	0,75	0,27	10	8,87	27	3	{15076, 22183, 32093}	0,999	1,000	0,999	0,999
9	10,0	0,01	0,75	0,80	100	7,96	4	3	{15026, 22030, 32026}	1,000	1,000	1,000	1,000
10	10,0	0,10	0,90	0,54	10	7,94	4	3	{15103, 22099, 32178}	1,000	1,000	1,000	1,000
11	10,0	1,00	0,10	0,27	10000	8,82	3	2	{15138, 32754}	0,833	0,667	1,000	0,800
12	10,0	10,00	0,25	0,01	1000	8,41	3	3	{15382, 22573, 32302}	1,000	1,000	1,000	1,000
13	100,0	0,01	0,90	0,27	1000	8,23	1	1	{32993}	0,500	0,333	1,000	0,500
14	100,0	0,10	0,75	0,01	10000	8,55	3	3	{15387, 22389, 32239}	1,000	1,000	1,000	1,000
15	100,0	1,00	0,25	0,80	10	8,93	2	1	{15048}	0,600	0,333	1,000	0,500
16	100,0	10,00	0,10	0,54	100	8,16	1	1	{15460}	0,500	0,333	1,000	0,500

Tabla 9.6: Resultados experimentales para el *data stream 3*.

Test	λ	δ	ϵ	ϕ	N	t	CD_{det}	CD_{true}	n_{CD}	Prec.	Recall	Espec.	F -score
1	0,1	0,01	0,10	0,01	10	19,46	19	1	{15546}	0,909	0,333	1,000	0,500
2	0,1	0,10	0,25	0,27	100	17,51	230	3	{15140, 22020, 32100}	0,994	1,000	0,994	0,997
3	0,1	1,00	0,75	0,54	1000	21,81	719	3	{15034, 22042, 32001}	0,981	1,000	0,981	0,990
4	0,1	10,00	0,90	0,80	10000	15,91	264	3	{15055, 32031, 32095}	0,993	1,000	0,993	0,996
5	1,0	0,01	0,25	0,54	10000	9,16	24	3	{15079, 22047, 32077}	0,999	1,000	0,999	0,999
6	1,0	0,10	0,10	0,80	1000	9,13	9	3	{15070, 22105, 32155}	1,000	1,000	1,000	1,000
7	1,0	1,00	0,90	0,01	100	10,35	7	1	{32928}	0,800	0,333	1,000	0,500
8	1,0	10,00	0,75	0,27	10	9,42	19	3	{15089, 22186, 32078}	1,000	1,000	1,000	1,000
9	10,0	0,01	0,75	0,80	100	8,60	3	3	{15142, 22040, 32159}	1,000	1,000	1,000	1,000
10	10,0	0,10	0,90	0,54	10	8,64	4	3	{15170, 22455, 32121}	1,000	1,000	1,000	1,000
11	10,0	1,00	0,10	0,27	10000	14,37	2	2	{15897, 22503}	0,800	0,667	1,000	0,800
12	10,0	10,00	0,25	0,01	1000	9,19	3	3	{15382, 22643, 32372}	1,000	1,000	1,000	1,000
13	100,0	0,01	0,90	0,27	1000	9,02	0	0	-	NaN	NaN	1,000	NaN
14	100,0	0,10	0,75	0,01	10000	9,70	0	0	-	NaN	NaN	1,000	NaN
15	100,0	1,00	0,25	0,80	10	8,73	0	0	-	NaN	NaN	1,000	NaN
16	100,0	10,00	0,10	0,54	100	9,01	0	0	-	NaN	NaN	1,000	NaN

La Tabla 9.5 muestra los resultados para el *data stream 2*. Para estos resultados la situación es ligeramente diferente: 62,5 % de los experimentos tienen un *recall* del 100 % y solo el 50 % de ellos tienen una precisión del 100 % (experimentos 6, 9, 10, 12 y 14), lo que significa que menos combinaciones de parámetros son capaces de detectar los *concept drift* predefinidos. Estos resultados muestran que el algoritmo es sensible a la diferencia entre los GMM debido a la desviación y el ruido definidos en la Tabla 9.1, detectados como *concept drift*.

En el experimento con el *data stream 3* (Tabla 9.6) los resultados son más críticos debido a que algunas combinaciones de parámetros no pueden detectar ningún *concept drift*, es decir, los experimentos 13 a 16. En este caso, el 56 % de los experimentos tiene una sensibilidad del 100 % y de estos, únicamente el 55 % de ellos con una precisión del 100 %; concretamente en los experimentos 6, 8 a 10 y 12. Para este *data stream* los experimentos que no pudieron

detectar ningún *concept drift* se diferencian en el parámetro λ , que tiene el nivel más alto (100). Como análisis preliminar para la selección de parámetros, niveles elevados de λ no son óptimos cuando los cambios de GMM (*concept drift*) son pequeños.

Con respecto al tiempo de procesamiento, los experimentos 1, 2, 3 y 4 con $\lambda = 0,1$ de cualquier *data stream* tienen el valor más alto, pero no siempre dan garantías de la detección del *concept drift*. Como idea preliminar, $\lambda > 1$ puede dar resultados similares con un 50% del tiempo de procesamiento. Otros experimentos necesitan alrededor de 7,21 s para procesar 37000 instancias, independientemente del *data stream* utilizado.

En resumen, las combinaciones de parámetros 9, 10 y 12 son las únicas combinaciones capaces de detectar *concept drift* en todos los *data streams*, lo que significa que es posible usar parámetros óptimos en el algoritmo propuesto. Es interesante observar que estas combinaciones tienen el mismo $\lambda = 10$, dando algunas pistas para entender la magnitud óptima para este parámetro.

Estos son resultados intuitivos que pueden ayudar a comprender la influencia del parámetro en el rendimiento del algoritmo. Como resultado, en la Sección 9.4 se presenta un estudio completo que utiliza los resultados experimentales para construir un modelo supervisado capaz de seleccionar parámetros con el mejor rendimiento.

9.4. Selección de parámetros para un alto rendimiento

A partir de los resultados de experimentación mostrados en la sección anterior es necesario encontrar la selección óptima de parámetros λ , δ , ϵ , ϕ y N para definir las características comunes subyacentes entre las combinaciones de parámetros 9, 10 y 12. En consecuencia, aprovechando la eficiencia del algoritmo y su capacidad para obtener un buen rendimiento con pequeñas cantidades de datos, se utiliza el clasificador *naïve Bayes* (Minsky, 1961) idéntico al usado en el Capítulo 6.

De esta manera, se diseñan tres clasificadores para cada figura de mérito: precisión, *recall* y especificidad, tomadas como las variables de clase. Las variables predictoras son los cinco parámetros del GMM: λ , δ , ϵ , ϕ y N .

Debido a que las variables de clase son continuas los valores de cada experimento se discretizan en tres grupos diferentes:

- *Bajo*, para valores menores a 50%.
- *Medio*, para valores que son mayores que 50% y menores que 99%.
- *Alto*, para valores mayores al 99%.

Por lo tanto, el modelo *naïve Bayes* se define por

$$c^* = \arg \max_c p(c|\mathbf{par}) = \arg \max_c p(c) \prod_{i=1}^{|\mathbf{par}|} f(\text{par}_i|c)$$

Tabla 9.7: Valores máximos de densidad $f(\text{par}_i|C = \text{Alto})$ para c que denota precisión (segunda columna), *recall* (tercera columna) y especificidad (cuarta columna) considerando todos los experimentos mostrados en las Tablas 9.4, 9.5 y 9.6 fusionados en un *data stream*.

Parámetro	Precisión	<i>Recall</i>	Especificidad
ϕ	0,0002	0,0002	0,0001
ϵ	0,0002	0,0002	0,0002
δ	0,0013	0,0013	0,0011
λ	0,0079	0,0076	0,0101
N	0,9904	0,9907	0,9885

Tabla 9.8: Probabilidad para cada figura de mérito.

Figura de mérito	<i>Alto</i>	<i>Medio</i>	<i>Bajo</i>	Precisión de la predicción
Precisión	0,63	0,25	0,12	58,3 %
<i>Recall</i>	0,69	0,10	0,22	68,8 %
Especificidad	0,92	0,08	0,00	93,8 %

donde c^* es el máximo *a posteriori* o MAP y $\text{par} = (\lambda, \delta, \epsilon, \phi, N)$, y C es la variable clase de cada figura de mérito descrita anteriormente.

9.4.1. Clasificación de parámetros por relevancia

El primer paso en el análisis de resultados del experimento con el *naïve Bayes* es clasificar la relevancia de cada parámetro para cada figura de mérito. Los datos experimentales de la Sección 9.3.3 se incorporan al modelo y las probabilidades condicionadas de cada medida de mérito para el valor *Alto* de la variable clase se muestran en la Tabla 9.7. En la Tabla 9.8 se puede encontrar la probabilidad de que la clase (precisión, *recall* y especificidad) tome valores altos, medios y bajos, junto con la precisión del modelo *naïve Bayes* para los valores máximos de los parámetros de la Tabla 9.7. De acuerdo a esto, se determina que tanto la precisión, como el *recall* y la especificidad tienen altas probabilidades de tomar valores altos.

De la Tabla 9.7 el parámetro más relevante, con la probabilidad condicionada máxima para cifras altas de mérito (precisión, *recall* y especificidad) es N . Los parámetros δ y λ del test de Page–Hinkley (Sección 9.2.3) son los segundos parámetros más relevantes y los relacionados con las cotas de Chernoff (Sección 9.2.4) ϕ y ϵ son los menos relevantes. Al analizar estos resultados existe una fuerte dependencia del algoritmo con el tamaño de la ventana de entrenamiento para poder aumentar el rendimiento. Sin embargo, el parámetro N es el más relevante en términos de tiempo de entrenamiento, ya que el algoritmo tiene que esperar hasta que la ventana se llene para continuar con el análisis del *data stream*. Adicionalmente, a grandes tamaños de N se puede causar un ajuste excesivo, donde los *concept drift* podrían enmascarse durante el entrenamiento.

Estos resultados permiten concluir que son necesarios conocimientos previos sobre los

Tabla 9.9: Media y desviación estándar de cada parámetro dada la cifra de mérito en su mejor valor estimado según el modelo *naïve Bayes*

Parámetro	Precisión	Sensibilidad	Especificidad
λ	$22,6 \pm 38,7$	$20,6 \pm 37,5$	$29,6 \pm 43,0$
δ	$3,65 \pm 4,79$	$3,41 \pm 4,63$	$2,89 \pm 4,34$
ϵ	$0,52 \pm 0,32$	$0,54 \pm 0,31$	$0,48 \pm 0,34$
ϕ	$0,46 \pm 0,29$	$0,46 \pm 0,28$	$0,39 \pm 0,30$
N	2837 ± 4310	2675 ± 4143	2896 ± 4347

Tabla 9.10: Clasificación con el modelo *naïve Bayes* y los parámetros óptimos

Prueba	λ	δ	ϵ	ϕ	N	Probabilidad para			Predicción <i>naïve Bayes</i>
						Bajo	Medio	Alto	
Precisión	22,6	3,65	0,52	0,46	2837	0	0	1	Alta
<i>Recall</i>	20,6	3,41	0,54	0,46	2675	0	0	1	Alta
Especificidad	29,6	2,89	0,48	0,39	2896	0	0	1	Alta

procesos analizados que permitan calibrar la existencia de *concept drift*. Por ejemplo, si se espera que el RUL para una herramienta sea 8 h y para un motor sea 2000 h , el tamaño de N debería ser menor para la herramienta.

De acuerdo con el papel de cada parámetro dentro del algoritmo de *clustering* dinámico, dado por la experimentación y el análisis posterior con *naïve Bayes*, la media y la desviación estándar para cada parámetro se dan en términos de la figura de mérito que se buscan optimizar. Por lo tanto, es necesario estimar la media y la desviación estándar de la distribución Gaussiana de cada parámetro para cada figura de mérito en su mejor valor, es decir, la media y la desviación estándar de la distribución de cualquier parámetro para que la figura de mérito adquiera un valor *Alto*. Un resumen de los valores de los parámetros para alta precisión, *recall* y especificidad se muestra en la Tabla 9.9.

De la Sección 9.3.3, donde se encontraron los experimentos más precisos (9, 10 y 12), se puede hacer una comparación con los parámetros óptimos descritos en la Tabla 9.9. Por lo tanto, casi todos los parámetros de los experimentos mostrados en la Sección 9.3.3 están dentro de los valores mostrados, lo que significa que es posible ajustar de forma óptima el algoritmo y controlar el tamaño del entrenamiento inicial.

Para verificar el rendimiento del modelo construido con los parámetros óptimos se diseña un experimento seleccionando los valores medios para cada parámetro (Tabla 9.9). Posteriormente, los parámetros se introducen en el modelo *naïve Bayes* para verificar el proceso de clasificación en términos de las categorías *Bajo*, *Medio* o *Alto* para cada figura de mérito con su correspondiente probabilidad *a posteriori*.

Los resultados del experimento se muestran en la Tabla 9.10, donde los parámetros seleccionados pueden predecir los valores más altos posibles relacionados con la figura de mérito correspondiente. De hecho, como se muestra en la tabla, la probabilidad *a posteriori* para la clasificación con *Alto* es 1.

A continuación, los parámetros se verifican utilizando los *data streams* predefinidos 1, 2 y

Tabla 9.11: Resultados para parámetros óptimos con *data stream 1*

Test	λ	δ	ϵ	ϕ	N	t	CD_{det}	CD_{true}	n_{CD}	Prec.	Recall	Espec.	F-score
Alta precisión	22,6	3,65	0,52	0,46	2837	8,83	3	3	{15135, 22107, 32054}	1,0	1,0	1,0	1,0
Alta sensibilidad	20,6	3,41	0,54	0,46	2675	8,53	3	3	{15130, 22030, 32030}	1,0	1,0	1,0	1,0
Alta especificidad	29,6	2,89	0,48	0,39	2896	9,94	3	3	{15100, 22061, 32114}	1,0	1,0	1,0	1,0

Tabla 9.12: Resultados para parámetros óptimos con *data stream 2*

Test	λ	δ	ϵ	ϕ	N	t	CD_{det}	CD_{true}	n_{CD}	Prec.	Recall	Espec.	F-score
Alta precisión	22,6	3,65	0,52	0,46	2837	8,52	3	3	{15123, 22070, 32082}	1,0	1,0	1,0	1,0
Alta sensibilidad	20,6	3,41	0,54	0,46	2675	9,57	3	3	{15122, 22115, 32079}	1,0	1,0	1,0	1,0
Alta especificidad	29,6	2,89	0,48	0,39	2896	7,85	3	3	{15091, 22051, 32072}	1,0	1,0	1,0	1,0

Tabla 9.13: Resultados para parámetros óptimos con *data stream 3*

Test	λ	δ	ϵ	ϕ	N	t	CD_{det}	CD_{true}	n_{CD}	Prec.	Recall	Espec.	F-score
Alta precisión	22,6	3,65	0,52	0,46	2837	8,83	3	3	{15088, 22114, 32059}	1,0	1,000	1,0	1,0
Alta sensibilidad	20,6	3,41	0,54	0,46	2675	8,54	2	2	{15088, 22038}	0,8	0,667	1,0	0,8
Alta especificidad	29,6	2,89	0,48	0,39	2896	7,62	3	3	{15076, 22036, 32166}	1,0	1,000	1,0	1,0

3. Los resultados se muestran en las Tablas 9.11, 9.12 y 9.13. Según esto, para el *data stream 1* y 2, se obtiene 100% de precisión, *recall* y especificidad. Es importante destacar que el mayor retraso en la detección es de 135, lo que significa que solo se necesitan 135 instancias para detectar el *concept drift*.

Sin embargo, para el *data stream 3* (Tabla 9.13) los parámetros de alta sensibilidad tienen problemas para detectar el último *concept drift* ubicado en la instancia 32000. Este último *concept drift* está relacionado con un ruido de distribución normal con una desviación estándar baja de 0,05 (Tabla 9.1), lo que explica las dificultades del algoritmo para detectar esos pequeños cambios con estos parámetros.

Como conclusión de esta sección se puede ver que los parámetros del algoritmo son sensibles al escenario analizado por lo que se necesita un ajuste la primera vez que se utiliza. Por lo tanto, como un procedimiento para ajustar los parámetros del algoritmo en función de la aplicación se utiliza un diseño de Taguchi L16 para analizar la respuesta y construir un modelo. En este caso, se utiliza un *naïve Bayes* para seleccionar los mejores parámetros para una alta precisión, *recall* y/o especificidad.

Otro aspecto es la capacidad del algoritmo para asignar las instancias a cada componente del modelo de mixturas (clúster) que es el objetivo final. La bondad de este proceso de asignación se analiza en la siguiente sección.

9.5. Medición del ajuste del modelo

Si no hay *concept drift*, el GDPC asigna una instancia a cada componente a través de la probabilidad *a posteriori* de que la instancia \mathbf{x}^i pertenezca al componente k de la mixtura, como se refleja en la Ecuación (4.3).

Con el objetivo de estudiar cómo se desenvuelve el GDPC con *data streams* que contienen *concept drifts* durante la asignación del clúster se mide la probabilidad posterior de que la

instancia \mathbf{x}^i pertenezca al componente k . Por lo tanto, es importante supervisar cómo el modelo asigna instancias a cada componente y cómo comienza a no ajustar debido a la caducidad del GMM actual.

Intuitivamente, se espera encontrar un mejor ajuste una vez se detecte el *concept drift* y se vuelva a calcular el modelo. Es decir, las probabilidades *a posteriori* de que una instancia pertenezca a cada componente k ($k = 1, \dots, K$) serán más similares entre ellas (parecido a distribución uniforme sobre los componentes), por lo que la asignación ya no es tan clara.

9.5.1. Aplicación del *Brier score*

Para evaluar el rendimiento de la asignación de instancias a clusters se utiliza el *Brier score* descrito en la Sección 3.4.2. Para este análisis, la puntuación es mejor cuanto más bajos son los valores, es decir, la probabilidad de asignación está más concentrada y menos distribuida entre los componentes. La ventana de tamaño N' , con N' obtenido de la Ecuación (9.3) de las cotas de Chernoff, se usará como la ventana de cálculo del *Brier score*.

Para estudiar cómo afecta el *concept drift* al *Brier score*, se utilizan los *data streams* definidos en las secciones anteriores.

9.5.2. Resultados de experimentación y análisis

El *Brier score* se monitoriza a lo largo de las tres secuencias de datos para analizar su comportamiento con respecto a cómo el algoritmo ajusta un mejor modelo después de la detección de un *concept drift*. Para estos experimentos se toman los parámetros de alta precisión obtenidos en la Sección 9.4 ($\lambda = 22,6$, $\delta = 3,65$, $\epsilon = 0,52$, $\phi = 0,46$, $N = 2837$) con los tres *data streams* definidos en la Tabla 9.2.

Los resultados de los experimentos se muestran en la Figura 9.2, en donde se puede apreciar que el GDPC es capaz de detectar los tres *concept drift*. En cada caso, cuando el modelo cambia (por ejemplo, en las instancias 15000, 22000 y 32000), el *Brier score* cambia su comportamiento, estabilizándose después de detectar el punto de inflexión del *concept drift*. Como se muestra en la Figura 9.2, el *Brier score* es estable en cada tramo de las diferentes variaciones descritas en la Tabla 9.1, lo que significa que el modelo está asignando constantemente, con la misma confianza, las instancias a componentes específicos de la mixtura, cambiando su comportamiento solo si se detecta un *concept drift*.

Por lo tanto, el *Brier score* permite evaluar la calidad del nuevo modelo detectado después de un *concept drift*, donde gracias al comportamiento dinámico del GDPC es posible ajustar el modelo durante el procesamiento de la transmisión de datos. Esto permite garantizar una alta calidad del *clustering*, teniendo en cuenta los cambios en el *data streams* a lo largo del tiempo. Por ejemplo, aquellos debidos a la degradación.

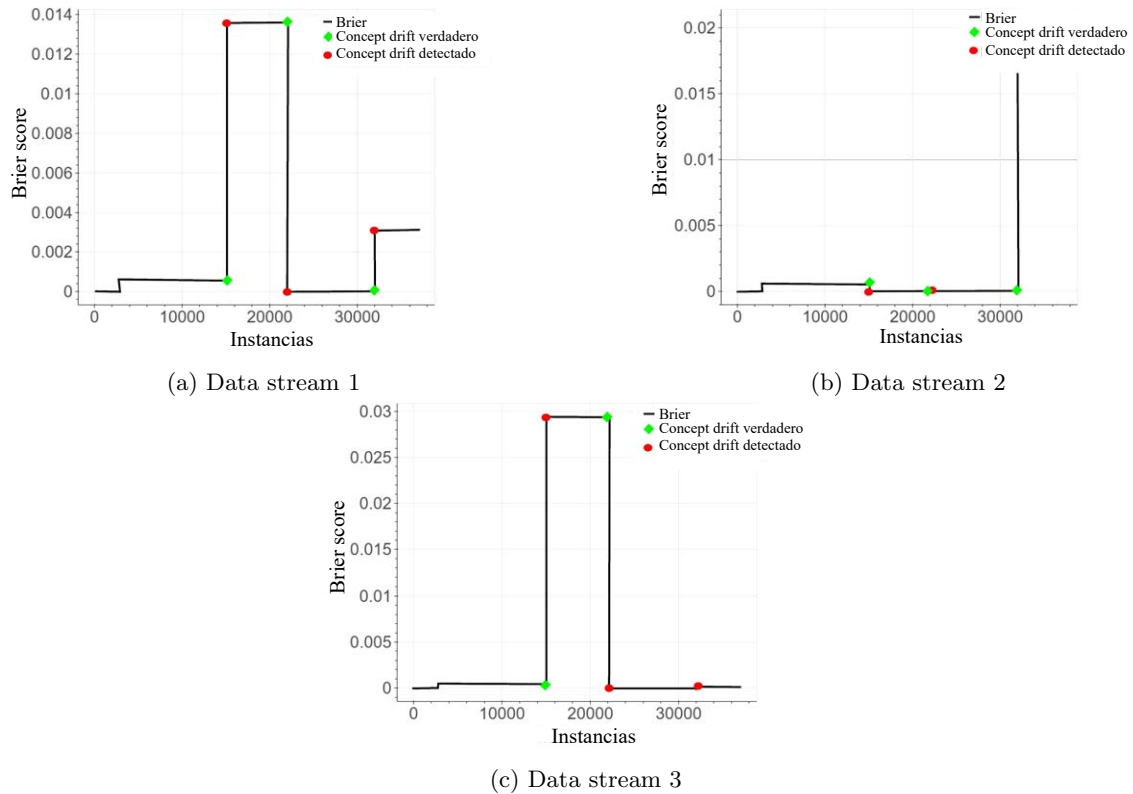


Figura 9.2: Evolución del *Brier score* para cada *data streams*.

9.6. Comparación del algoritmo GDPC con algoritmos del estado del arte

Usando los parámetros óptimos definidos para alta precisión, etiquetados como λ , δ , ϵ , ϕ y N (Tabla 9.10) y el *data stream* 3 (Tabla 9.2), se compara el GDPC con algunos algoritmos de *clustering* de *data streams* conocidos descritos en la Sección 5.2: CluStream (basado en K -medias), D-Stream (basado en DBSCAN) y ClusTree (algoritmo híbrido con K -medias y DBSCAN). Para realizar esta comparativa se han usado las versiones de estos algoritmos disponibles en la plataforma *Massive Online Analysis* (MOA) (Bifet et al., 2010).

El análisis se ha realizado en el *data streams* 3 sintético, con tres *concept drifts* localizados en las instancias 15000, 22000 y 32000. Las etiquetas reales del clúster de cada instancia se han generado desde el *data stream* sintético. A partir de los resultados del *clustering*, el *recall* y el *F-score* se han calculado en lotes de 1000 instancias, para luego representarlas y verificar su evolución a lo largo del *data streams*. Los resultados del análisis se muestran en la Figura 9.3.

Al observar la Figura 9.3, el rendimiento de D-Stream es comparable al GDPC en términos de *recall* durante la primera parte del *data stream*. Después del primer *concept drift* el rendimiento cae por debajo de los otros algoritmos. Esto significa que D-Stream no tiene capacidades robustas para recalculer el modelo después de un *concept drift*. En términos de

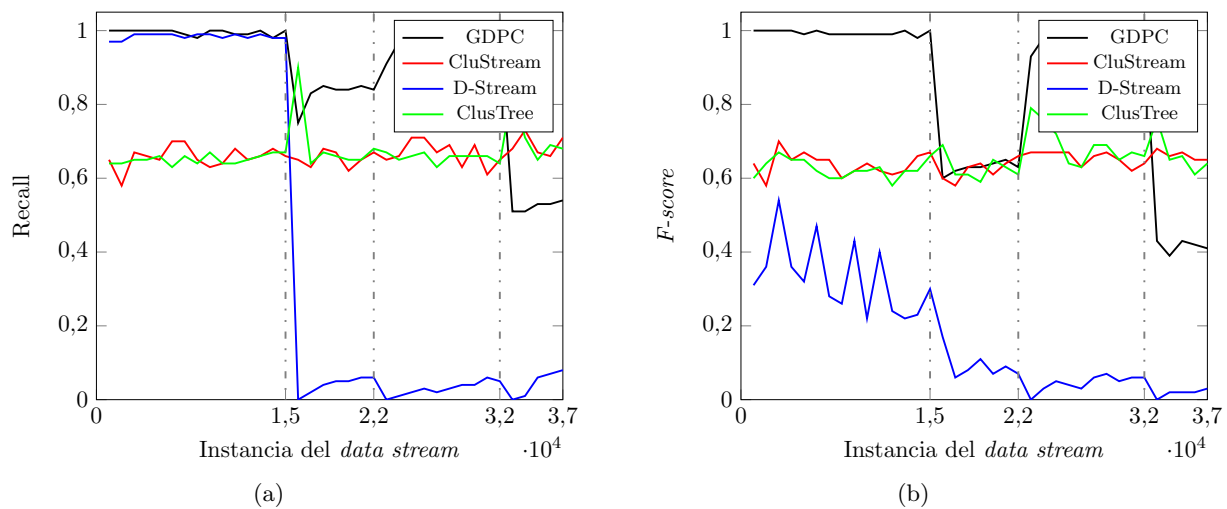


Figura 9.3: (a) *Recall* y (b) *F-score* para el *clustering* del *data stream*.

F-score, el valor de D-Stream está por debajo del resto a lo largo del todo el *data stream*.

CluStream y ClusTree se han comportado de manera similar, mostrando tasas bajas de verdaderos positivos en todo el *data stream*. Sin embargo, el ClusTree reacciona cuando aparece el *concept drift*, aunque no detecta el que está ubicado en la instancia 15000. Ambos algoritmos se han comportado mejor durante el último tramo del *data stream*, aunque no está claro si este comportamiento se debe a que el algoritmo no es sensible a la presencia de *concept drift*.

Por otro lado, el algoritmo GDPC sí reacciona al *concept drift*, adaptándose en tiempo real, lo que le permite mantener una tasa de detección de verdaderos positivos elevada. Adicionalmente, el algoritmo detecta perfectamente el *concept drift* ubicado en la instancia 15000, aunque se observa que su rendimiento disminuye en algunos casos. Una vez se calcula un nuevo modelo de GMM, los ratios *recall* y *F-score* aumentan, es decir, el algoritmo se adapta para tener en cuenta los nuevos datos del *data stream*. También se detecta el siguiente *concept drift* en la instancia 22000, y se lanza un nuevo modelo GMM. Esto aumenta aún más la tasa de verdaderos positivos a casi uno. También se detecta el último *concept drift*, instancia 32000, volviendo a lanzar el GMM. En este caso, el nuevo ajuste hace que el GDPC tenga un rendimiento algo inferior a los algoritmos CluStream y ClusTree.

La Tabla 9.14 proporciona un resumen de estos resultados.

9.7. *Clustering* dinámico probabilístico aplicado a *data streams* del banco de pruebas

9.7.1. Adquisición de datos

Para analizar el rendimiento del algoritmo con *data streams* se utiliza un banco de pruebas experimental mostrado en la Figura 7.4. Para simular los *concept drift* producidos por una máquina el banco de pruebas se programa para realizar un movimiento de servomotor

Tabla 9.14: Comparativa entre algoritmos.

Algoritmo	Tipo	Detección de <i>concept drift</i>	Tiempo real adaptativo
GDPC	Probabilidad	Si	Si
CluStream	Distancia	No	Si
D-Stream	Densidad	No	No
ClusTree	Híbrido	No	Si

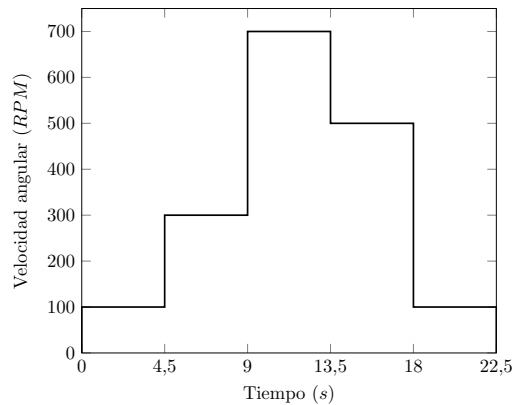


Figura 9.4: Perfil de velocidad angular para el banco de pruebas.

con diferentes pasos de velocidad angular, expresados en revoluciones por minuto (*RPM*), recreando de esta manera diferentes estados o fases. Por lo tanto, el cambio en la velocidad angular definido como un *concept drift* se ubica cada 4500 instancias, totalizando 25 *concept drifts*.

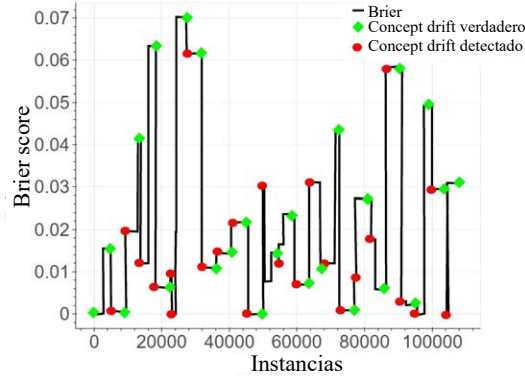
El perfil de velocidades programado se muestra en la Figura 9.4, donde el servomotor gira a 100, 300, 700, 500 y 100 *RPM*, 4,5 s, respectivamente. Este ciclo se repite cinco veces con la misma estructura. De cada ciclo se adquieren cuatro variables: tiempo, potencia activa, velocidad angular y par. Cuando cambia la velocidad angular las variables del servomotor también varían simulando una alteración en el comportamiento denotado como un *concept drift*. En consecuencia, se ha generado un *data stream* con 117822 instancias con una tasa de muestreo de 100 ms.

9.7.2. Resultados y discusión

Usando los parámetros del algoritmo para alta precisión, alto *recall* y alta especificidad de la Tabla 9.9, se analiza el *data stream*. Los resultados se muestran en la Tabla 9.15. El algoritmo muestra un alto grado de precisión para la detección del *concept drift* (98,7%). En este caso, los parámetros de alta precisión y *recall* tienen el mejor rendimiento del algoritmo, logrando 98,7% de precisión y *recall* del 96%, lo que significa que casi todas las detecciones son verdaderos positivos. Sin embargo, hay aproximadamente el doble de la cantidad de

Tabla 9.15: Resultados experimentales para el *data stream* del banco de pruebas.

Test	λ	δ	ϵ	ϕ	N	t	CD_{det}	CD_{true}	n_{CD}	Prec.	Recall	Espec.	F-score
Alta precisión	22,6	3,65	0,52	0,46	2837	21,07	50	24	{4770, 9215, 13706, 18237, 22775, 22877, 27467...}	0,987	0,96	1,0	0,98
Alta sensibilidad	20,6	3,41	0,54	0,46	2675	21,69	56	24	{4780, 9239, 13689, 18236, 22686, 22836, 27378...}	0,987	0,96	1,0	0,98
Alta especificidad	29,6	2,89	0,48	0,39	2896	21,13	49	24	{4696, 9198, 13819, 18199, 22699, 22881, 27437...}	0,986	0,96	1,0	0,98

Figura 9.5: Análisis con el *Brier score* para el *data stream* del banco de pruebas.

instancias detectadas como puntos de inflexión para el *concept drift*. El trabajo posterior relacionado con el desarrollo del GDPC debe orientarse a encontrar una manera efectiva de reducir esa detección en entornos reales y/o desconocidos. Este tema será tratado más en profundidad en el Capítulo 10.

El tiempo t que se requiere para procesar el *data stream* completo (es decir, el tiempo requerido una vez llega la nueva instancia), de 117822 instancias y 25 *concept drifts* es de 22 s, lo que significa que el algoritmo necesita un promedio de 185 μ s por instancia. Este total incluye el tiempo necesario para volver a calcular el modelo cuando se encuentra un *concept drift*. En este escenario particular, donde la monitorización se realiza a 100 ms, se logra un comportamiento en tiempo real del algoritmo.

Por otro lado, tomando los parámetros para una alta precisión, se monitoriza el *Brier score* (Figura 9.5) para verificar cómo se detecta el *concept drifts* y luego se ajusta el modelo disminuyendo el valor de la puntuación, por lo que hay una alta confianza en la asignación de instancias a cada componente.

De los resultados, se puede ver que el GDPC tiene la capacidad de clusterizar un *data stream* basado en un modelo de mixturas de Gaussianas, con una precisión razonable cercana al 99%, realizando su tarea en el tiempo requerido. Esto se ha logrado con solo el 2% del total de instancias disponibles. Se han detectado en su totalidad los 25 *concept drift* inducidos junto a otros cambios. Estas detecciones podrían alertar a un sistema de monitorización de la existencia de cambios temporales dentro de un proceso, probablemente debidas a la degradación o algún otro comportamiento anómalo. El conocimiento obtenido a partir de estos *concept drift* y su evolución a lo largo del tiempo puede ser utilizado para un análisis adicional relacionado con el RUL mencionado anteriormente.

9.8. Conclusiones

Las principales conclusiones de este capítulo son:

- Se ha desarrollado un nuevo algoritmo denominado GDPC (*clustering* probabilístico dinámico basado en mixturas de Gaussianas), el cual tiene la capacidad de analizar un *data stream* con una necesidad mínima de datos históricos en tiempo real. También es capaz de detectar *concept drifts* y ajustar el modelo, con la garantía de mantener una representación óptima de los datos.
- El *concept drift* más fácilmente detectado en un *data stream* es el primero, independientemente de su tipo. Desde este punto de inflexión en adelante los siguientes cambios representan un desafío para su detección. Este comportamiento es causado por el tamaño de ventana utilizado por las cotas de Chernoff, que tiende a ser más pequeño que el primer conjunto de entrenamiento. Esto se debe a que el algoritmo siempre usa la ventana más pequeña posible. Este efecto debe controlarse utilizando valores mínimos de ϵ y ϕ para maximizar el tamaño de la ventana.
- Los parámetros del algoritmo, especialmente el tamaño de la ventana de entrenamiento N , tienen un efecto crítico en el resultado. Básicamente, en su capacidad para reconocer un *concept drift*, pero también en su precisión. Una ligera variación en algunos de los parámetros puede causar grandes cambios en los valores de mérito.
- La relación existente entre los parámetros del algoritmo no permite una representación simple con herramientas tradicionales para el análisis experimental. Es por eso que un clasificador *naïve Bayes* puede ayudar a encontrar una buena configuración de parámetros de forma automática.
- Independientemente del modelo utilizado, el tamaño de la ventana de entrenamiento es el factor más crítico y los parámetros de las cotas de Chernoff son los factores menos críticos.
- El rendimiento del algoritmo está relacionado con la naturaleza del *concept drift*. Los cambios pequeños, como la introducción de ruido Gaussiano con una desviación estándar baja, son difíciles de detectar.
- El *clustering* con modelos de mixturas de Gaussianas es una herramienta realmente potente que puede brindar información útil para el análisis posterior, como el *Brier score*, los parámetros de las mixturas y la probabilidad de asignación al clúster, entre otros.
- El *Brier score* es una herramienta adecuada para evaluar la capacidad de los algoritmos de *clustering* dinámico en cuanto al ajuste de los modelos durante el análisis de un *data stream*. Si el nuevo modelo no puede reducir el *Brier score*, el rendimiento del algoritmo no es lo suficientemente bueno.

- La aplicabilidad en tiempo real del algoritmo a la industria es posible, ya que no requiere gran cantidad de datos y a su vez, requiere un tiempo de procesamiento reducido. Es decir, el algoritmo utiliza un tamaño pequeño de ventana de *data stream* que se adapta continuamente al mismo para aumentar la eficiencia. Desde esta ventana el algoritmo puede actualizar el modelo si se produce un cambio sin tomar un tiempo de procesamiento excesivo, como ocurre con el GMM estático tradicional. Esto representa un primer paso para aplicar de manera real una técnica de *clustering* basado en modelos probabilidad, en la que se necesitan resultados en el momento adecuado y el acceso a los grandes sistemas informáticos es limitado.

Como trabajo posterior relacionado con el desarrollo del algoritmo se podría mejorar la selección de parámetros con un clasificador más avanzado que el *naïve Bayes*, el cual no tiene en cuenta la relación entre las variables (parámetros). Además, se debe hacer un trabajo específico para optimizar la detección de pequeños cambios, como el ruido Gaussiano con una desviación estándar baja. Por otro lado, se debe trabajar para aumentar las características del algoritmo, como la estimación del número de componentes K , en función de los datos de entrenamiento, donde podrían surgir nuevos componentes o eliminarse durante el *data stream*. Además, los diferentes modelos de mixturas de Gaussianas producidos durante el *concept drift* deben estudiarse para diferenciar las magnitudes de estas derivas, es decir, medir la distancia entre los modelos. Estos dos últimos puntos son especialmente útiles cuando el algoritmo se utiliza en entornos reales sin supervisión, donde se desconoce el punto de inflexión del *concept drift*. Esto se tratará en el Capítulo 10.

Capítulo 10

Clustering dinámico basado en modelos de mixturas Gaussianas para entornos desconocidos

De acuerdo a lo comentado en el Capítulo 9, el GDPC tiene la capacidad de detectar *concept drifts* en *data streams*. Sin embargo, la configuración de sus parámetros debe ser hecha de manera manual con el apoyo del conocimiento previo del entorno. Algunas veces, este conocimiento previo puede carecer de la experiencia suficiente que permita valorar de una manera correcta estos parámetros.

Por esta razón, uno de los dos puntos más importantes a desarrollar es la selección automática del número de componentes, que representa uno de los mayores retos, ya que aún teniendo el conocimiento previo, este número podría variar debido a los *concept drifts*, asociado por ejemplo, con la aparición de nuevos estados dentro del elemento analizados. Otro punto importante en el efecto que puede generar en el rendimiento del algoritmo ante comportamientos no lineales dentro del *data stream*, algo que puede encontrarse comúnmente dentro de la industria.

Por esta razón, a continuación se propone una versión del GDPC que permite hacer frente a este tipo de problemas. Esta capítulo se ha publicado en ([Diaz-Rozo et al., 2019](#)).

10.1. Introducción

El algoritmo GDPC propuesto en el Capítulo 9 puede explotar el conocimiento obtenido de los procesos de ingeniería a diferentes niveles y tareas con *data streams*, donde el algoritmo EM se inicia solo si se detecta un *concept drift*. Sin embargo, GDPC trabaja con parámetros como el número de componentes de la muestra que deben estimarse mediante un entrenamiento preliminar con datos reales y/o de acuerdo a la experiencia de los usuarios en el proceso analizado. Por ejemplo, si un ciclo de mecanizado tiene tres estados que comprenden *detenido*, *ciclo en vacío* y *mecanizando*, y el problema implica determinar el patrón de cada estado,

entonces el usuario podría configurar el algoritmo con tres componentes. Esto podría no ser fiable si existiesen otros patrones interesantes a explorar. Adicionalmente, el algoritmo de actualización del modelo puede ser inestable después de un *concept drift* porque los datos no estacionarios todavía podrían experimentar cambios de relación entre la entrada y la salida. Esta situación desencadena nuevas actualizaciones tal vez innecesarias, reduciendo el rendimiento del algoritmo al requerir más cálculos del algoritmo EM, y por otro lado, aumentando el número de falsos positivos (es decir, *concept drift* falsos).

El objetivo principal de este capítulo es proponer una mejora llamada GDPC+ sobre el algoritmo de GDPC para superar estas limitaciones. Estas características son descritas a continuación:

- Reducir la necesidad de tener conocimiento previo del usuario y evitar realizar pruebas costosas. La cantidad de datos de entrenamiento N va a ser el único parámetro a controlar por el usuarios.
- Minimizar la inestabilidad del GDPC causada por datos no estacionarios durante el *concept drift*.

Por lo tanto, el GDPC+ puede operar en escenarios completamente desconocidos con un número reducido de falsos positivos y una mejor aplicabilidad en escenarios reales.

10.2. Metodología del algoritmo GDPC+

Según el Capítulo 9, la secuenciación del algoritmo GDPC se basa en seis pasos, como se muestra en la Figura 10.1a, y se encuentran descritos en la Sección 9.2. Sin embargo, este esquema hace dos suposiciones que conducen a problemas importantes cuando el GDPC está operando con un *data stream* parcial o completamente desconocido, principal objetivo de los algoritmos no supervisados como el *clustering*.

El primer supuesto se relaciona con los pasos 1 y 4 de la secuenciación (Figura 10.1a), donde cada instancia se asigna a los K componentes con un valor constante para todo el algoritmo durante el procesamiento del *data stream*. En este caso, si los datos están representados por un GMM, el número de componentes que mejor se ajusta a los datos podría cambiar con el tiempo debido a fenómenos de unión y división de componentes que pueden existir durante el proceso (Spiliopoulou et al., 2006). Adicionalmente, es necesaria una estimación inicial del número de componentes por parte del usuario que debe tener un conocimiento previo del proceso bajo análisis. Sin embargo, es posible usar medidas como el BIC (Sección 3.5) para determinar dinámicamente el número de componentes cada vez que se actualiza un GMM (Schwarz, 1978). Esta propuesta de modificación se muestra en la Figura 10.1b para los pasos 2 y 8, que se emplean para la estimación de componentes dinámicos.

El segundo supuesto se relaciona con el paso 6 (Figura 10.1a), donde se detecta un *concept drift* y se actualiza el modelo. Cuando esto ocurre el modelo se vuelve a estimar desde la última ventana de datos. Un cambio en el patrón de los datos produce un *concept drift*, por

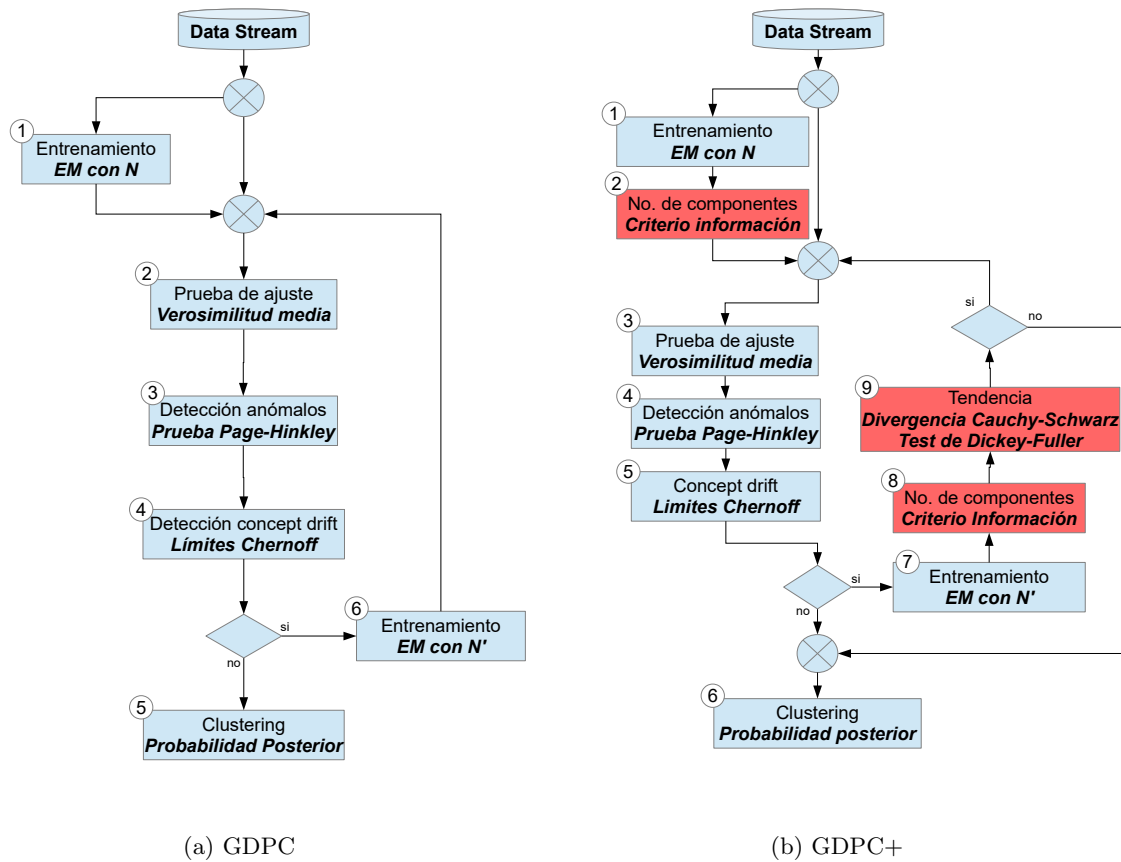


Figura 10.1: Esquemas de los algoritmos GDPC y GDPC+. Las nuevas características en GDPC+ se muestran en cuadros rojos.

lo que la última ventana y la siguiente podrían incluir datos no estacionarios y otros datos con ruido que se incorporan erróneamente al proceso de actualización. Dependiendo de la naturaleza del *concept drift*, este proceso de actualización podría ser inestable en el tiempo, lo que requeriría el lanzamiento de una actualización del modelo en cada iteración posterior hasta que los nuevos datos alcancen un estado estacionario.

Por lo tanto, se propone un sistema de monitorización para controlar la inestabilidad del algoritmo. Este sistema de monitorización mide la divergencia entre dos GMM (el GMM antes del *concept drift* y el GMM después de la actualización del modelo debido al *concept drift*) usando la divergencia de Cauchy-Schwarz descrita por [Kampa et al. \(2011\)](#) y su variación a lo largo del tiempo usando el test de Dickey-Fuller ([Dickey y Fuller, 1979](#)). Por lo tanto, se agrega en la Figura 10.1b el paso 9, donde un cambio en la tendencia en la medición de la divergencia determina el proceso de actualización del modelo.

Las descripciones detalladas de cada nuevo paso se dan a continuación.

10.2.1. Estimación dinámica del número de componentes

Del algoritmo GDPC, la densidad en el k -ésimo componente es $f_k(\mathbf{x}; \boldsymbol{\theta}_k)$ y el modelo GMM fue descrito en la Sección 4.5. De esta manera, en el algoritmo GDPC+, para encontrar K se busca el valor BIC mínimo, definido en la Sección 3.5, dentro de un rango de posibles valores de K , donde se prefiere el valor BIC* más bajo para definir un GMM disperso. Dicho valor se calcula de la siguiente manera:

$$\text{BIC}^* = \min_{\hat{\Psi}} \left(-2 \log L(\hat{\Psi}) + \dim(\hat{\Psi}) \log N \right).$$

10.2.2. Estabilización transitoria del *concept drift*

En este caso, la mejora propuesta para el algoritmo se basa en la hipótesis de que un *concept drift* ocasionaría una gran divergencia entre el GMM actual y el GMM calculado después de detectarse. Para medir esta divergencia entre dos mixturas de Gaussianas con diferentes parámetros $\hat{\Psi}$ y no necesariamente el mismo número de componentes, se utiliza la divergencia de Cauchy–Schwarz que tiene una expresión cerrada (Kampa et al., 2011). Por lo tanto, partiendo de dos GMMs:

$$p(\mathbf{x}; \Psi) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (10.1)$$

y

$$q(\mathbf{x}; \Psi) = \sum_{m=1}^M \tau_m \mathcal{N}(\mathbf{x}; \boldsymbol{\nu}_m, \boldsymbol{\Lambda}_m), \quad (10.2)$$

donde K es el número de componentes y $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ son los parámetros para el GMM $p(\mathbf{x}; \Psi)$, y M es el número de componentes y $\tau_m, \boldsymbol{\nu}_m, \boldsymbol{\Lambda}_m$ son los parámetros para el GMM $q(\mathbf{x}; \Psi)$. Cada componente en el GMM correspondiente está dado por:

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}_k|^{1/2}} \exp \left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right) \quad (10.3)$$

y

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\nu}_m, \boldsymbol{\Lambda}_m) = \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Lambda}_m|^{1/2}} \exp \left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\nu}_m)^T \boldsymbol{\Lambda}_m^{-1} (\mathbf{x} - \boldsymbol{\nu}_m) \right), \quad (10.4)$$

con $\mathbf{x} \in \mathbb{R}^d$.

La divergencia de Cauchy–Schwarz entre $p(\mathbf{x}; \Psi)$ y $q(\mathbf{x}; \Psi)$ viene dada por:

$$D_{CS}(p, q) = -\log \left(\frac{\int p(\mathbf{x}; \Psi) q(\mathbf{x}; \Psi) d\mathbf{x}}{\sqrt{\int p^2(\mathbf{x}; \Psi) d\mathbf{x} \int q^2(\mathbf{x}; \Psi) d\mathbf{x}}} \right),$$

que puede reescribirse como:

$$D_{CS}(p, q) = -\log \left(\int p(\mathbf{x}; \Psi)q(\mathbf{x}; \Psi)d\mathbf{x} \right) + \frac{1}{2} \log \left(\int p^2(\mathbf{x}; \Psi)d\mathbf{x} \right) + \frac{1}{2} \log \left(\int q^2(\mathbf{x}; \Psi)d\mathbf{x} \right).$$

Resolviendo las integrales y utilizando las Ecuaciones (10.1) – (10.4), la expresión en forma cerrada está dada por:

$$\begin{aligned} D_{CS}(p, q) = & -\log \left(\sum_{k=1}^K \sum_{m=1}^M \pi_k \tau_m z_{km} \right) \\ & + \frac{1}{2} \log \left(\sum_{k=1}^K \frac{\pi_k^2}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} + 2 \sum_{k=1}^K \sum_{k' < k} \pi_k \pi_{k'} z_{kk'} \right) \\ & + \frac{1}{2} \log \left(\sum_{m=1}^M \frac{\tau_m^2}{(2\pi)^{d/2} |\Lambda_m|^{1/2}} + 2 \sum_{m=1}^M \sum_{m' < m} \tau_m \tau_{m'} z_{mm'} \right), \end{aligned} \quad (10.5)$$

donde

$$\begin{aligned} z_{km} &= \mathcal{N}(\boldsymbol{\mu}_k; \boldsymbol{\nu}_m, (\boldsymbol{\Sigma}_k + \boldsymbol{\Lambda}_m)), \\ z_{kk'} &= \mathcal{N}(\boldsymbol{\mu}_k; \boldsymbol{\mu}_{k'}, (\boldsymbol{\Sigma}_k + \boldsymbol{\Sigma}_{k'})), \text{ y} \\ z_{mm'} &= \mathcal{N}(\boldsymbol{\nu}_m; \boldsymbol{\nu}_{m'}, (\boldsymbol{\Lambda}_m + \boldsymbol{\Lambda}_{m'})). \end{aligned}$$

Después de medir la divergencia entre dos GMMs es importante valorar el tamaño de la misma. Así, [Mushtaq \(2011\)](#) propuso el test de Dickey–Fuller aumentado (ADF), que permite cumplir con este propósito. Esta prueba es un test de hipótesis generalmente usado para comprobar tendencias en datos macroeconómicos y determinar si los mismos son estacionarios o no. Los datos se definen como estacionarios cuando los parámetros de media y covarianza son invariantes en el tiempo durante un cierto periodo de tiempo o rezago.

De acuerdo con [Cheung y Lai \(1995\)](#), si $\Delta D_{CS_t}(p, q) = D_{CS_t}(p, q) - D_{CS_{t-1}}(p, q)$ es la variación en el tiempo t de la divergencia de Cauchy–Schwarz de dos GMMs p y q , el test ADF realiza primero la regresión:

$$\begin{aligned} \Delta D_{CS_t}(p, q) = & \alpha + \beta t + \gamma D_{CS_{t-1}}(p, q) \\ & + \delta_1 \Delta D_{CS_{t-1}}(p, q) + \cdots + \delta_{u-1} \Delta D_{CS_{t-u+1}}(p, q) \\ & + \epsilon_t \end{aligned} \quad (10.6)$$

donde α es una constante, β es el coeficiente de tiempo, ϵ_t es el error, u es el orden de rezago del proceso autorregresivo, δ_i son los coeficientes autoregresivos ($i = 1, \dots, u - 1$) y $\gamma \leq 0$ es el coeficiente a monitorizar, en el que se examina su negatividad a lo largo del tiempo t . Si $\gamma = 0$, entonces el valor de la divergencia $D_{CS_{t-1}}(p, q)$ no proporciona información relacionada

con su cambio en el tiempo t , y por lo tanto se define como estacionario. Por lo tanto, el test ADF se define de la siguiente manera:

$$\begin{cases} H_0 : \gamma = 0 \\ H_1 : \gamma < 0 \end{cases}$$

De la Ecuación (10.6), si $\gamma = 0$ significa esto que la variación $\Delta D_{CS_t}(p, q)$ es constante. En cambio, si $\gamma < 0$ existe un cambio de tendencia dado por $D_{CS_{t-1}}(p, q)$, significando esto que se rechaza H_0 . Este cambio de tendencia en la divergencia de Cauchy-Schwarz, $\Delta D_{CS_t}(p, q)$, significa que existe un proceso no es estacionario, y por lo tanto hay un gran cambio entre las distribuciones p y q . Es decir, el valor anterior $\Delta D_{CS_{t-1}}(p, q)$ es relevante, confirmando así un *concept drift* en este momento. De esta manera, monitorizar γ y proporcionar un nivel significación para rechazar la hipótesis nula evita actualizar el GMM durante periodos no estacionarios del *data stream* y filtrar así los efectos de las inestabilidades.

Además, se pueden estudiar diferentes comportamientos no estacionarios basados en la Ecuación (10.6):

- Si $\beta = 0$ y $\alpha = 0$, entonces el test ADF examina el comportamiento estacionario.
- Si $\beta = 0$ y $\alpha \neq 0$, entonces el test ADF examina el comportamiento estacionario con deriva.
- Si $\beta \neq 0$ y $\alpha \neq 0$, entonces el test ADF examina el comportamiento estacionario con deriva y tendencia temporal.

Para demostrar cómo estas nuevas características ayudan a mejorar el GDPC se evalúa en primer lugar su desempeño con *data streams* simulados, como se describe a continuación. Posteriormente (Sección 10.4) se aplica a un escenario real.

10.3. Evaluación del rendimiento del algoritmo GDPC+

El desempeño mejorado de GDPC+ en comparación con el algoritmo GDPC se evalúa en términos de diferentes medidas de mérito. Para esto se utiliza precisión, *recall*, especificidad y *F-score*. Se ha generado un conjunto de datos sintéticos que comprende 20 variables y 20000 instancias utilizando diferentes distribuciones con parámetros aleatorios. Para evaluar la detección del *concept drift* se introdujeron cuatro de éstos cambiando aleatoriamente los parámetros de distribución. El resultado marca la ubicación de los *concept drifts* en las instancias 6133, 6393, 9819 y 18400.

Se utilizaron, como en el Capítulo 9, las siguientes medidas para estimar los valores de mérito:

1. Instancias donde se detecta un *concept drift* (n_{CD}): el número de instancia donde se detecta *concept drift* dentro de un *data stream*.

Tabla 10.1: Resultados de rendimiento obtenidos con la selección de componentes estática filas con $K = 3$ hasta $K = 9$ y dinámica última fila

K	$Inst/s$	CD_{true}	n_{CD}	Prec. (%)	Recall (%)	Espec. (%)	$F-score$ (%)
3	1293,19	0	12	99,90	0,00	99,90	0,00
4	2238,38	0	104	99,50	0,00	99,50	0,00
5	676,44	4	383	98,10	100,00	98,10	99,00
6	1239,82	4	383	98,10	100,00	98,10	99,00
7	591,78	4	383	98,10	100,00	98,10	99,00
9	1014,21	4	383	98,10	100,00	98,10	99,00
Dinámico	358,12	4	383	98,10	100,00	98,10	99,00

2. *Concept drift* verdaderos (CD_{true}): La cantidad de *concept drift* verdaderos en comparando con sus ubicaciones $CD_{set} = \{6133, 6393, 9819, 18400\}$.

El desempeño se ha evaluado utilizando los valores de mérito derivados de la detección de *concept drift*, es decir, se comparan aquellos que son *concept drift* verdaderos y el total detectado por el GDPC+ para de ahí estimar precisión, *recall*, especificidad, y *F-score*.

10.3.1. Evaluación del método de selección de componentes dinámico

El número de componentes seleccionados se determina dinámicamente mediante el uso de la métrica BIC, como se describe en la Sección 10.2.1, y su rendimiento se analiza, a continuación, comparando con el GDPC (configuración de componentes estática) para diferentes valores de componentes de $K = 3, 4, 5, 6, 7, 9$ y una ventana temporal $N = 10$. El BIC se ha obtenido utilizando la función `bic()` de la biblioteca `Scikit-Learn` (Pedregosa et al., 2011).

Los resultados se muestran en la Tabla 10.1. Cabe señalar que la selección dinámica de componentes ha obtenido el mismo rendimiento que el análisis estático cuando K toma los valores 5, 6, 7 y 9. Sin embargo, la frecuencia de procesamiento (es decir, el número de instancias procesadas por segundo) ha sido más lenta debido a que el método propuesto pretende minimizar el valor BIC, mientras que la versión estática GDPC no necesita realizar una búsqueda debido a que está definido de antemano.

El análisis de la selección del modelo dinámico en términos de los cambios en el número de componentes muestra que los valores oscilan entre $K = 4$ y $K = 9$ (Figura 10.2a). Usando estos valores, se ha comenzado a detectar el *concept drift* verdadero con el GDPC, excepto cuando $K = 4$ (Tabla 10.1). Por lo tanto, cuando se trabaja con un *data stream* desconocido (es decir, no existe información disponible sobre la estructura de datos para ayudar a estimar el número de componentes) la selección dinámica de componentes podría obtener los mismos resultados que la configuración de los componentes antes de ejecutar el GDPC. Este es un resultado interesante porque la selección dinámica de componentes puede permitir que el algoritmo GDPC+ trabaje sin necesidad de estimaciones iniciales del número de componentes, proporcionados por un experto y con bajos costos computacionales de acuerdo con la frecuencia de procesamiento.

La Tabla 10.1 muestra que después de la detección del *concept drift*, la cantidad de

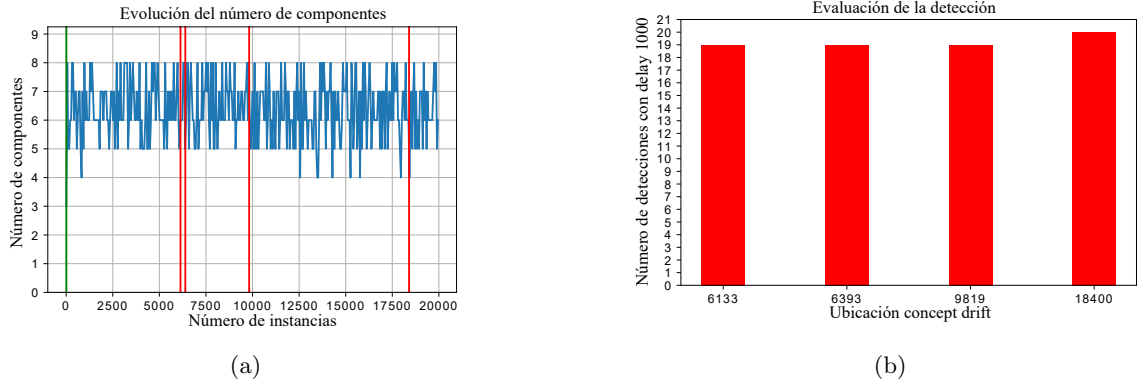


Figura 10.2: (a) Evolución de la selección dinámica de componentes basada en el *data stream*. Las líneas rojas indican la ubicación del *concept drift* real. (b) Número de *concept drifts* detectados con un retraso de 1000 instancias.

falsos positivos comienza a aumentar y el indicador de especificidad para el algoritmo GDPC disminuye de 99,90 % a 98,10 %, lo que podría llevar a rendimiento deficiente cuando se requiere el algoritmo para detectar el comportamiento de degradación, porque dispararía falsas alarmas continuamente al marcar cualquier cambio como un *concept drift*. Un análisis más profundo de este comportamiento muestra la detección de múltiples *concept drifts* en el rango entre CD_{set} y $CD_{set} + 1000$ instancias, donde se han señalado 19 detecciones de *concept drift* en la instancia número 6133, 6393, y 9819 y 20 detecciones en la instancia 18400 (Figura 10.2b).

Las múltiples detecciones se atribuyen a las inestabilidades que han ocurrido después de la detección del *concept drift*, donde el nuevo modelo ha utilizado datos no estacionarios para la actualización. Después del cálculo de un nuevo modelo las siguientes instancias provenientes del *data stream* también podrían tener valores inestables porque se ha producido un *concept drift*, lo que ha provocado un nuevo cálculo del modelo. Este problema hace que el algoritmo sea ineficiente debido a la mayor cantidad de falsos positivos. Por lo tanto, se ha introducido una técnica de estabilización con capacidad de filtrado en el algoritmo GDPC+ para reducir los efectos del *concept drift* en el rendimiento del algoritmo. La característica de estabilización del *concept drift* se presenta en la siguiente sección.

10.3.2. Evaluación de la estabilización transitoria del *concept drift*

Cuando el algoritmo detecta un *concept drift* se estima un nuevo GMM para el *data stream*. Se supone que si hay varios *concept drifts* lo suficientemente cerca, el cambio en el valor de la divergencia de Cauchy–Schwarz $\Delta D_{CS_t}(p, q)$ entre los GMMs (Ecuación (10.6)) es suficientemente estable, por lo que resulta $\gamma \approx 0$ con un nivel de significación del 5 % y H_0 se acepta en el test de ADF. Por lo tanto los datos son estacionarios (es decir, las GMM son muy parecidas) y el GMM no se actualiza, por lo que la detección del *concept drift* puede ser desestimado. Sin embargo, si el valor de $\Delta D_{CS_t}(p, q)$ cambia abruptamente, $\gamma < 0$ tiene un valor de significación y se señala un *concept drift*, por lo que se actualiza el GMM.

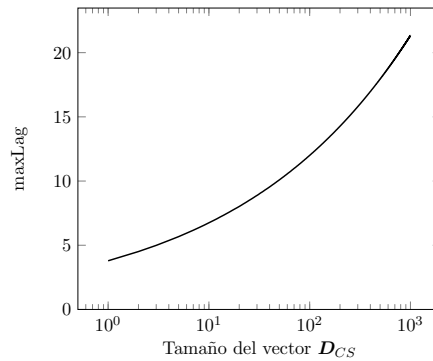


Figura 10.3: rezago máximo vs tamaño del vector D_{CS} .

Este método se usa junto con el proceso de selección dinámica de componentes descrito en la Sección 10.3.1, por lo que $D_{CS}(p, q)$ se debe calcular entre GMMs con diferente número de instancias y componentes durante una ventana temporal de instancias específica. La divergencia de Cauchy–Schwarz (Ecuación (10.5)) puede funcionar bajo esta condición, midiendo las tendencias no estacionarias entre dos distribuciones, $\Delta D_{CS_t}(p, q)$. El comportamiento no estacionario se define como un cambio lo suficientemente elevado en la divergencia entre dos distribuciones como para causar la detección de un nuevo *concept drift*.

En este caso, el test ADF se ejecuta utilizando el módulo de código abierto Statsmodels¹ de Python y su función `statsmodels.tsa.stattools.adfuller()`. De acuerdo con esta función, varios parámetros pueden influir en la detección de *concept drift* de la siguiente manera:

- D_{CS} es el vector de valores de divergencia en \mathfrak{R} entre el GMM actual, $p(\mathbf{x}; \Psi)$, y el actualizado GMM, $q(\mathbf{x}; \Psi)$.
- `regression` es el orden de la Ecuación (10.6) con las siguientes opciones diferentes:
 - `c` constante, es decir, $\beta = 0$ y $\alpha = 0$,
 - `cd` constante y deriva, es decir, $\beta = 0$ y $\alpha \neq 0$, o
 - `cdt` constante, deriva y tendencia temporal, es decir, $\beta \neq 0$ y $\alpha \neq 0$.
- `maxlag` es el rezago máximo en la prueba y se calcula con la siguiente ecuación:

$$\text{maxlag} = 12 \times \left(\frac{\text{longitud del vector } D_{CS}}{100} \right)^{1/4}$$

que tiene la forma mostrada en la Figura 10.3. Uno de los objetivos del algoritmo GDPC+ es reducir significativamente el requisito de almacenamiento de datos, por lo que los valores de `maxlag` deben mantenerse pequeños.

- `autolag` se usa para seleccionar el rezago u de la Ecuación (10.6), de la siguiente manera:

¹www.statsmodels.org

Tabla 10.2: Diseño de experimentos: parámetros y niveles.

Parámetros	Nivel 1	Nivel 2	Nivel 3
regression	c	cd	cdt
maxlag	8	10	12
autolag	None	AIC	BIC

Tabla 10.3: Resultados de las pruebas de GDPC+ con diferentes parámetros y $N = 10$.

Prueba	regression	autolag	maxlag	Inst/s	CD_{true}	n_{CD}	CD_{det}	Prec. (%)	Recall (%)	Espec. (%)	F-score (%)
1	c	AIC	8	779,56 ± 138,17	3,10 ± 0,94	13,00 ± 2,28	{6133, 6393, 9819, 18400}	99,93 ± 0,05	77,50 ± 23,58	99,95 ± 0,05	84,92 ± 18,05
2	cd	BIC	10	677,31 ± 47,33	0,70 ± 0,78	2,90 ± 1,14	{18400}	100,00 ± 0,00	17,50 ± 19,53	100,00 ± 0,00	25,34 ± 26,98
3	cdt	None	12	818,66 ± 255,93	1,00 ± 0,45	1,80 ± 0,40	{9819}	100,00 ± 0,00	25,00 ± 11,18	100,00 ± 0,00	38,67 ± 15,15
4	c	BIC	12	750,97 ± 69,70	0,00 ± 0,00	0,80 ± 0,40	-	100,00 ± 0,00	0,00 ± 0,00	100,00 ± 0,00	0,00 ± 0,00
5	cd	None	8	835,40 ± 280,89	2,60 ± 1,02	14,30 ± 3,55	{6133, 6393, 9819}	99,91 ± 0,03	65,00 ± 25,50	99,91 ± 0,03	74,73 ± 26,59
6	cdt	AIC	10	732,83 ± 146,84	0,60 ± 0,66	4,80 ± 1,25	{9819}	99,97 ± 0,05	15,00 ± 16,58	100,00 ± 0,00	22,67 ± 23,89
7	c	None	10	734,14 ± 43,03	1,30 ± 0,46	5,60 ± 0,49	{9819}	100,00 ± 0,00	32,50 ± 11,46	100,00 ± 0,00	48,01 ± 12,24
8	cd	AIC	12	690,87 ± 37,38	0,50 ± 1,02	1,10 ± 1,04	-	100,00 ± 0,00	12,50 ± 25,62	100,00 ± 0,00	15,24 ± 30,77
9	cdt	BIC	8	782,47 ± 127,27	3,20 ± 0,87	13,50 ± 2,16	{6133, 6393, 9819, 18400}	99,91 ± 0,03	80,00 ± 21,79	99,91 ± 0,03	87,08 ± 14,42

- None (ninguno), cuando se usa el valor de `maxlag`.
- AIC (criterio de información de Akaike) o BIC, cuando se selecciona el rezago para minimizar el criterio de información correspondiente.

Similar al enfoque de GDPC propuesto en el Capítulo 9, se utiliza un diseño de Taguchi para comprender cómo estos parámetros influyen en los resultados. Consideramos tres parámetros de prueba ADF: `regression`, `maxlag`, y `autolag`, donde cada uno tiene tres niveles. Por lo tanto, el diseño más adecuado para ajustarse a este número de parámetros y niveles es el diseño ortogonal Taguchi L9 (Taguchi y Wu, 1979) con tres parámetros en cuatro niveles, es decir, un diseño de 4^3 con nueve observaciones. Los valores para cada parámetro se muestran en la Tabla 10.2.

Los procesos aleatorios dentro del algoritmo GDPC+ podrían influir en el test ADF, por lo que los experimentos se han repetido 10 veces, promediando los indicadores de rendimiento y su desviación estándar. Los resultados se muestran en la Tabla 10.3, donde la columna denotada como CD_{det} indica en dónde el GDPC+ ha detectado un *concept drift*.

De acuerdo con la Tabla 10.3, la prueba 1 ha obtenido la mayor cantidad de resultados de detección verdaderos para el algoritmo GDPC+. Sin embargo, la variabilidad en CD_{true} denota que los cuatro cambios de concepto no se han detectado en algunos casos. Se ha obtenido un rendimiento similar en la prueba 9 pero con una precisión, especificidad y valores de *recall* ligeramente inferiores. Además, las pruebas 1 y 9 han obtenido valores estrechos similares en términos de su varianza, lo que indica que son más robustos que las otras opciones. La prueba 5 ha obtenido un rendimiento similar pero con solo tres detecciones.

Estos resultados experimentales demuestran que se ha obtenido una reducción considerable en el número de falsos positivos, independientemente de los parámetros utilizados en el test ADF. En comparación con los resultados de la Tabla 10.1, la reducción en el número de detecciones de *concept drift* (véanse los valores n_{CD}) ha sido de alrededor de un orden de magnitud, y el algoritmo GDPC+ ha mejorado la especificidad en comparación con GDPC.

Además, la velocidad del algoritmo en términos de la frecuencia de procesamiento se ha

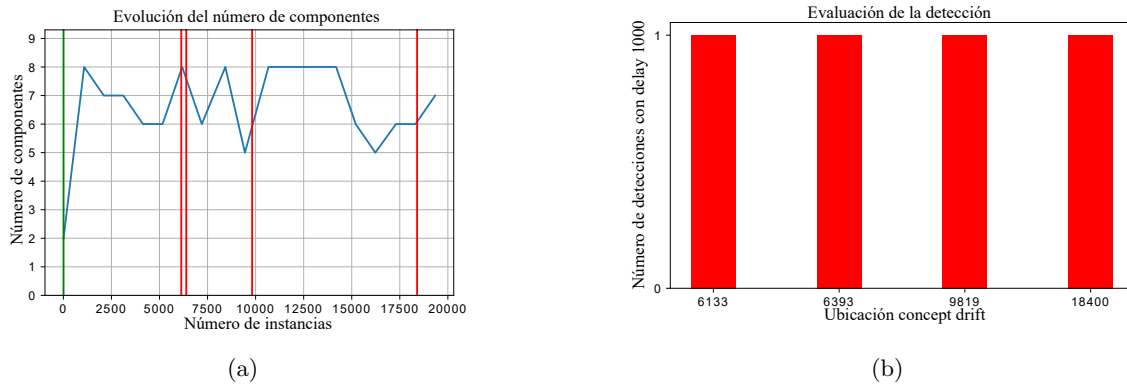


Figura 10.4: (a) La selección dinámica de componentes y evolución de la estabilización del *concept drift* en función del *data stream*. (b) Número de instancias detectadas como *dconcept drift* con un retraso de 1000 instancias.

mantenido entre 629,98 y 1116,29 instancias por segundo, algo aceptable para trabajar en una base de tiempo de muestreo de milisegundos.

Por lo tanto, los parámetros óptimos para el GDPC+ se establecen de la siguiente manera: regresión constante (c), AIC como criterio de información para minimizar el rezago, y 8 como el rezago máximo. Este rezago asegura que el vector D_{CS} requeriría alrededor de 20 mediciones de divergencia (ver Figura 10.3) para identificar el comportamiento no estacionario, algo asequible en términos de almacenamiento de datos. Usando estos parámetros y gracias al test ADF, la actualización del modelo es rechazada cuando los datos no son estacionarios, tal y como se muestra en la Figura 10.4a. Se ha obtenido una mejora significativa en la estabilidad en comparación con los resultados mostrados en la Figura 10.2a, donde solo se ha conseguido una detección de *concept drift* dentro de un rango de 1000 instancias (ver la Figura 10.4b).

10.4. Aplicación del GDPC+ al proceso de mecanizado de cigüeñales

10.4.1. Descripción del conjunto de datos

Para probar la idoneidad de GDPC+ para su uso en escenarios reales, se ha obtenido un *data stream* con 31 ciclos de mecanizado realizados por una máquina-herramienta real (Figura 8.1a). En este caso, la máquina-herramienta se ha usado para fabricar 31 cigüeñales (Figura 10.5), donde las variables del proceso incluían la velocidad angular, la temperatura, la potencia y el par tomados de cada uno de los dos husillos de la máquina (Figura 8.1b). Cada ciclo ha requerido alrededor de 30 s, equivalente a 3000 instancias.

Los ciclos de mecanizado se muestran en la Figura 10.6 en términos de la velocidad angular (Figura 10.6a), potencia (Figura 10.6c), y temperatura (Figura 10.6e). Para obtener los detalles del ciclo, las Figuras 10.6b, 10.6d y 10.6f muestran los datos de estas tres variables durante dos ciclos de mecanizado, es decir, 6000 instancias.



Figura 10.5: Cigüeñal producido por la máquina mostrada en la Figura 8.1a

Tabla 10.4: Resultados para `regression = c`, `autolag = AIC`, `maxlag = 8`.

Prueba	N	$Inst/s$	n_{CD}
1	1000	688,59	65,0
2	3000	698,64	64,0
3	5000	668,64	62,0
4	10000	1737,84	0,0

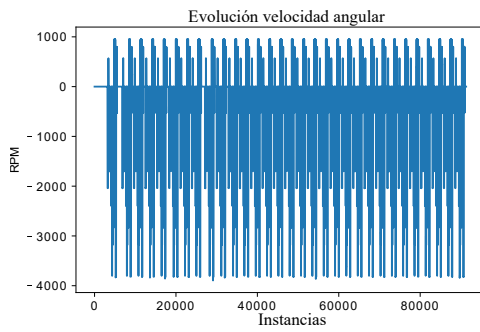
10.4.2. Detección del *concept drift* por el GDPC+

La secuencia de datos se ha probado utilizando los parámetros que obtuvieron el mejor rendimiento en la prueba ADF (Tabla 10.3, prueba 1). Para estudiar diferentes perspectivas del procesamiento del *data stream*, se utiliza el parámetro GDPC+ denominado longitud de ventana temporal N (véase el recuadro 1 en la Figura 10.1b). Este parámetro considera la cantidad de datos para generar el primer modelo, que luego se utiliza como referencia para el posterior análisis *online*, es decir, los datos utilizados para el entrenamiento. Por lo tanto, se han seleccionado varios valores de N dependiendo del comportamiento de la máquina. Es decir, antes de comenzar el mecanizado: $N = 1000$ y 3000 ; después de un ciclo de mecanizado: $N = 5000$; y después de tres ciclos de mecanizado: $N = 10000$.

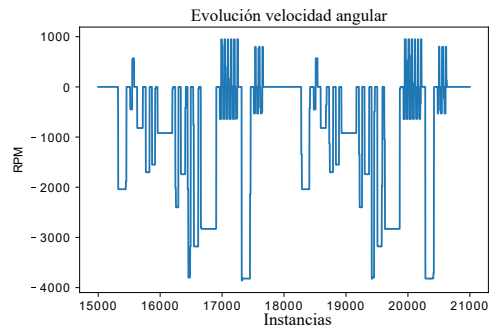
De la Tabla 10.4 se extrae el siguiente análisis:

1. Antes del inicio del proceso de mecanizado (pruebas 1 y 2): en este caso, el algoritmo se ha entrenado utilizando los datos de la máquina en *standby*. Con $N = 1000$ y 3000 , las cantidades de *concept drift* (68) han sido las mismas, con frecuencias de procesamiento ligeramente diferentes (688,59 y 698,64) debido al aumento del tamaño de la ventana temporal inicial, es decir, se ha requerido más tiempo para el entrenamiento.

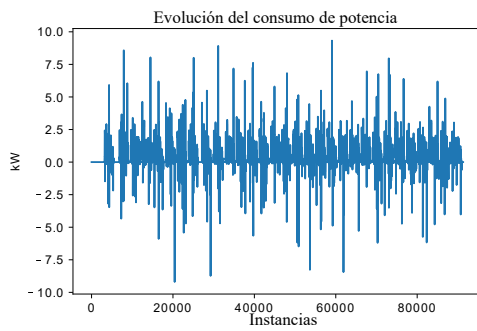
Los cambios en el número de componentes se muestran en las Figuras 10.7a y 10.7c para $N = 1000$ y $N = 3000$, respectivamente. El número de componentes ha tenido una variación cercana a seis, con un mínimo de cinco para $N = 1000$ y cuatro para $N = 3000$, y el máximo ha sido nueve en ambos casos. Los *concept drifts* detectados con ambos valores para N se muestran en la Figura 10.7b ($N = 1000$) y la Figura 10.7d ($N = 3000$) usando la señal de potencia, que es la variable más crítica para examinar los ciclos de mecanizado. Cabe señalar que los *concept drifts* se han ubicado a diferentes valores de potencia con separaciones similares entre ellos. Por lo tanto, durante el ciclo



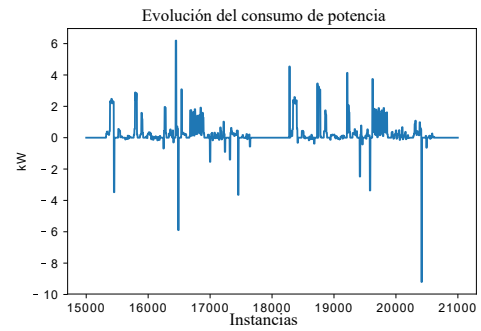
(a) Ciclos de velocidad angular



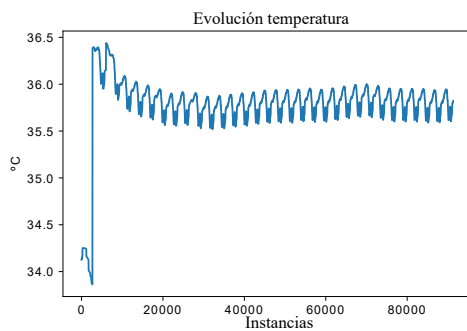
(b) Velocidad angular (dos ciclos)



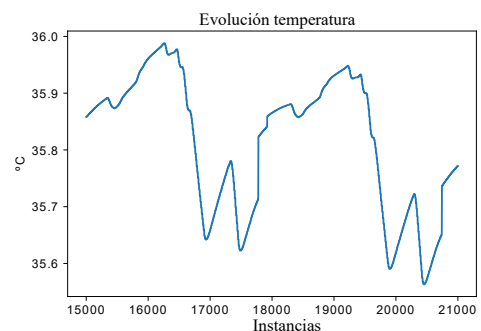
(c) Ciclos de potencia



(d) Potencia (dos ciclos)



(e) Ciclos de temperatura



(f) Temperatura (dos ciclos)

Figura 10.6: *Data stream* de una aplicación de ingeniería real.

(la máquina realizando el mismo proceso), las fluctuaciones de potencia de la máquina se han controlado para mantener las mismas condiciones de corte, lo que explica los *concept drifts*. En este nivel de análisis los valores de N para las pruebas 1 y 2 han configurado el algoritmo para detectar cambios de concepto en el sistema de control en términos de suministro de energía, es decir, anomalías del sistema de control. Además, se podrían monitorizar otros tipos de degradación, como los debidos a la herramienta, los rodamientos de bolas y los devanados eléctricos.

2. Después de un ciclo de mecanizado (prueba 3): con este tamaño de ventana el algoritmo comienza a detectar *concept drifts* en lugares similares a aquellos con $N = 3000$. Sin embargo, la cantidad es ligeramente menor. La Figura 10.7f muestra que las regiones son similares pero con cambios en el tiempo. El número de componentes varía alrededor de seis (Figura 10.7e), pero el patrón de comportamiento ha sido similar al de $N = 1000$. Por lo tanto, se podrían utilizar pequeñas cantidades de datos de entrenamiento y la selección podría realizarse en términos de la frecuencia de procesamiento, con $N = 1000$ como el mejor valor.
3. Después de tres ciclos de mecanizado (prueba 4): no se ha detectado ningún *concept drift* con este parámetro. Las piezas de trabajo se han producido sin problemas de calidad, tal como se esperaba, sin ninguna degradación de los ciclos de mecanizado. Es importante tener en cuenta que en las pruebas adicionales con datos de más ciclos de mecanizado realizados exactamente de la misma manera no se ha detectado ningún *concept drift*.

10.4.3. GDPC+: resultados de *clustering* sobre el proceso de mecanizado

Después de analizar el rendimiento de GDPC+ en la detección del *concept drift* se deben interpretar los clústers para obtener información sobre el proceso de mecanizado y su evolución, especialmente después de la aparición del *concept drift*. Como ejemplo, se consideran los resultados para $N = 1000$ y los *concept drift* que se muestran en la Figura 10.7b ubicados en las instancias 3898, 5215, 6235, 7469 y 8738.

Para interpretar los resultados se selecciona la etiqueta del clúster como la variable clase supervisada para inducir un conjunto de reglas utilizando el algoritmo *repeated incremental pruning to produce error reduction* (RIPPER) implementado en WEKA como JRip (Hall et al., 2009). RIPPER construye las reglas de clasificación basado en la ganancia de información y luego simplifica las reglas utilizando una estrategia de poda. Las reglas que se muestran en la Tabla 10.5.

Tabla 10.5: Reglas extraídas de los resultados de GDPC+.

<i>Concept drift</i>	Componente	Regla	número de instancias
3898	1	Potencia $\neq 0$ W	359
		Temperatura $\in [34, 12, 34, 16]^{\circ}C$	
		Par ≈ 0 $N \cdot m$	

		Velocidad angular ≈ 0 RPM	
		Temperatura $\approx 34, 2^{\circ}C$	
	2	Par > 0 N · m	339
		Velocidad angular $\geq 85, 6$ RPM	
		Potencia $\neq 0$ W	
	3	Temperatura $\in [34, 12, 34, 16]^{\circ}C$	374
		Par $\in [0, 23, 0, 31]$ N · m y $[-0, 12, -1, 03]$ N · m	
	4	Otros	1717
		Potencia ≥ 0 W	
	5	Temperatura $\in [34, 14 - 34, 25]^{\circ}C$	42
		Par $\geq 0, 26$ N · m	
		Temperatura $\approx 34, 2^{\circ}C$	
	6	Par ≈ 0 N · m	305
		Temperatura $\approx 34, 2^{\circ}C$	
	7	Par $\geq -0, 20$ N · m	745
		Velocidad angular $\leq -57, 12$ RPM	
		Temperatura $\geq 36, 3^{\circ}C$	
	1	Par $\approx -4, 42$ N · m	9
		Temperatura $\leq 36, 0^{\circ}C$	
5215	2	Par $\in [4, 6]$ N · m y $[21, 3, 36, 0]$ N · m	61
	3	Otro	634
	4	Velocidad angular $\leq -1580, 6$ RPM	521
		Temperatura $\approx 36, 4^{\circ}C$	
	5	Par $\leq 0, 77$ N · m	60
		Temperatura $\geq 36, 4^{\circ}C$	
	6	Par $\leq -6, 7$ N · m	10
		Temperatura $\approx 36, 4^{\circ}C$	
	7		17
		Velocidad angular $\neq 3820$ RPM	
	1		10
		Potencia $\leq 0, 16$ W	
6235	2	Velocidad angular ≤ -3820 RPM	20
	3	Otro	936
		Temperatura $\leq 36, 0^{\circ}C$	
	4	Velocidad angular ≤ -3819 RPM	32
		Velocidad angular ≈ -3819 RPM	
	5		18
		Otro	
	1		788
		Potencia $\neq 0$ W	
7469	2	Temperatura $\neq 36, 3^{\circ}C$	166
		Par ≈ 0 N · m	
		Par $\neq 0$ N · m	
	3	Velocidad angular $\geq 316, 6$ RPM	92
		Temperatura $\geq 36, 4^{\circ}C$	
	4	Par ≥ 0 N · m	175
		Velocidad angular ≥ 0 RPM	
		Temperatura $\approx 36, 4^{\circ}C$	
	5	Velocidad angular ≥ 0 RPM	21
		Temperatura $\approx 36, 3^{\circ}C$	
	1	Par $\in [-2, 3, 4, 2]$ N · m	877
		Velocidad angular $\leq -920, 2$ RPM	
8738		Potencia $\neq 0$ W	
		Temperatura $\in [35, 36]^{\circ}C$	
	2	Par $0, 1, 0, 2, 0, 3$ y $\geq 4, 6$ N · m	585
		Velocidad angular $\in [0 - 445, 53, 5]$ RPM	
		Potencia ≤ 0 W	
		Temperatura $\approx 36, 0^{\circ}C$	
	3	Par $\leq -22, 8$ N · m	672
		Velocidad angular $\in [-743, 16, 8]$ RPM	
	4	Otro	1253
		Potencia $\geq 0, 8$ W	
	5		53

	Temperatura $\leq 36, 3^{\circ}C$	
	Velocidad angular $\approx -3821 \text{ RPM}$	
6	Potencia $\in [-3, 0, 2] \text{ W}$	521
	Temperatura $\leq 36, 3^{\circ}C$	
	Par $\in [-3, 6, -0, 4] \text{ N} \cdot \text{m}$	
	Velocidad angular $\in [-3818, -919, 8] \text{ RPM}$	

De acuerdo con las reglas mostradas en la Tabla 10.5, para el primer *concept drift* en la instancia 3898, el Clúster 1 representa el estado de la máquina detenida, que no existe en los siguientes *concept drifts*. Por lo tanto, una de las causas del *concept drift* es debida a que la máquina comienza por debajo de la temperatura de operación. Los clústers 2, 3, 5, 6 y 7 en 3898 eran similares al clúster 2 en 5215 y representan a la máquina trabajando a bajas temperaturas, es decir, durante su calentamiento. Después de alcanzar una cierta temperatura, el GMM actual ya no es válido, por lo que se lanza una actualización del modelo en la instancia 6235. Por lo tanto, los clústers 1, 4, 5, 6 y 7 en 5215 se pueden interpretar como la operación de máquina durante el calentamiento tardando alrededor de 1 s en el *concept drift* 6235. Este *concept drift* fue similar al ubicado en la instancia 8738.

En la instancia 6235 la máquina está funcionando a altas velocidades angulares con una finalización del ciclo de mecanizado. Por lo tanto, se ha detectado un *concept drift* causado por la parada de máquina. El clúster 4 con baja potencia se relaciona con una alta velocidad angular donde el husillo gira sin mecanizar. En este clúster se encuentran solo 20 casos, lo que indica la alta eficiencia de la máquina evitando una cantidad excesiva de trabajo en vacío.

La máquina ya está caliente en 7469, pero la alta variabilidad en los valores del clúster en términos de par, potencia y velocidad angular indican que el algoritmo GDPC+ detecta la finalización del ciclo de mecanizado, comenzando otro nuevo. Por lo tanto, se ha detectado un *concept drift* durante la transición de un ciclo de mecanizado al siguiente. Durante este período, la máquina ha realizado varias maniobras de posicionamiento mientras se introducía una nueva pieza en la misma. El clúster 3 proporciona información sobre la aceleración requerida para lograr las condiciones de corte. Por lo tanto, los clústers describen diferentes etapas de aceleración, donde la temperatura aumenta para el clúster 2 y el clúster 4 cuando el husillo se detiene y comienza a moverse para alcanzar la velocidad de operación descrita en el clúster 4.

Los clúster obtenidos del *concept drift* ubicado en 8738 han sido similares a los de 5215, donde se ha realizado la operación de mecanizado. Los clúster 5 y 6 han sido similares a los clúster 1 y 4 donde se ha realizado el proceso de mecanizado.

El diferente número de clústers entre los *concept drifts* proporciona información sobre los diferentes estados de la máquina que no estaban disponibles hasta ese momento o nuevos estados que nunca habían ocurrido. Por ejemplo, entre 5215 y 6235, se ha generado un nuevo clúster relacionado con un husillo detenido, es decir, clúster 3. Esta es una característica importante obtenida por GDPC+ porque este nuevo comportamiento del proceso no se podría etiquetar utilizando un algoritmo como GDPC con un número fijo de componentes.

Una vista esquemática de los resultados de la Tabla 10.5 se presenta en la Figura 10.8, donde se usa la velocidad angular del proceso como una referencia para etiquetar cada clúster

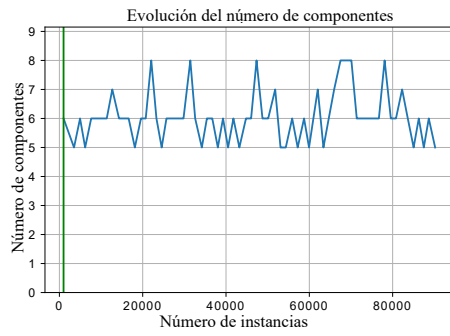
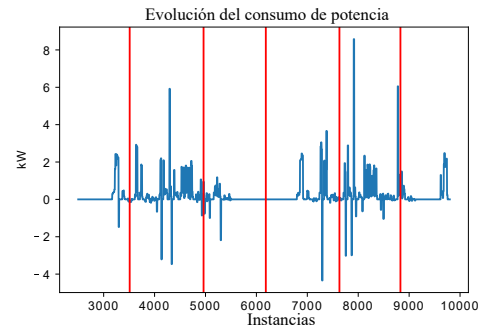
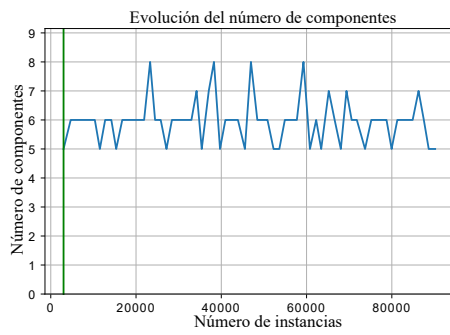
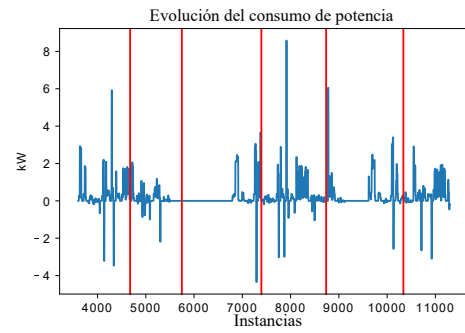
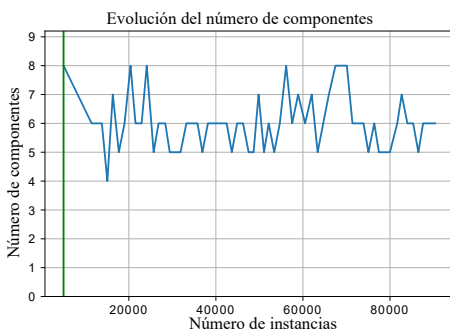
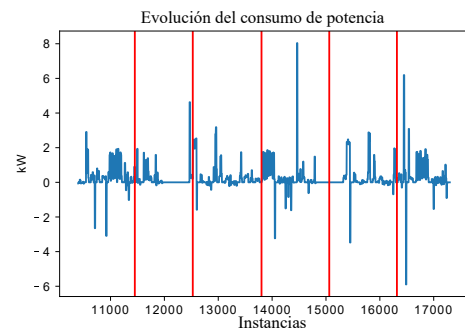
(a) $N = 1000$ (b) *Concept drift* en $N = 1000$ (c) $N = 3000$ (d) *Concept drift* en $N = 3000$ (e) $N = 5000$ (f) *Concept drift* en $N = 5000$

Figura 10.7: Cambios en el número de componentes.

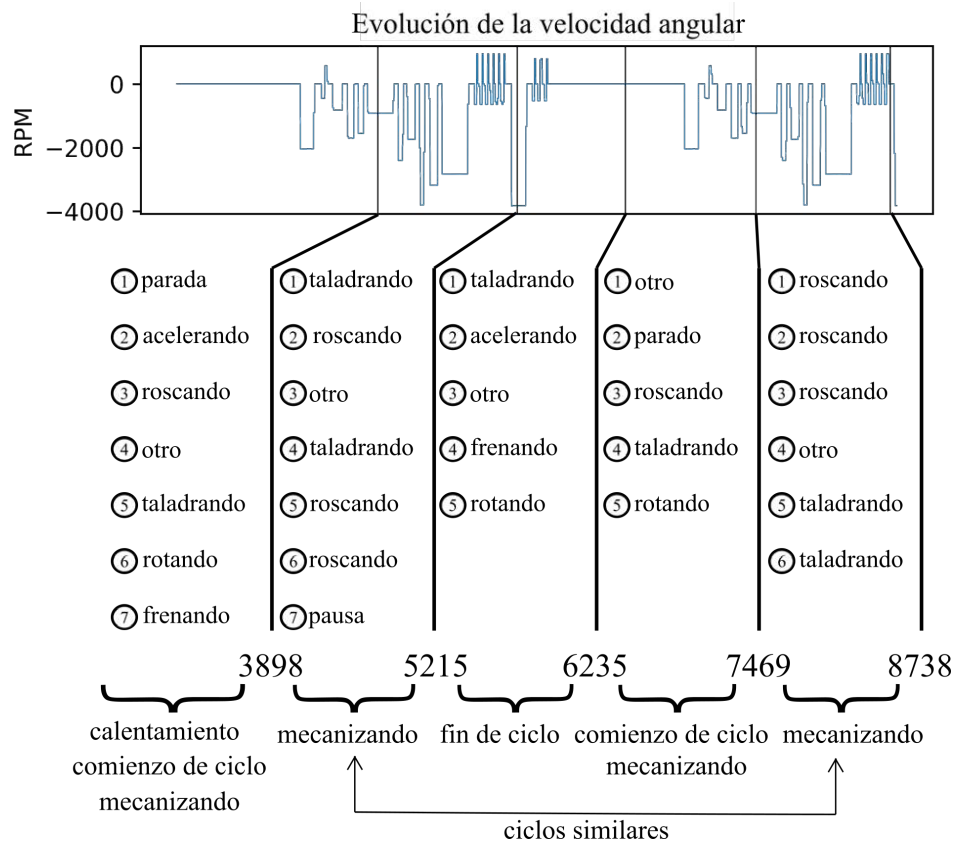


Figura 10.8: Interpretación esquemática de los resultados de la Tabla 10.5.

y cada *concept drift*.

De acuerdo con estos resultados, seleccionar $N = 1000$ en el algoritmo GDPC+ ha sido suficiente para diferenciar pequeños detalles, como la parada de la máquina, el mecanizado, el cambio de la pieza de trabajo (inicio y finalización del ciclo) y el giro del husillo sin carga. Por lo tanto, esta configuración es útil para explorar patrones que pueden ocurrir dentro de un ciclo de mecanizado. Sin embargo, una configuración con una pequeña ventana temporal, como $N = 1000$, podría dar lugar a muchos falsos positivos debido a la detección de pequeñas variaciones relacionadas con las diferentes piezas de trabajo, la degradación de la herramienta, la temperatura de la máquina, etc. La Tabla 10.4, para $N = 10000$, no muestra *concept drifts* al tener menos sensibilidad a los pequeños cambios que podrían pasarse por alto dependiendo del análisis exploratorio requerido.

10.4.4. Comparación entre GDPC y GDPC+

Para analizar cualitativamente las diferencias en el rendimiento de los algoritmos GDPC y GDPC+ cuando operan en escenarios desconocidos, se utiliza el *data stream* presentado en la Sección 10.4.2 con el GDPC y diferentes valores para el número de componentes $K = 3, 5$. La ventana de entrenamiento se establece en $N = 10000$ para obtener una vista del proceso de mecanizado completo con tres ciclos de mecanizado (prueba 4, Tabla 10.4), y los resultados

Tabla 10.6: Resultados de la comparativa entre GDPC y GDPC+.

Número of componentes	N	$Inst/s$	n_{CD}
GDPC; $K = 3$	10000	826,32	1306
GDPC; $K = 5$	10000	605,45	1487
GDPC+	10000	2068,66	0,0

se muestran en la Tabla 10.6.

El GDPC ha tenido un desempeño pobre en términos de la cantidad de falsos positivos porque no se esperaba que se detectaran *concept drifts*, pero sí ha generado una gran cantidad de detecciones de *concept drift*, imposibles de manejar desde el punto de vista del proceso de mecanizado. Por el contrario, el GDPC+ ha podido obtener un control efectivo debido a la estabilidad del algoritmo y ha sido dos veces más rápido que GDPC.

10.5. Conclusiones

En este capítulo se presentan mejoras importantes en el algoritmo de GDPC en dos aspectos: selección de componentes y estabilización durante situaciones transitorias, no estacionarias. De esta manera, la función de selección dinámica de componentes puede detectar el mejor número de componentes para el GMM sin requerir una estimación inicial por parte de un experto. Además, la divergencia de Cauchy–Schwarz combinada con el test ADF permite controlar la cantidad de falsos positivos desencadenados por datos no estacionarios durante el *concept drift*. El algoritmo GDPC+ recoge ambos aspectos de mejora.

El número de parámetros y su selección se reducen a uno solo debido a estas mejoras. Para utilizar GDPC+ como una herramienta de descubrimiento de conocimiento el único parámetro que debe definirse previamente por el usuario es el tamaño de la ventana de entrenamiento inicial, N . Esta posibilita además seleccionar la granularidad adecuada según los requisitos de monitorización. Esta granularidad del análisis predefinida permite al GDPC+ ser altamente flexible.

Sobre la base de las pruebas realizadas con el algoritmo GDPC+ se puede concluir lo siguiente:

- El *concept drift* detectado por el algoritmo GDPC+ podría compararse efectivamente con los fenómenos físicos reales detectados.
- El algoritmo GDPC+ puede operar bajo condiciones de *data stream* con frecuencias de análisis tan altas como 0,3 a 1,5 *kHz*.
- El GDPC+ supera al GDPC en condiciones completamente desconocidas, donde puede actualizar el modelo GMM automáticamente con una reducción significativa en el número de falsos positivos.

- La actualización del modelo produce una gran variedad de reglas de comportamiento, que pueden contener información para cada estado estacionario, así como información sobre *concept drifts*.

Parte IV

Conclusiones

Capítulo 11

Conclusiones y trabajo futuro

De acuerdo a la hipótesis planteada en la Sección 1.5:

la degradación de los elementos de máquina, sujetos a comportamientos dinámicos, es detectable de una manera más rápida mediante el estudio de los patrones que forman sus principales variables de funcionamiento a lo largo del tiempo

se han desarrollado actividades de investigación descritas en los apartados capítulos que han permitido generar las herramientas apropiadas para el planteamiento de la tesis. Las actividades se resumen en:

1. el estudio del comportamiento de algoritmos de *clustering* para aplicaciones industriales,
2. la selección del algoritmo de *clustering* apropiado para estudiar la degradación de elementos, y
3. la propuesta de un nuevo algoritmo de *clustering* probabilístico que monitoriza la degradación dinámica de los elementos (*concept drift*) en condiciones dinámicas (*data stream*),

permitiendo cumplir con los objetivos planteados:

- Investigar y desarrollar de algoritmos de aprendizaje automático en el campo industrial, específicamente en el sector de la máquina-herramienta y los procesos de tratamiento térmico láser.
- Investigar patrones de funcionamiento mediante la aplicación de algoritmos de clustering en datos estáticos de elementos de maquinaria.
- Desarrollar sistemas de clustering probabilístico dinámico para monitorizar *online* la degradación de elementos de máquina.

- Desarrollar e implementar algoritmos de selección de variables espacio-temporales para incrementar el rendimiento de algoritmos de aprendizaje automático y reducir las necesidades de almacenamiento de datos históricos.

Todo esto ha permitido extraer las conclusiones, resaltar los logros y exponer el trabajo a futuro que se describe a continuación.

11.1. *Clustering*

Los aportes al *clustering* se hallan, tanto a nivel de algoritmos, como de adquisición de datos. Ambas son piezas fundamentales para poder entender el valor añadido que puede aportar este tipo de técnicas en la detección de patrones de comportamiento de los elementos.

Por esta razón, para poder estudiar el comportamiento de los elementos ha sido fundamental asegurar la calidad de los datos. Esto implica hacer un pre-procesamiento que permita obtener las variables en los tiempos de muestreo diseñados para lograr sacar el máximo provecho en términos de nuevo conocimiento. Es por ello que se ha construido un banco de pruebas con elementos singulares aplicados a la máquina-herramienta que se encuentra certificado por el IIC . Este banco ha permitido realizar las actividades descritas en los Capítulos 7 y 9.

En el Capítulo 6 se han aplicado técnicas de selección de variables. Adicionalmente, se ha estudiado la posibilidad de aplicar diferentes algoritmos de *clustering*. Especialmente, el algoritmo jerárquico aglomerativo orientado a la reducción dimensional en aplicaciones de visión de alta velocidad para procesos térmicos.

En el Capítulo 7 las técnicas de imputación de variables han sido imprescindibles debido a la existencia de variables provenientes de diferentes dominios (temperatura, par, potencia, etc.). La reducción dimensional ha sido útil para permitir entregar de manera adecuada los resultados al usuario final para su posterior análisis.

En el campo del procesamiento con técnicas de *clustering*, se han podido analizar diferentes tipos de algoritmos de acuerdo a resultados encontrados en términos del aporte de nuevo conocimiento. En el Capítulo 7, los algoritmos K -means, jerárquicos, espectral, propagación de afinidad y GMMs se han puesto a prueba con datos de elementos de máquina. El trabajo descrito en este capítulo ha permitido seleccionar aquellos algoritmos que pueden funcionar correctamente en condiciones de operación. Como resultado, se han descartado los algoritmos espectral y de propagación de afinidad por la complejidad en sus parámetros, en los mismos es necesario una gran cantidad de conocimiento del dominio y del algoritmo para poder configurarlos de manera adecuada. Por otro parte, esta condición hace que estos algoritmos sean especialmente complejos para extrapolar a otros elementos de máquina.

En el Capítulo 8 se han aplicado los tres algoritmos restantes, K -means, jerárquico y GMM, al elemento más crítico de una máquina-herramienta, en este caso, el electromandrino o husillo de mecanizado, durante un proceso real de mecanizado. En los análisis se ha podido comprobar que el GMM ha sido el único algoritmo con la capacidad de aportar nueva información al usuario, e incluso, discriminar los diferentes estados del elemento para poder analizar sus patrones de funcionamiento a lo largo del proceso. Esta es una conclusión

a destacar, ya que el estudio de la degradación es diferente dependiendo de los estados del elemento. Adicionalmente, el enfoque *soft clustering* del GMM permite detectar y analizar aquellas instancias que pueden pertenecer, con diferentes probabilidades, a varios clústers a la vez.

Como resultado de estos análisis exploratorios, se selecciona el algoritmo GMM como algoritmo subyacente para su aplicación a entornos industriales, altamente dinámicos, que se describen a continuación.

11.1.1. Aportaciones al *clustering*: GDPC+

Se ha propuesto un nuevo algoritmo de *clustering* probabilístico dinámico basado en GMMs que permite estudiar los patrones de comportamiento de los elementos y su degradación. Este algoritmo supone una novedad en el campo del *clustering* de *data streams*, ya que en la actualidad, dentro de las técnicas existentes no se encuentran soluciones que contemplen el uso del GMM como el algoritmo subyacente.

En el Capítulo 9, se proponen las líneas generales del algoritmo probabilístico, denominado GDPC. En esta fase se trabaja especialmente en hacer más eficiente el uso del algoritmo EM, al ser el paso que más recursos computacionales requiere. Adicionalmente, se busca la eficiencia en el uso y el almacenamiento de los datos, ya que teniendo en cuenta las aplicaciones industriales, estos recursos son limitados.

Por tanto, el algoritmo GDPC propuesto ha logrado detectar con éxito diferentes *concept drifts* encontrados en *data streams* sintéticos y producidos de manera controlada con un banco de pruebas. Se ha desarrollado una estrategia para que una vez detectada la incapacidad del modelo para representar a los datos que se van adquiriendo del *data stream*, se pueda actualizar un nuevo GMM de manera rápida y eficiente. Se ha estudiado el algoritmo a la hora de asignar instancias a diferentes componentes, en donde ha sido posible analizar su comportamiento adaptativo a medida que el *data stream* va evolucionando en el tiempo.

En el Capítulo 10, se trabaja en la segunda fase de desarrollo del algoritmo probabilístico, denominado GDPC+. El primer objetivo es permitirle a esta nueva herramienta trabajar en entornos parcial o completamente desconocidos, reduciendo así las necesidades de configuración de parámetros que requieren un conocimiento profundo del dominio y del algoritmo. Especialmente, se ha implementado un paso para seleccionar el número de componentes de manera automática, basado en la minimización del valor BIC. Al eliminar la necesidad de introducir el número de componentes como parámetro, el algoritmo puede ir variando su número a lo largo del tiempo permitiendo monitorizar la generación de nuevos estados a partir de la generación, unión y división de los componentes.

Por otro lado, se ha trabajado en reducir los falsos positivos producidos por comportamientos no estacionarios del *data stream*. Para ello se ha trabajado en una solución basada en la divergencia de Cauchy–Schwarz y el test ADF aumentado para disminuir la tendencia de inestabilidad producida por ventanas temporales que contienen datos provenientes de fenómenos no estacionarios.

Con todos estos desarrollos ha sido posible proponer, desarrollar y validar un nuevo algo-

ritmo probabilístico basado en GMMs que permite generar patrones de comportamiento de los elementos y a su vez, monitorizar la degradación de los mismos.

11.2. Publicaciones y otros resultados

A continuación, se detallan las actividades de difusión más importantes asociadas con los capítulos presentados anteriormente:

1. Revistas:

- J. Diaz-Rozo, C. Bielza y P. Larrañaga. Machine-tool condition monitoring with Gaussian mixture models-based dynamic probabilistic clustering. *Journal of Engineering Applications of Artificial Intelligence*. Enviado en marzo 2019.
- J. Diaz-Rozo, C. Bielza y P. Larrañaga. Clustering of data streams with dynamic Gaussian mixture models: An IoT application in industrial processes. *IEEE Internet of Things Journal*, vol. 5, n.º 5, pp. 3533-3547, 2018.

2. Libro:

- P. Larrañaga, D. Atienza, J. Diaz-Rozo, A. Ogbechie, C. E. Puerto-Santana y C. Bielza, *Industrial Applications of Machine Learning*. CRC Press, 2019.

3. Congresos:

- J. Diaz-Rozo, C. Bielza, y P. Larrañaga. Machine learning-based CPS for clustering high throughput machining cycle conditions. En *Procedia Manufacturing*, vol. 10, pp. 997-1008, 2017.
- J. Diaz, C. Bielza, J. L. Ocaña, y P. Larrañaga. Development of a Cyber-Physical System based on selective Gaussian naïve Bayes model for a self-predict laser surface heat treatment process control. En *Machine Learning for Cyber Physical Systems*, Springer. pp. 1-8. 2016.
Premiado como mejor artículo del primer congreso *Machine Learning for Cyber-Physical Systems* en Lemgo, Alemania.

Adicionalmente, como resultado de las actividades de investigación se ha generado la patente de ámbito mundial:

- J. Diaz y J. J. Gabilondo, Device and system including multiple devices for supervision and control of machines in industrial installation, US10031500B1, 2018.

Esta patente está relacionada con el diseño de un dispositivo y/o conjunto de dispositivos que permiten el despliegue de algoritmos de aprendizaje automático de manera distribuida, aprovechando los recursos disponibles en cada dispositivo, permitiendo así aumentar la capacidad computacional a pie de máquina.

11.3. Trabajo futuro

Tal y como se ha comentado en la Parte III, el trabajo relacionado con el *clustering* de *data streams* aún no está muy desarrollado, con lo que existen líneas interesantes de trabajo futuro. Especialmente, el campo del *clustering* dinámico probabilístico está prácticamente inexplorado, con lo que el desarrollo del GDPC+ es una línea que tiene diferentes retos y oportunidades.

De esta manera, para el algoritmo GDPC+ es necesario seguir trabajando en la reducción de los falsos positivos, para lo cual, será necesario investigar en nuevos métodos de estimación de parámetros que permitan asegurar una correcta identificación. De acuerdo a esto, será necesario investigar en nuevas técnicas de ventanas temporales adaptativas, detección de valores atípicos, medición de divergencias entre distribuciones, para luego implementar y validar de acuerdo a las necesidades.

Por otro lado, hace falta desplegar de manera desatendida el algoritmo dentro de un dispositivo a pie de máquina, en donde será posible validar su rendimiento a lo largo del tiempo ante situaciones de *data stream*. Será muy interesante comprobar también las posibilidades de aceleración que permiten los diferentes tipos de hardware, como son la lógica programable y los procesadores, muy presentes en otras líneas de desarrollo a lo largo de estos años de investigación. Por lo tanto, se deberá trabajar para implementar el algoritmo en un sistema con un Zynq® Ultrascale+™ MPSoC integrado como unidad de procesamiento dentro de dispositivo comercial, que está siendo desarrollado por Aingura IIoT. Una vez que el algoritmo funcione correctamente en este dispositivo, se podrá realizar un estudio de validación a largo plazo durante tiempos de producción prolongados.

Bibliografía

- M. R. Ackermann, M. Märtens, C. Raupach, K. Swierkot, C. Lammersen, y C. Sohler. StreamKM++: A clustering algorithm for data streams. *Journal of Experimental Algorithms*, 17:2–4, 2012.
- C. C. Aggarwal, P. S. Yu, J. Han, y J. Wang. A framework for clustering evolving data streams. In *Proceedings of the 2003 Very Large Data Bases Conference*, pages 81–92. Elsevier, 2003.
- H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.
- I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, y E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422, 2002.
- A. Ali, G. A. Shah, M. O. Farooq, y U. Ghani. Technologies and challenges in developing machine-to-machine applications: A survey. *Journal of Network and Computer Applications*, 83:124–139, 2017.
- D. Arthur y S. Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of 18th Symposium on Discrete Algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.
- R. F. Babiceanu y R. Seker. Manufacturing cyber-physical systems enabled by complex event processing and big data environments: A framework for development. In *Service Orientation in Holonic and Multi-Agent Manufacturing*, Studies in Computational Intelligence, pages 165–173. Springer, Cham, 2015.
- D. S. Badkar, K. S. Pandey, y G. Buvanashakaran. Parameter optimization of laser transformation hardening by using Taguchi method and utility concept. *International Journal of Advanced Manufacturing Technology*, 52(9):1067–1077, 2011.
- R. Baheti y H. Gill. Cyber-physical systems. *The Impact of Control Technology*, 12(1): 161–166, 2011.
- D. Barbará y P. Chen. Tracking clusters in evolving data sets. In *Proceedings of the 14th Florida Artificial Intelligence Research Society Conference*, pages 239–243, 2001.

- S. T. Barnard, A. Pothen, y H. Simon. A spectral algorithm for envelope reduction of sparse matrices. *Numerical Linear Algebra with Applications*, 2(4):317–334, 1995.
- R. G. Bennett. Computer integrated manufacturing. *Plastic World*, 43(6):65–68, 1985.
- C. Biernacki, G. Celeux, y G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719–725, 2000.
- A. Bifet, G. Holmes, R. Kirkby, y B. Pfahringer. MOA: Massive online analysis. *Journal of Machine Learning Research*, 11:1601–1604, 2010.
- H. Borchani, P. Larrañaga, J. Gama, y C. Bielza. Mining multi-dimensional concept-drifting data streams using Bayesian network classifiers. *Intelligent Data Analysis*, 20(2):257–280, 2016.
- M.-R. Bouguelia, A. Karlsson, S. Pashami, S. Nowaczyk, y A. Holst. Mode tracking using multiple data streams. *Information Fusion*, 43:33–46, 2018.
- G. W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.
- F. Cao, M. Ester, W. Qian, y A. Zhou. Density-based clustering over an evolving data stream with noise. In *Proceedings of the 2006 International Conference on Data Mining*, volume 6, pages 328–339. Society for Industrial and Applied Mathematics, 2006.
- Y. Chen y L. Tu. Density-based clustering for real-time stream data. In *Proceedings of the 13th International Conference on Knowledge Discovery and Data Mining*, pages 133–142. ACM, 2007.
- H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, pages 493–507, 1952.
- Y.-W. Cheung y K. S. Lai. Lag order and critical values of the augmented Dickey–Fuller test. *Journal of Business & Economic Statistics*, 13(3):277–280, 1995.
- C.-F. Chien, W.-C. Wang, y J.-C. Cheng. Data mining for yield enhancement in semiconductor manufacturing and an empirical study. *Expert Systems with Applications*, 33(1):192–198, 2007.
- A. K. Choudhary, J. A. Harding, y M. K. Tiwari. Data mining in manufacturing: A review based on the kind of knowledge. *Journal of Intelligent Manufacturing*, 20(5):501, 2008.
- A. P. Dempster, N. M. Laird, y D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

- J. Diaz y J. J. Gabilondo. Device and system including multiple devices for supervision and control of machines in industrial installation, 2018. US10031500B1.
- J. Diaz, J. Posada, I. Barandiaran, y C. Toro. Recommendations for sustainability in production from a machine-tool manufacturer. In *Sustainable Design and Manufacturing Conference*. KES International, 2015.
- J. Diaz, C. Bielza, J. L. Ocaña, y P. Larrañaga. Development of a Cyber-Physical System based on selective Gaussian naïve Bayes model for a self-predict laser surface heat treatment process control. In *Machine Learning for Cyber Physical Systems, Technologien Für Die Intelligente Automation*, pages 1–8. Springer Vieweg, Berlin, Heidelberg, 2016.
- J. Diaz-Rozo y D. Isaacs. Smart factory machine learning for predictive maintenance testbed. <http://www.iiconsortium.org/smart-factory-machine-learning.htm>, 2017.
- J. Diaz-Rozo, C. Bielza, y P. Larrañaga. Machine learning-based CPS for clustering high throughput machining cycle conditions. *Procedia Manufacturing*, 10:997–1008, 2017.
- J. Diaz-Rozo, C. Bielza, y P. Larrañaga. Clustering of data streams with dynamic Gaussian mixture models: An IoT application in industrial processes. *IEEE Internet of Things Journal*, 5(5):3533–3547, 2018.
- J. Diaz-Rozo, C. Bielza, y P. Larrañaga. Machine-tool condition monitoring with Gaussian mixture models-based dynamic probabilistic clustering. *Journal of Engineering Applications of Artificial Intelligence*, 2019. Enviado.
- D. A. Dickey y W. A. Fuller. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366):427–431, 1979.
- E. R. Faria, I. J. Gonçalves, A. C. de Carvalho, y J. Gama. Novelty detection in data streams. *Artificial Intelligence Review*, 45(2):235–269, 2016.
- L. Fei-Fei, R. Fergus, y P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.
- I. K. Fodor. A survey of dimension reduction techniques. Technical Report UCRL-ID-148494, Lawrence Livermore National Lab., CA (US), 2002.
- C. Frederickson, T. Gracie, S. Portley, M. Moore, D. Cahall, y R. Polikar. Adding adaptive intelligence to sensor systems with MASS. In *2017 IEEE Sensors Applications Symposium*, pages 1–6. IEEE, 2017.
- B. J. Frey y D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.
- J. Gama. *Knowledge Discovery from Data Streams*. CRC Press, 2010.

- J. Gama, P. P. Rodrigues, y L. Lopes. Clustering distributed sensor data streams using local processing and reduced communication. *Intelligent Data Analysis*, 15(1):3–28, 2011.
- J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, y A. Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):1–37, 2014.
- P. J. García-Laencina, J.-L. Sancho-Gómez, y A. R. Figueiras-Vidal. Pattern classification with missing data: A review. *Neural Computing and Applications*, 19(2):263–282, 2010.
- M. Gardner y J. Bieker. Data mining solves tough semiconductor manufacturing problems. In *Proceedings of the Sixth ACM International Conference on Knowledge Discovery and Data Mining*, pages 376–383. ACM Press, 2000.
- A. D. Gordon. A review of hierarchical classification. *Journal of the Royal Statistical Society: Series A (General)*, 150(2):119–137, 1987.
- P. J. F. Groenen y M. Velden. Multidimensional scaling. In *Wiley StatsRef: Statistics Reference Online*. American Cancer Society, wiley edition, 2014.
- M. P. Groover. *Fundamentals of Modern Manufacturing: Materials Processes, and Systems*. John Wiley & Sons, 2007.
- I. Guyon y A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
- M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, y I. H. Witten. The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.
- M. A. Hall. *Correlation-Based Feature Selection for Machine Learning*. PhD thesis, University of Waikato, 1999.
- S. Har-Peled y S. Mazumdar. On coresets for k-means and k-medians clustering. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, pages 291–300. ACM, 2004.
- D. V. Hinkley. Inference about the change-point from cumulative sum tests. *Biometrika*, 58(3):509–523, 1971.
- D. Isaacs, Javier Diaz-Rozo, A. Astarloa, y B. Arejita. Making factories smarter through machine learning. IIC Journal of Innovation, Industrial Internet Consortium, 2017.
- A. K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.
- Y. Jin y Y. Ishino. DAKA: Design activity knowledge acquisition through data-mining. *International Journal of Production Research*, 44(14):2813–2837, 2006.

- H. Kagermann, W. Wahlster, y J. Helbig. Securing the future of German manufacturing industry. Recommendations for implementing the strategic initiative Industrie 4.0. Technical report, National Academy of Science and Engineering (ACATECH), 2013.
- B. Kamp, A. Ochoa, y J. Diaz-Rozo. Smart servitization within the context of industrial user–supplier relationships: Contingencies according to a machine tool manufacturer. *International Journal on Interactive Design and Manufacturing*, 11(3):651–663, 2017.
- K. Kampa, E. Hasanbelliu, y J. C. Principe. Closed-form Cauchy-Schwarz PDF divergence for mixture of Gaussians. In *Proceedings of the 2011 International Joint Conference on Neural Networks*, pages 2578–2585. IEEE, 2011.
- P. Keerin, W. Kurutach, y T. Boongoen. Cluster-based KNN missing value imputation for DNA microarray data. In *2012 IEEE International Conference on Systems, Man, and Cybernetics*, pages 445–450, 2012.
- M. Kezunovic, Z. Obradovic, T. Dokic, B. Zhang, J. Stojanovic, P. Dehghanian, y P.-C. Chen. Predicting spatiotemporal impacts of weather on power systems using Big Data science. In *Data Science and Big Data: An Environment of Computational Intelligence*, pages 265–299. Springer, 2017.
- L. Kleinrock. *Information Flow in Large Communication Nets*. Doctoral dissertation, MIT, 1961.
- R. Kohavi y G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.
- S. Kolozali, D. Puschmann, M. Bermudez-Edo, y P. Barnaghi. On the effect of adaptive and nonadaptive analysis of time-series sensory data. *IEEE Internet of Things Journal*, 3(6):1084–1098, 2016.
- H. Kopetz. Internet of things. In *Real-Time Systems*, pages 307–323. Springer, 2011.
- P. Kranen, I. Assent, C. Baldauf, y T. Seidl. The ClusTree: Indexing micro-clusters for anytime stream mining. *Knowledge and Information Systems*, 29(2):249–272, 2011.
- P. Larrañaga, D. Atienza, J. Diaz-Rozo, A. Ogbechie, C. E. Puerto-Santana, y C. Bielza. *Industrial Applications of Machine Learning*. CRC Press, 2019.
- J. Lee, H.-A. Kao, y S. Yang. Service innovation and smart analytics for Industry 4.0 and big data environment. *Procedia CIRP*, 16:3–8, 2014.
- J. Lee, B. Bagheri, y H.-A. Kao. A cyber-physical systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3:18–23, 2015.
- K. Li, X. Zhang, J. Y.-T. Leung, y S.-L. Yang. Parallel machine scheduling problems in green manufacturing industry. *Journal of Manufacturing Systems*, 38:98–106, 2016.

- S. Li, L. D. Xu, y X. Wang. Compressed sensing signal and data acquisition in wireless sensor networks and internet of things. *IEEE Transactions on Industrial Informatics*, 9(4):2177–2186, 2013.
- T. W. Liao, D. M. Li, y Y. M. Li. Detection of welding flaws from radiographic images with fuzzy clustering methods. *Fuzzy Sets and Systems*, 108(2):145–158, 1999.
- S.-C. Lin y K.-C. Chen. Statistical QoS control of network coded multipath routing in large cognitive machine-to-machine networks. *IEEE Internet of Things Journal*, 3(4):619–627, 2016.
- S.-W. Lin, M. Crawford, y S. Mellor. The Industrial Internet of Things Reference Architecture. Technical Report Volume G1, Industrial Internet Consortium, 2017.
- J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.
- G. McLachlan y D. Peel. *Finite Mixture Models*. John Wiley & Sons, 2004.
- G. W. Milligan y M. C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, 1985.
- M. Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961.
- R. K. Mobley. *An Introduction to Predictive Maintenance*. Butterworth-Heinemann, 2002.
- H. Mouss, D. Mouss, N. Mouss, y L. Sefouhi. Test of Page-Hinckley, an approach for fault detection in an agro-alimentary production system. In *Proceedings of the 5th Asian Control Conference*, volume 2, pages 815–818, 2004.
- R. Mushtaq. Augmented Dickey Fuller test. SSRN Scholarly Paper ID 1911068, Social Science Research Network, Université Paris, 2011.
- S. Nandi, H. A. Toliyat, y X. Li. Condition monitoring and fault diagnosis of electrical motors—A review. *IEEE Transactions on Energy Conversion*, 20(4):719–729, 2005.
- A. Ogbechie, J. Diaz-Rozo, P. Larrañaga, y C. Bielza. Dynamic Bayesian network-based anomaly detection for in-process visual inspection of laser surface heat treatment. In *Machine Learning for Cyber Physical Systems*, pages 17–24. Springer, 2017.
- S. M. Pandit, D. Paul, y J. T. Roth. Monitoring and diagnosis of bearing defects using data dependent systems. *Integrated Computer-Aided Engineering*, 3(4):268–278, 1996.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, y E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

- J. A. Pérez y M. López. Design and implementation of an innovative quadratic Gaussian control system for laser surface treatments. *International Journal of Advanced Manufacturing Technology*, 65(9):1785–1790, 2013.
- J. A. Pérez, J. L. Ocaña, y C. Molpeceres. Hybrid fuzzy logic control of laser surface heat treatments. *Applied Surface Science*, 254(4):879–883, 2007.
- J. A. Perez, J. L. Ocaña, y C. Molpeceres. Real time fuzzy logic control of laser surface heat treatments. In *2007 IEEE International Symposium on Industrial Electronics*, pages 42–45, 2007.
- J. Posada, C. Toro, I. Barandiaran, D. Oyarzun, D. Stricker, R. de Amicis, E. B. Pinto, P. Eisert, J. Döllner, y I. Vallarino. Visual computing as a key enabling technology for Industrie 4.0 and industrial internet. *IEEE Computer Graphics and Applications*, 35(2): 26–40, 2015.
- PricewaterhouseCoopers. Innovation for the earth. Technical Report 161222-113251-LA-OS, World Economic Forum, Davos, 2017.
- D. Puschmann, P. Barnaghi, y R. Tafazolli. Adaptive clustering for dynamic IoT data streams. *IEEE Internet of Things Journal*, 4(1):64–74, 2017.
- B. Qian, L. Taimisto, A. Lehti, H. Piili, O. Nyrhilä, A. Salminen, y Z. Shen. Monitoring of temperature profiles and surface morphologies during laser sintering of alumina ceramics. *Journal of Asian Ceramic Societies*, 2(2):123–131, 2014.
- P. Rodrigues, J. Gama, y J. Pedroso. Hierarchical clustering of time-series data streams. *IEEE Transactions on Knowledge and Data Engineering*, 20(5):615–627, 2008.
- Y. Saeys, I. Insa, y P. Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.
- A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229, 1959.
- G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.
- R. Sebastiao y J. Gama. A study on change detection methods. In *Proceedings of the 14th Portuguese Conference on Artificial Intelligence*, pages 12–15, 2009.
- Y. M. Sebzalli y X. Z. Wang. Knowledge discovery from process operational data using PCA and fuzzy clustering. *Engineering Applications of Artificial Intelligence*, 14(5):607–616, 2001.
- J. Shi y J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

- J. E. Shigley y C. R. Mischke. *Standard Handbook of Machine Design*. McGraw-Hill, 1956.
- J. E. Shigley, R. G. Budynas, y C. R. Mischke. *Mechanical Engineering Design*. McGraw-Hill, 2004.
- A. Siddique, G. Yadava, y B. Singh. A review of stator fault monitoring techniques of induction motors. *IEEE Transactions on Energy Conversion*, 20(1):106–114, 2005.
- J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. de Carvalho, y J. Gama. Data stream clustering: A survey. *ACM Computing Surveys*, 46(1):13, 2013.
- M. Spiliopoulou, I. Ntoutsis, Y. Theodoridis, y R. Schult. MONIC: Modeling and monitoring cluster transitions. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 706–711. ACM Press, 2006.
- M. Strohbach, J. Daubert, H. Ravkin, y M. Lischka. Big data storage. In *New Horizons for a Data-Driven Economy*, pages 119–141. Springer, 2016.
- J. Sztipanovits, S. Ying, I. Cohen, D. Corman, J. Davis, H. Khurana, P. Mosterman, V. Prasad, y L. Stormo. Strategic R&D opportunities for 21st century cyber-physical systems. Technical report, Steering Committee for Foundation in Innovation for Cyber-Physical Systems, 2012.
- G. Taguchi y Y. Wu. *Introduction to Off-Line Quality Control*. Central Japan Quality Control Association, 1979.
- U. Thombansen, A. Gatej, y M. Pereira. Process observation in fiber laser-based selective laser melting. *Optical Engineering*, 54(1):011008, 2014.
- R. Tibshirani, G. Walther, y T. Hastie. Estimating the number of clusters in a data set via the GAP statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.
- D. A. Tobon-Mejia, K. Medjaher, y N. Zerhouni. CNC machine tool's wear diagnostic and prognostic by using dynamic Bayesian networks. *Mechanical Systems and Signal Processing*, 28:167–182, 2012.
- G. Tuna, D. G. Kogias, V. C. Gungor, C. Gezer, E. Taşkın, y E. Ayday. A survey on information security threats and solutions for machine to machine (M2M) communications. *Journal of Parallel and Distributed Computing*, 109:142–154, 2017.
- A. M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.
- Y. Wang, Y. Peng, Y. Zi, X. Jin, y K. Tsui. A two-stage data-driven-based prognostic approach for bearing degradation problem. *IEEE Transactions on Industrial Informatics*, 12(3):924–932, 2016.

- Y. Wang, G. Chen, y Z. Wang. A streaming data prediction method based on evolving Bayesian network. In *Web and Big Data*, volume 10367 of *Lecture Notes in Computer Science*, pages 294–302. Springer, 2017.
- O. Watanabe. Simple sampling techniques for discovery science. *IEICE Transactions on Information and Systems*, E83-D(1):19–26, 2000.
- D. Xu y Y. Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193, 2015.
- J. Yan, M. Koç, y J. Lee. A prognostic algorithm for machine performance assessment and its application. *Production Planning & Control*, 15(8):796–801, 2004.
- T. Zhang, R. Ramakrishnan, y M. Livny. BIRCH: A new data clustering algorithm and its applications. *Data Mining and Knowledge Discovery*, 1(2):141–182, 1997.
- A. Zhou, F. Cao, W. Qian, y C. Jin. Tracking clusters in evolving data streams over sliding windows. *Knowledge and Information Systems*, 15(2):181–214, 2008.