

UNIVERSIDAD DE VALLADOLID
ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería de Organización Industrial

Detección de anomalías en el proceso de
atornillado

Autor:
Manrique Román, Daniel

Tutor(es):
Galende Hernández, Marta
Departamento de Ingeniería de
Sistemas y Automática

Robles Martínez, Victor
Horse Powertrain Spain

Valladolid, noviembre de 2025.

RESUMEN

En este trabajo se presenta una propuesta de una metodología para la detección de anomalías en procesos de atornillado industrial utilizando técnicas de clustering sobre datos de sensores. El análisis se basa en las series temporales de par y ángulo registradas durante el ensamblaje, pudiendo identificar patrones anómalos en los datos. Se evalúan algoritmos especializados como TimeSeriesKMeans y K-Shape, aplicando métricas como Dynamic Time Warping y SoftDTW para la comparación de formas. También se proponen índices de evaluación de la calidad de los clústeres.

El método propuesto puede llegar a la detección de posibles fallos en el atornillado, contribuyendo al control de calidad y la confiabilidad de los procesos industriales. Además, la metodología es escalable y puede integrarse en sistemas de monitorización y mantenimiento predictivo, ayudando a reducir costes y mejorar la seguridad de la producción.

PALABRAS CLAVE

Atornillado industrial, detección de anomalías, clustering, series temporales, deformación temporal dinámica.

ABSTRACT

In this work, a proposal for a methodology for anomaly detection in industrial screwing processes is presented, using clustering techniques on sensor data. The analysis is based on time series of par and angle recorded during assembly, enabling the identification of anomalous patterns in the data. Specialized algorithms such as TimeSeriesKMeans and K-Shape are evaluated, applying metrics like Dynamic Time Warping and SoftDTW for shape comparison. Additionally, indices for evaluating the quality of the clusters are proposed.

The proposed method can lead to the detection of possible failures in screwing, contributing to quality control and the reliability of industrial processes. Furthermore, the methodology is scalable and can be integrated into monitoring and predictive maintenance systems, helping to reduce costs and improve production safety.

KEYWORDS

Industrial screwing, anomaly detection, clustering, time series, Dynamic Time Warping.

ÍNDICE

RESUMEN	2
ABSTRACT	2
ÍNDICE DE FIGURAS	6
ÍNDICE DE TABLAS	7
1. INTRODUCCIÓN Y OBJETIVOS	9
1.1. Introducción	9
1.2. Objetivos	12
1.3. Organización de la memoria	12
2. EL USO DEL CLUSTERING EN EL PROCESO DE ATORNILLADO	15
2.1. El proceso de atornillado	15
2.1.1. Curva de atornillado	15
2.1.2. Fases del atornillado	18
2.1.3. Tipologías de fallo en el atornillado	22
2.2. Uso de Clustering para Detección de Anomalías	24
2.2.1. Series temporales	24
2.2.2. TimeSeriesKMeans	26
2.2.3. KShape	30
2.2.4. Índices de valoración del clustering	33
2.2.5. Criterio de valoración	35
3. PROCESO PARA LA DETECCIÓN DE ANOMALÍAS	37
3.1. Adquisición de Datos de Series Temporales	38
3.2. Preprocesamiento	39
3.2.1. Limpieza de datos	40
3.2.2. Acondicionamiento de Series Temporales	40
3.3. Ejecución de algoritmos	41
3.3.1. Selección y Configuración de Algoritmos	41
3.3.2. Optimización de Parámetros	42
3.4. Análisis de resultados	42
3.4.1. Evaluación Cuantitativa	43
3.4.2. Análisis Cualitativo y Visualización	47
4. EXPERIMENTACIÓN	49
4.1. Caso de uso	49
4.2. Metodología experimental	50
4.2.1. Preprocesamiento	51

4.2.2.	Algoritmos de clustering	53
4.2.3.	Mejores configuraciones	55
4.3.	Resultados obtenidos.....	56
4.3.1.	[0.0, 0.0, 0.0, 0.5, 0.5]	57
4.3.2.	[0.0, 0.0, 0.0, 1.0, 0.0]	58
4.3.3.	[0.1, 0.1, 0.1, 0.5, 0.2].....	60
4.3.4.	[0.25, 0.25, 0.25, 0.0, 0.25].....	62
4.3.5.	[0.2, 0.2, 0.2, 0.2, 0.2].....	63
4.3.6.	Conclusión sobre los resultados	64
4.4.	Bibliotecas, Herramientas y Software Utilizados	70
5.	CONCLUSIONES Y TRABAJO FUTURO.....	75
5.1.	Conclusiones.....	75
5.2.	Líneas de Trabajo Futuro	76
	Bibliografía.....	79

ÍNDICE DE FIGURAS

Figura 1: Curva de par y ángulo. Fuente: Propia (Python)	16
Figura 2: Comparación de curvas bajo diferentes condiciones. Fuente: Propia (Python).....	17
Figura 3: Curva con ruido simulado. Fuente: Propia (Python).....	18
Figura 4: Curva par-ángulo y sus fases. Fuente: Propia (Python)	19
Figura 5: Fase de asentamiento. Fuente: Propia (Python)	19
Figura 6: Fase elástica. Fuente: Propia (Python)	20
Figura 7: Fase plástica. Fuente: Propia (Python)	20
Figura 8: Fase de fallo. Fuente: Propia (Python)	21
Figura 9: Diagrama de flujo. Fuente: Propia (LucidChart)	37
Figura 10: Diagrama del proceso de atornillado. Ángulo-tiempo. Fuente: Propia (Python) ..	38
Figura 11: Diagrama del proceso de atornillado. Par-tiempo. Fuente: Propia (Python)	39
Figura 12: Evolución de K_Index en función de k. Fuente: Propia (Python)	46
Figura 13: Ejemplo de procesamiento de datos. Fuente: Propia (Python).....	52
Figura 14: Máximos valores por algoritmo. Fuente: Propia (Python).....	65
Figura 15: Clustering Configuración 65. Fuente: Propia (Pyhton).....	67

ÍNDICE DE TABLAS

Tabla 1: Parámetros Clave TimeSeriesKMeans. Fuente: Propia	28
Tabla 2: Estadísticas descriptivas de las series temporales. Fuente: Propia (Python)	50
Tabla 3: Pseudocódigo asociado al preprocesamiento de datos. Fuente: Propia.....	52
Tabla 4: Pseudocódigo de evaluación de curvas par-ángulo. Fuente: Propia.....	54
Tabla 5: Pseudocódigo para la generación de resultados. Fuente: Propia	56
Tabla 6: Mejores configuraciones [0.0,0.0,0.0,0.5,0.5]. Fuente: Propia.....	57
Tabla 7: Configuración 65. Fuente: Propia.....	58
Tabla 8: Configuración 69. Fuente: Propia.....	58
Tabla 9: Mejores configuraciones [0.0,0.0,0.0,0.1,0.0,0]. Fuente: Propia.....	59
Tabla 10: Configuración 71. Fuente: Propia	59
Tabla 11: Configuración 55. Fuente: Propia	60
Tabla 12: Mejores configuraciones [0.1,0.1,0.1,0.5,0.2]. Fuente: Propia.....	61
Tabla 13: Configuración 68. Fuente: Propia	62
Tabla 14: Mejores configuraciones [0.25,0.25,0.25,0.0,0.25]. Fuente: Propia	62
Tabla 15: Configuración 63. Fuente: Propia	63
Tabla 16: Mejores configuraciones [0.2,0.2,0.2,0.2,0.2]. Fuente: Propia.....	64
Tabla 17: Configuración 72. Fuente: Propia	64
Tabla 18: : Mejores configuraciones globales. Fuente: Propia	66
Tabla 19: Desglose completo Configuración 65. Fuente: Propia.....	66
Tabla 20: Mejores configuraciones KShape. Fuente: Propia	70
Tabla 21: Configuración de los parámetros: Fuente: Propia.....	71

1. INTRODUCCIÓN Y OBJETIVOS

1.1. Introducción

La fiabilidad de las uniones atornilladas es un aspecto crítico en la industria moderna, ya que los tornillos y pernos constituyen uno de los sistemas de ensamblaje más utilizados en sectores como la automoción, la aeronáutica, la construcción y la fabricación de maquinaria. Sin embargo, a pesar de su aparente sencillez, el atornillado es una fuente frecuente de problemas técnicos, económicos y de seguridad, lo que justifica la necesidad de abordar en profundidad el estudio de los fallos asociados a este tipo de unión y sus consecuencias.

El problema de los fallos en el atornillado se manifiesta en múltiples formas, desde el aflojamiento prematuro y la pérdida de tensión hasta la rotura del tornillo o el daño en la unión atornillada. Estas incidencias pueden deberse a errores en el proceso de montaje (sub-apriete, sobre-apriete, cruce de roscas), a la selección inadecuada de materiales, a la falta de mantenimiento o a la exposición a condiciones de servicio agresivas, como vibraciones, cargas dinámicas o ambientes corrosivos [1]. El resultado es una disminución de la integridad estructural y funcional de los conjuntos mecánicos, que puede derivar en costosos paros de producción, reparaciones no planificadas y, en casos extremos, accidentes graves con consecuencias humanas y económicas.

Desde el punto de vista económico, los fallos en el atornillado representan una de las principales fuentes de costes ocultos en la industria. Según estudios sobre gestión de la calidad y mantenimiento, los costes asociados a fallos internos (detectados antes de la entrega al cliente) y externos (detectados tras la puesta en servicio) pueden superar ampliamente el coste inicial de fabricación, especialmente cuando implican la retirada de productos, la reparación en campo o la gestión de reclamaciones [2]. Además, los fallos en uniones atornilladas suelen tener un efecto multiplicador sobre otros sistemas, generando daños colaterales en componentes adyacentes y aumentando la complejidad de las intervenciones correctivas.

En términos de seguridad, los fallos en el atornillado pueden tener consecuencias especialmente graves en aplicaciones críticas. En el sector de la automoción, como en el caso de Horse Powertrain Spain (empresa dedicada a la fabricación de sistemas de transmisión para movilidad), un fallo en la unión de un componente estructural o de seguridad puede poner en riesgo el correcto funcionamiento de los sistemas de transmisión. Por ello, la gestión del riesgo y la prevención de fallos mediante metodologías como el Análisis Modal de Fallos y Efectos (AMFE) se han convertido en prácticas habituales para garantizar la seguridad y la conformidad con los estándares de calidad [3].

La calidad del producto final también se ve directamente afectada por la fiabilidad de las uniones atornilladas. Un ensamblaje defectuoso puede traducirse en ruidos, vibraciones, fugas, pérdida de prestaciones o reducción de la vida útil del producto, lo que impacta negativamente en la satisfacción del cliente y la reputación de la empresa. Además, en un contexto de competencia global y mejora continua, la prevención de fallos en el atornillado es un requisito imprescindible para alcanzar la excelencia operativa y cumplir con las exigencias de certificaciones internacionales de calidad y seguridad, como la ISO 9001 [4] o la IATF.

Por todo ello, la problemática de los fallos en el atornillado y sus consecuencias constituye un campo de estudio prioritario en la ingeniería industrial y el mantenimiento. La identificación, análisis y prevención de estos fallos no solo permite reducir costes y mejorar la calidad, sino que también contribuye de manera decisiva a la seguridad de los productos y procesos. El presente trabajo, desarrollado en colaboración con Horse Powertrain Spain, se plantea, por tanto, como una contribución al conocimiento y la mejora de las técnicas de control y diagnóstico de fallos en uniones atornilladas, con el objetivo final de minimizar su impacto en la industria y la sociedad.

La aplicación de técnicas de analítica de datos y, en particular, de clustering, puede ser una estrategia eficaz para abordar la problemática de los fallos en el atornillado industrial. Esta problemática, caracterizada por la alta variabilidad de los procesos, la presencia de grandes volúmenes de datos y la dificultad para identificar patrones anómalos de forma manual, requiere enfoques automáticos y escalables que permitan detectar, clasificar y prevenir defectos en tiempo real.

Uno de los principales retos en el control de calidad de procesos de atornillado es la detección temprana de anomalías entre miles o millones de operaciones, donde los fallos son eventos poco frecuentes y, a menudo, presentan características sutiles o atípicas respecto al comportamiento estándar [1]. Tradicionalmente, la supervisión se ha basado en la inspección visual, el análisis de curvas par-ángulo o la aplicación de umbrales fijos sobre variables como el par o el ángulo final. Sin embargo, estos métodos resultan insuficientes ante la complejidad y el volumen de datos generados en entornos industriales modernos, donde la digitalización y la sensorización masiva han dado lugar a escenarios de big data [5] [6].

En este contexto, la analítica de datos basada en clustering permite agrupar automáticamente grandes cantidades de registros de atornillado según su similitud, sin necesidad de etiquetas previas ni intervención humana. Los algoritmos de clustering, como K-Means, DBSCAN, K-Shape, identifican patrones de comportamiento normal y detectan grupos de operaciones que se desvían del estándar, facilitando la identificación de fallos potenciales o condiciones anómalas [7] [8]. Esta capacidad es especialmente relevante cuando los datos presentan alta variabilidad, ruido o cuando los fallos no se manifiestan de forma evidente en variables individuales.

El uso de clustering en la detección de fallos en atornillado tiene varias ventajas clave:

- No requiere datos etiquetados: A diferencia del aprendizaje supervisado, el clustering es un método no supervisado, lo que lo hace ideal en entornos donde no se dispone de grandes bases de datos de fallos conocidos [9].
- Permite descubrir patrones desconocidos: Al no estar limitado por categorías predefinidas, el clustering puede revelar nuevos tipos de fallos, degradaciones progresivas o cambios en el proceso que pasarían desapercibidos con métodos tradicionales [8].
- Escalabilidad y eficiencia: Los algoritmos modernos permiten analizar grandes volúmenes de datos en tiempo real, adaptándose a la producción continua y a la monitorización masiva de líneas de montaje [7].
- Soporte a la toma de decisiones: Los resultados de clustering pueden integrarse en sistemas de alerta, mantenimiento predictivo o visualización avanzada para que los operarios y responsables de calidad puedan actuar de forma proactiva.

En la literatura reciente, se han desarrollado soluciones que combinan el clustering con técnicas de extracción de características y aprendizaje profundo para mejorar la detección de defectos en procesos industriales. Por ejemplo, en la industria del acero, la combinación de clustering y redes neuronales ha permitido reducir la tasa de fallos de clasificación a solo un 4%, optimizando la inspección y reduciendo la necesidad de intervención manual [9].

En el caso concreto del atornillado, el clustering aplicado a curvas par-ángulo, señales de par o variables de proceso permite segmentar automáticamente los ciclos en grupos homogéneos, identificar operaciones defectuosas y proporcionar información explicativa a los operarios mediante herramientas de inteligencia artificial explicable. Esta aproximación facilita la mejora continua del proceso, la reducción de costes por defectos y la mejora de la seguridad y la calidad del producto final.

En resumen, la analítica de datos basada en clustering se justifica plenamente como una solución avanzada y eficiente para el diagnóstico y la prevención de fallos en el atornillado industrial, aportando valor tanto en la detección temprana como en la comprensión y mejora de los procesos.

1.2. Objetivos

Objetivo general

Analizar y proponer soluciones basadas en analítica de datos, especialmente técnicas de clustering, para la detección y prevención de fallos en procesos de atornillado industrial.

Objetivos específicos

- Recopilar y analizar datos reales de procesos de atornillado, incluyendo curvas par-ángulo, par, para caracterizar el comportamiento normal y anómalo del proceso.
- Implementar y comparar diferentes algoritmos de clustering para la segmentación y análisis de series temporales asociadas a procesos de atornillado, evaluando su capacidad para identificar patrones anómalos y optimizar la detección de fallos.
- Comparar el rendimiento y la capacidad de discriminación de distintas métricas de similitud para series temporales, evaluando su impacto en la identificación de fallos.
- Analizar el impacto de la selección y ajuste de los parámetros clave de los algoritmos de clustering sobre la calidad de la segmentación y la robustez en la detección de patrones anómalos en datos industriales de atornillado.
- Diseñar y validar dos índices métricos propios para evaluar la calidad del agrupamiento y la detección de anomalías en los procesos de atornillado industrial.
- Desarrollar un mecanismo de toma de decisión basado en un operador de agregación para seleccionar las mejores configuraciones de clustering, integrando métricas cuantitativas y facilitando la elección óptima en la detección de anomalías.

1.3. Organización de la memoria

La memoria de este TFG se estructura en 5 capítulos, comenzando por la introducción y los objetivos de este mismo. En el Capítulo 2, se explora el marco teórico y el estado del arte, cubriendo el proceso de atornillado y las técnicas de clustering. El Capítulo 3 se centrará en la explicación de la solución propuesta y sus distintos criterios y validaciones. En el Capítulo 4, se detalla la experimentación del proyecto, incluyendo el caso de uso, la metodología (con sus pasos y la implementación de los algoritmos), los resultados obtenidos junto con sus tablas y

conclusiones y las herramientas utilizadas. El Capítulo 5 se cerrará con conclusiones y líneas futuras en las que poder continuar con el trabajo. Finalmente, se detalla la bibliografía utilizada para la redacción del trabajo.

Durante la elaboración de este trabajo se ha contado con el apoyo de una herramienta de generación de textos basada en inteligencia artificial, concretamente Perplexity AI, para la redacción inicial y simplificación de algunas secciones. Además se ha utilizado Grok para la aplicación de algunas funciones del código.

2. EL USO DEL CLUSTERING EN EL PROCESO DE ATORNILLADO

2.1. El proceso de atornillado

El proceso de atornillado es una técnica de unión mecánica ampliamente utilizada en la industria debido a su versatilidad, facilidad de implementación y capacidad para permitir el desmontaje y reutilización de componentes. Este procedimiento, fundamental en sectores como la automoción, la aeronáutica, la electrónica y la maquinaria pesada, consiste en aplicar un par de apriete a un tornillo para generar una fuerza de sujeción que garantice la integridad estructural y funcional de las piezas unidas. La calidad de una unión atornillada depende de múltiples factores, como la selección adecuada de materiales, el control preciso del par y el ángulo de giro, y la monitorización de las fases del proceso, lo que ha impulsado el desarrollo de herramientas y metodologías avanzadas para su análisis y optimización.

Empezaremos explorando la curva de atornillado, una representación gráfica del par frente al ángulo que revela cómo se comporta una unión durante el proceso. Se describirán sus fases principales, los parámetros clave para su análisis y su importancia en el control de calidad industrial. A continuación, se abordarán las tipologías de fallo en el atornillado, identificando problemas comunes como el sub-apriete, el sobre-apriete, el aflojamiento, la rotura, el cruce de roscas y la corrosión. Finalmente, se analizará el uso de clustering para detectar anomalías, centrandonos en el análisis de series temporales y algoritmos como TimeSeriesKMeans (TSKM) y K-Shape, que ayudan a identificar patrones y desviaciones, mejorando la eficiencia y la calidad en procesos industriales automatizados. Se incluirán también la explicación de los índices de validación del clustering (Silhouette, Calinski-Harabasz, Davies-Bouldin) y el operador de agregación WOWA (Weighted Ordered Weighted Average) para evaluar la calidad de los agrupamientos.

2.1.1. Curva de atornillado

La curva de atornillado, también conocida como curva par-ángulo, es hoy en día una pieza clave para vigilar y mejorar los procesos de unión por tornillo en la industria. En esencia, grafica el par de apriete frente al ángulo de giro del tornillo mientras avanzamos en el atornillado. Gracias a ella podemos interpretar cómo se comporta la unión, detectar fallos incipientes y ajustar los parámetros de montaje para obtener resultados más fiables [6] [7].

En la práctica, esta curva se genera con atornilladoras equipadas con sensores de par y de ángulo que recogen datos en cada instante. Contar con esa información en tiempo real resulta imprescindible para asegurar la calidad de la unión, sobre todo en sectores exigentes como la automoción, la electrónica o la maquinaria pesada, donde un error puede tener consecuencias muy costosas [10].

Fases características de la curva par-ángulo

Una curva par-ángulo típica presenta varias fases bien diferenciadas. Al inicio del proceso, en la fase de asentamiento, el tornillo entra en contacto con la superficie de la pieza y comienza a girar, registrándose un aumento progresivo del ángulo con un par relativamente bajo. Esta fase corresponde al ajuste inicial y a la eliminación de holguras entre las superficies en contacto [7].

A continuación, se desarrolla la fase elástica, donde el material bajo la cabeza del tornillo y en la rosca comienza a deformarse de manera reversible. En este tramo, la relación entre el par y el ángulo suele ser aproximadamente lineal, reflejando la rigidez del conjunto. Finalmente, se alcanza la fase plástica, caracterizada por un aumento más lento del par a medida que el material cede y se produce deformación irreversible. El punto máximo de la curva corresponde al par de fallo, seguido en ocasiones por una caída brusca que indica la rotura del tornillo o del material [11].

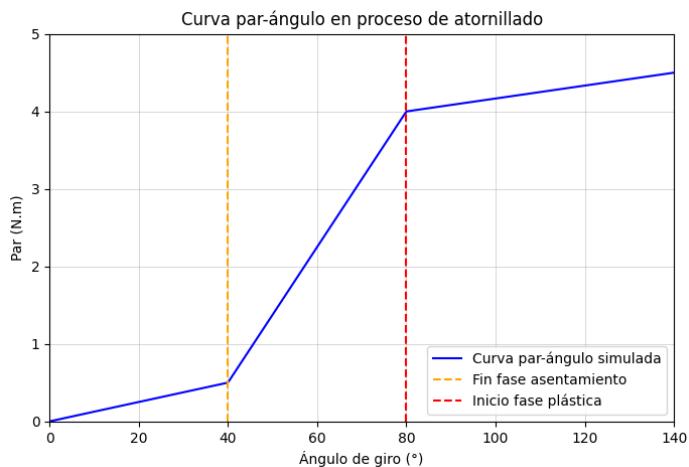


Figura 1: Curva de par y ángulo. Fuente: Propia (Python)

El análisis detallado de la curva par-ángulo (**Figura 1**) permite extraer parámetros críticos para el control de calidad y la optimización del proceso de atornillado. Entre los más relevantes se encuentran al final de la fase de asentamiento, el par máximo o de fallo, el ángulo de fluencia (punto donde comienza la deformación plástica) y la pendiente de la fase elástica (indicativa de la rigidez de la unión) [11].

La interpretación de estos parámetros es esencial para detectar anomalías como el sub-apriete (par máximo bajo y pendiente reducida), el sobre-apriete (par excesivo y ángulo de fluencia bajo), o defectos como el cruce de roscas, que suele manifestarse como oscilaciones o irregularidades en la curva (**Figura 2**). En entornos industriales, estas curvas registradas se comparan automáticamente con plantillas de referencia. Cualquier desviación importante dispara una alerta en tiempo real y permite descartar la pieza defectuosa antes de continuar con la línea de producción [7].

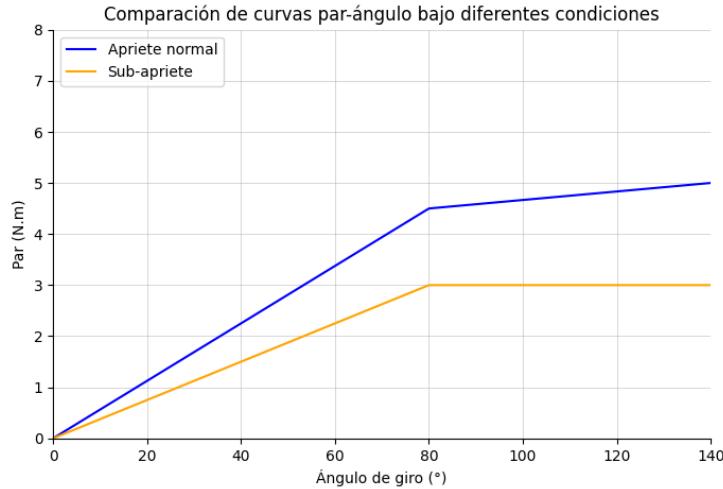


Figura 2: Comparación de curvas bajo diferentes condiciones. Fuente: Propia (Python)

El uso sistemático de la curva par-ángulo es clave en la automatización del control de calidad en líneas de montaje. Por ejemplo, en la fabricación de inversores para automoción, la optimización del par de apriete basada en el análisis de la curva ha permitido reducir significativamente los fallos por vibración y desconexión de componentes, ajustando el par óptimo para cada tipo de unión [11].

Cuando hablamos de montaje automatizado con varios tornillos, entran en juego sistemas de visión artificial y herramientas de análisis de señal. Estas soluciones capturan cada curva de par-ángulo y la cotejan al instante con un patrón de referencia. Para medir cuán parecidas son, se aplican métricas como la distancia de Fréchet, que cuantifica de forma precisa cualquier variación sutil. Al detectar una desviación significativa, el sistema alerta automáticamente, lo que abre la puerta a un control predictivo y evita costes extra derivados de piezas mal montadas [7].

Más allá de la recogida de datos, también podemos anticipar el comportamiento de la unión gracias a modelos matemáticos y simulaciones (**Figura 3**). Por un lado, ajustando funciones gaussianas a la curva; por otro, recurriendo a métodos de elementos finitos. Esto resulta especialmente útil en aplicaciones exigentes, como las bombas hidráulicas, donde identificar el par de apriete ideal puede minimizar la deformación geométrica y alargar la vida útil del conjunto atornillado [6].

En la literatura técnica se han propuesto modelos que describen la curva como una combinación de fases elásticas y plásticas, permitiendo ajustar parámetros como la rigidez, la resistencia máxima y la ductilidad del sistema. Estos modelos son especialmente útiles para el diseño de uniones críticas y la validación de procedimientos de montaje en sectores como la aeronáutica o la automoción [6].

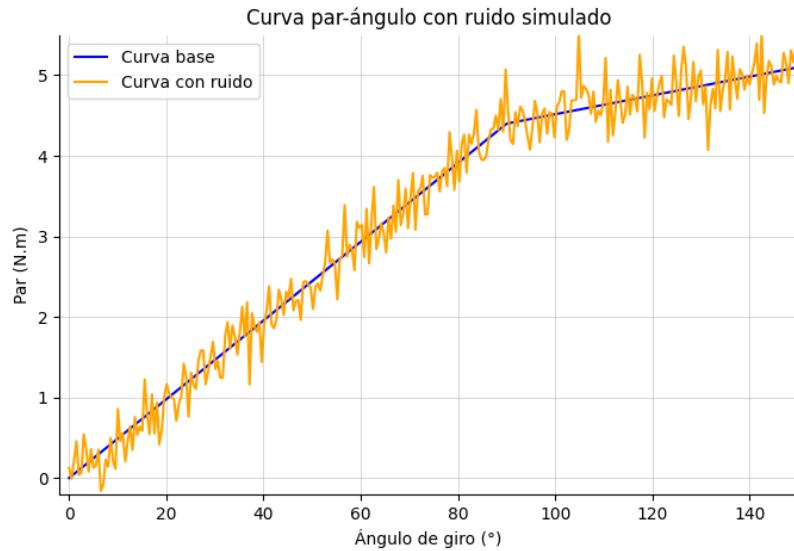


Figura 3: Curva con ruido simulado. Fuente: Propia (Python)

En definitiva, la curva par-ángulo es una herramienta indispensable para el análisis, control y optimización de procesos de atornillado en la industria moderna. Su interpretación permite garantizar la integridad estructural de las uniones, detectar defectos en tiempo real y reducir costes de fabricación. La integración de sensores avanzados, algoritmos de análisis de señales y técnicas de inteligencia artificial abre nuevas posibilidades para el control automático y predictivo de la calidad en procesos de ensamblaje.

2.1.2. Fases del atornillado

En este subcapítulo se describen las etapas principales que caracterizan el apriete del tornillo, desde el contacto inicial, pasando por la deformación elástica y plástica del material, hasta el momento crítico de falla o rotura.

La **Figura 4** ilustra la curva par-ángulo junto con sus fases características que se tratan posteriormente.

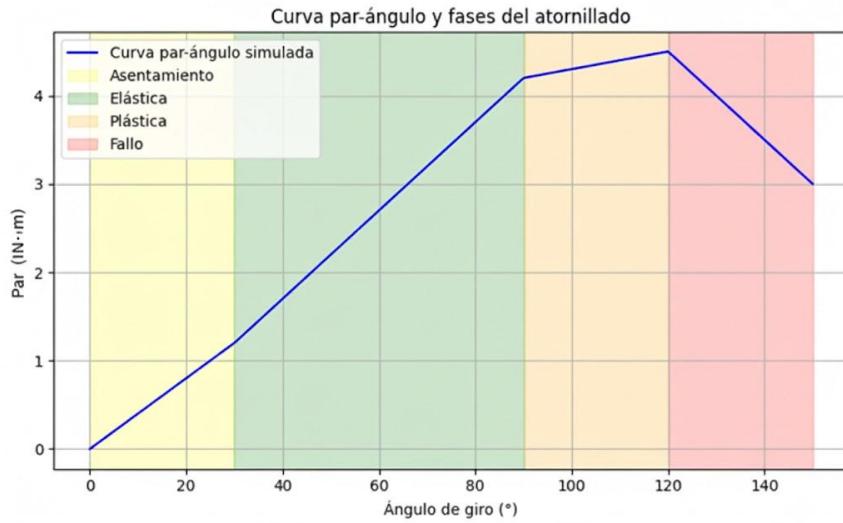


Figura 4: Curva par-ángulo y sus fases. Fuente: Propia (Python)

1. Fase de asentamiento

La primera etapa del atornillado es la **fase de asentamiento (Figura 5)**. En esta fase, el tornillo comienza a girar y entra en contacto con la superficie de la pieza a unir. Inicialmente, el par de apriete (par) es bajo y el ángulo de giro aumenta rápidamente. Durante este periodo, se eliminan las holguras entre las superficies, se ajustan las imperfecciones y se acomodan las rugosidades de contacto. El objetivo principal de esta fase es lograr el contacto efectivo entre las superficies, asegurando que el tornillo y la rosca estén correctamente alineados y que la fuerza de fricción comience a actuar [7].



Figura 5: Fase de asentamiento. Fuente: Propia (Python)

2. Fase de apriete (fase elástica)

Una vez superado el asentamiento, el proceso entra en la **fase de apriete** o **fase elástica** (Figura 6). Aquí, el par de apriete comienza a incrementarse de manera más pronunciada en relación con el ángulo de giro. El material bajo la cabeza del tornillo y en la rosca se deforma elásticamente, es decir, de forma reversible. En esta fase, la relación entre el par y el ángulo es aproximadamente lineal, lo que refleja la rigidez del conjunto tornillo-material. El objetivo es alcanzar el nivel de tensión necesario para asegurar la unión sin exceder el límite elástico del material [11].

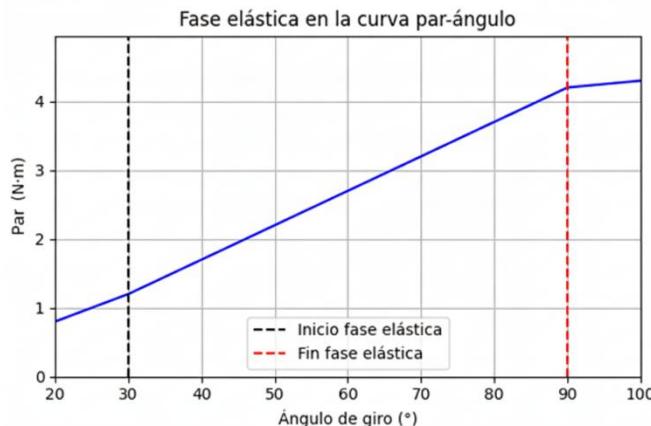


Figura 6: Fase elástica. Fuente: Propia (Python)

3. Fase de fluencia (fase plástica)

Si se continúa aplicando par más allá del límite elástico, se entra en la **fase de fluencia** o **fase plástica** (Figura 7). En este tramo, el material comienza a deformarse de manera irreversible. El par de apriete sigue aumentando, pero la pendiente de la curva disminuye, indicando que el ángulo de giro crece más rápidamente respecto al aumento de par. Esta fase es crítica, ya que una deformación plástica excesiva puede comprometer la integridad de la unión y llevar al fallo del tornillo o del material unido [6].

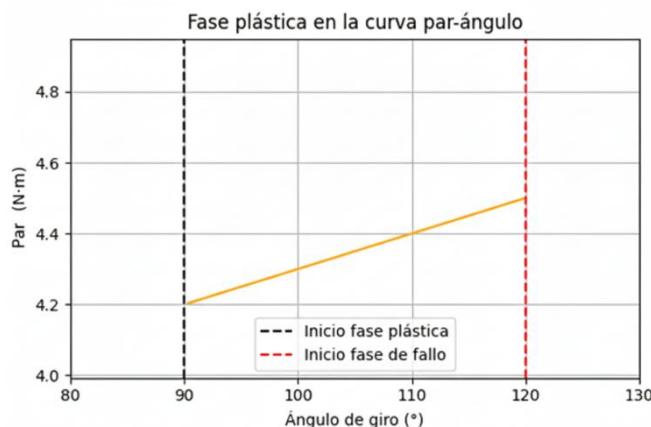


Figura 7: Fase plástica. Fuente: Propia (Python)

4. Fase de fallo

Finalmente, si el proceso de apriete continúa, se alcanza el **par máximo o par de fallo** (**Figura 8**). Este punto representa la máxima resistencia que puede soportar la unión antes de que ocurra la rotura del tornillo, el deslizamiento de la rosca o la fractura del material. Tras este punto, el par puede disminuir abruptamente, aunque el ángulo siga aumentando, lo que indica la pérdida total de la capacidad de carga de la unión [11].



Figura 8: Fase de fallo. Fuente: Propia (Python)

El análisis detallado de estas fases permite detectar anomalías como sub-apriete (cuando no se alcanza la fase elástica adecuada), sobre-apriete (cuando se excede la fase plástica) o defectos como el cruce de roscas, que pueden manifestarse como irregularidades en la curva par-ángulo. La monitorización en tiempo real de estas fases es esencial para la automatización del control de calidad y la reducción de defectos en la producción industrial [7].

El conocimiento profundo de las fases del atornillado permite optimizar los procesos de montaje, seleccionar los parámetros de apriete adecuados y prevenir fallos mecánicos costosos. La representación gráfica mediante la curva par-ángulo y el análisis de sus distintas fases constituyen herramientas imprescindibles para ingenieros y técnicos en la industria moderna.

2.1.3. Tipologías de fallo en el atornillado

El atornillado es uno de los métodos de unión más utilizados en la industria, tanto por su sencillez como por la posibilidad de desmontaje y reutilización de componentes. Sin embargo, la fiabilidad de una unión atornillada depende de numerosos factores y, en la práctica, pueden aparecer diferentes tipologías de fallo que comprometen la seguridad, la funcionalidad o la durabilidad del conjunto. El conocimiento y la prevención de estos fallos es esencial para el diseño, la fabricación y el mantenimiento de cualquier sistema mecánico [1].

El análisis de las tipologías de fallo en el atornillado permite establecer procedimientos de control de calidad, seleccionar los pares de apriete adecuados y diseñar sistemas de monitorización que detecten anomalías en tiempo real. El uso de curvas par-ángulo, el análisis estadístico de los procesos y la formación del personal son herramientas clave para reducir la incidencia de fallos y mejorar la fiabilidad de las uniones atornilladas [1].

Fallo por sub-apriete

El sub-apriete ocurre cuando el par aplicado al tornillo es insuficiente para garantizar la correcta sujeción de las piezas. Esto puede deberse a un error en el ajuste de la herramienta, a la falta de control en el proceso o a la presencia de lubricantes o contaminantes en la rosca. El principal riesgo del sub-apriete es la aparición de holguras, que pueden derivar en movimientos relativos, ruidos, pérdida de estanqueidad y, en casos extremos, el desensamblaje espontáneo de la unión. Además, el sub-apriete favorece la fatiga por micro-movimientos, acelerando el desgaste de las superficies de contacto [1].

Fallo por sobre-apriete

El sobre-apriete se produce cuando el par aplicado excede el valor recomendado para el tornillo o el material de las piezas a unir. Este exceso de carga puede provocar la deformación plástica del tornillo, el deterioro de la rosca o incluso la rotura por tracción o cizalladura. El sobre-apriete también puede inducir tensiones residuales que, a largo plazo, favorecen la aparición de grietas o la fatiga del material. En aplicaciones críticas, como la automoción o la aeronáutica, el sobre-apriete es una de las principales causas de fallo catastrófico de uniones atornilladas [6].

Fallo por aflojamiento

El aflojamiento de los tornillos es un problema habitual en uniones que están sometidas a vibraciones, cargas dinámicas o variaciones térmicas. Este fenómeno puede deberse a un apriete insuficiente al inicio o a la relajación progresiva de las tensiones internas, provocada por la fluencia de los materiales o por deformaciones en las superficies de contacto. Cuando se produce un aflojamiento progresivo, la fuerza de sujeción disminuye y, con el tiempo, puede perderse completamente la

función estructural de la unión. Para prevenir esta situación, es común utilizar elementos como arandelas de seguridad, adhesivos especiales o sistemas de bloqueo mecánico [1].

Fallo por rotura del tornillo

La rotura del tornillo representa uno de los fallos más críticos en uniones atornilladas. Generalmente, está asociada a un exceso de apriete, defectos de fabricación —como microgrietas o inclusiones en el material— o a la aplicación de cargas superiores a las especificadas en el diseño. Dependiendo del tipo de esfuerzo predominante, la rotura puede producirse por tracción, cizalladura o fatiga. En muchos casos, la fractura comienza en zonas de alta concentración de tensiones, como la raíz de la rosca o la transición entre la cabeza y el vástago [11].

Fallo por cruce de roscas

El cruce de roscas se produce cuando el tornillo no sigue el camino correcto en la rosca de la pieza, generando un contacto anómalo entre los filetes. Esto puede deberse a un desalineamiento durante el montaje, a la presencia de suciedad o a defectos en la fabricación de las roscas. El cruce de roscas provoca un aumento irregular del par, daños irreversibles en la rosca y, en muchos casos, la imposibilidad de desmontar el conjunto sin destruirlo [1].

Fallo por corrosión y desgaste

La corrosión y el desgaste son mecanismos de fallo a largo plazo que afectan especialmente a uniones expuestas a ambientes agresivos o a movimientos relativos entre las piezas. La corrosión puede debilitar el tornillo y la rosca, facilitando el aflojamiento o la rotura, mientras que el desgaste reduce la capacidad de transmisión de carga y puede generar holguras. La selección adecuada de materiales y recubrimientos, así como el mantenimiento preventivo, son esenciales para minimizar estos riesgos [6].

Las uniones atornilladas, a pesar de su aparente simplicidad, pueden fallar por múltiples mecanismos si no se controlan adecuadamente los parámetros de montaje y las condiciones de servicio. La identificación y prevención de los distintos tipos de fallo es fundamental para garantizar la seguridad y la durabilidad de los productos industriales.

2.2. Uso de Clustering para Detección de Anomalías

2.2.1. Series temporales

Las series temporales constituyen una de las estructuras de datos más relevantes y frecuentes en el ámbito científico, tecnológico e industrial. Se definen como una secuencia ordenada de observaciones recogidas a intervalos regulares de tiempo, lo que permite analizar la evolución de un fenómeno, identificar patrones, realizar pronósticos y detectar anomalías [10]. A diferencia de otros tipos de datos, las series temporales incorporan explícitamente la dimensión temporal, lo que implica que el orden y la dependencia entre los datos son aspectos fundamentales para su análisis.

El estudio de series temporales es esencial en campos tan variados como la economía (por ejemplo, para el análisis de precios y mercados) [12], la meteorología (temperaturas, precipitaciones) [13], la energía [14], la medicina (señales biomédicas como el electrocardiograma) [15], la ingeniería (monitorización de maquinaria) [16], y muchos otros. La capacidad de modelar y comprender la dinámica temporal de estos procesos permite no solo describir su comportamiento pasado, sino también anticipar su evolución futura y tomar decisiones fundamentadas.

El análisis de series temporales tiene particularidades que lo diferencian del tratamiento convencional de datos. Entre sus aspectos más relevantes se encuentran:

- **Dependencia temporal:** Los valores de la serie suelen estar influenciados por los anteriores, lo que se conoce como autocorrelación. Esta propiedad es clave para el desarrollo de modelos que intentan predecir el comportamiento futuro.
- **Tendencias y estacionalidades:** Muchas series muestran patrones repetitivos o cambios graduales a lo largo del tiempo. Detectar estos componentes permite comprender mejor el sistema y construir modelos más precisos.
- **Presencia de ruido y valores atípicos:** Las series pueden contener ruido aleatorio o anomalías que enmascaran la señal real. Es necesario aplicar técnicas que filtren estos elementos para evitar interpretaciones erróneas.
- **No estacionalidad:** Algunas series cambian su comportamiento estadístico con el tiempo, lo que dificulta su modelado. En estos casos, se suelen aplicar transformaciones para estabilizar la varianza o la media, o se opta por modelos específicos.

Debido a estos desafíos, se han desarrollado múltiples enfoques, que van desde métodos estadísticos tradicionales —como ARIMA o los modelos de suavizado— hasta

técnicas más modernas basadas en inteligencia artificial y aprendizaje profundo [10].

Las series temporales son fundamentales para entender cómo evolucionan muchos procesos reales. En la industria, su análisis permite monitorizar operaciones, anticipar fallos en equipos, detectar comportamientos anómalos y mejorar la eficiencia en la producción. Por ejemplo, en el sector energético se utilizan para prever el consumo y optimizar la distribución de recursos [14]. En medicina, el análisis de señales como el ECG (Electrocardiografía) o las ondas cerebrales contribuye al diagnóstico y seguimiento clínico [15].

Debido a la digitalización y la expansión del Internet de las Cosas (IoT), la cantidad de datos temporales ha crecido enormemente. Esto ha hecho imprescindible contar con herramientas avanzadas que permitan aprovechar esta información. Algoritmos como K-Shape o TimeSeriesKMeans han demostrado ser eficaces para clasificar series similares y extraer patrones útiles para la toma de decisiones [17].

Las técnicas para trabajar con series temporales son diversas y se pueden agrupar en varias categorías:

- **Modelos tradicionales:** Incluyen métodos como AR (AutoRegressive), MA (Moving Average), ARMA (AutoRegressive Moving Average), ARIMA (AutoRegressive Integrated Moving Average) o SARIMA (Seasonal AutoRegressive Integrated Moving Average). Todos ellos utilizan combinaciones lineales de valores anteriores y errores para representar la evolución de la serie.
- **Modelos de descomposición:** Separan la serie en componentes como tendencia, estacionalidad y ruido. Esta descomposición facilita tanto el análisis como la elaboración de predicciones.
- **Técnicas de suavizado:** Métodos como el suavizado exponencial son útiles especialmente para hacer predicciones a corto plazo o cuando la serie presenta poca estructura.
- **Modelos multivariantes:** Técnicas como VAR (Vector AutoRegressive) o la cointegración permiten analizar series relacionadas entre sí, ofreciendo una visión más completa del sistema.
- **Enfoques de machine learning:** Aquí se encuentran algoritmos de clasificación, agrupamiento y redes neuronales, especialmente las de tipo recurrente. Estas técnicas resultan muy eficaces para captar relaciones complejas y anticipar eventos futuros.

Cada tipo de enfoque tiene sus propias ventajas, y su elección depende del tipo de datos disponibles, del comportamiento de la serie y de los objetivos específicos del análisis.

El análisis de series temporales es una disciplina en constante evolución. Actualmente, la integración de técnicas de machine learning y deep learning está

permitiendo abordar problemas cada vez más complejos, como la predicción en tiempo real, la detección automática de anomalías y el clustering de grandes volúmenes de datos temporales. Además, la combinación de análisis de series temporales con otras fuentes de datos (por ejemplo, datos espaciales o contextuales) está abriendo nuevas posibilidades en áreas como la industria 4.0, la salud digital y la gestión inteligente de infraestructuras.

En resumen, el análisis de series temporales es una herramienta imprescindible para entender, modelar y anticipar el comportamiento de sistemas dinámicos en un mundo cada vez más conectado y orientado a los datos.

2.2.2. TimeSeriesKMeans

El algoritmo TimeSeriesKMeans es una extensión especializada del método K-Means clásico diseñada para series temporales, capaz de manejar la dimensionalidad temporal y la variabilidad en la longitud de los datos [18]. A diferencia de los algoritmos de clustering tradicionales que operan en espacios euclidianos estáticos, TimeSeriesKMeans incorpora métricas específicas para secuencias temporales, como la Dynamic Time Warping (DTW) y su variante diferenciable softDTW, permitiendo comparar patrones con deformaciones temporales no lineales [19]. Esta capacidad es crucial en procesos industriales como el atornillado, donde las señales de par-ángulo presentan variaciones en duración y velocidad [10].

2.2.2.1. Dynamic Time Warping (DTW)

DTW calcula la similitud entre dos series temporales $X = (x_1, x_2, \dots, x_n)$ e $Y = (y_1, y_2, \dots, y_n)$ mediante un alineamiento no lineal que minimiza la distancia acumulativa: $DTW(X, Y) = \min_{\pi} \sqrt{\sum_{(i,j) \in \pi} \phi(x_i, y_j)}$

donde:

- π es una trayectoria de alineación que satisface:
 - **Monotonidad:** $i_k \leq i_{k+1}, j_k \leq j_{k+1}$
 - **Condiciones de frontera:** $\pi_1 = (1,1), \pi_k = (n, m)$
 - **Continuidad:** $i_{k+1} - i_k \leq 1, j_{k+1} - j_k \leq 1$
- La función de costo ϕ suele ser la distancia euclídea $\phi(x_i, y_j) = \|x_i - y_j\|^2$. Esta métrica es invariantes a compresiones/expansiones temporales, esencial para comparar ciclos de atornillado con diferentes velocidades. [20]

Esta métrica hace que el agrupamiento sea mucho más robusto, ya que detecta patrones similares, aunque ocurran en momentos distintos o a distintas velocidades, pudiendo detectar comportamientos anómalos o patrones repetitivos en procesos industriales.

Los centroides de los clústeres no se calculan como simples medias, sino mediante un promedio especial basado en DTW, lo que garantiza que el centroide represente fielmente la forma temporal típica del grupo.

2.2.2.2. Soft Dynamic Time Warping (softDTW)

SoftDTW suaviza la minimización de DTW mediante un parámetro de regularización $\gamma > 0$:

$$\text{softDTW}(X, Y) = -\gamma \log \sum_{\pi \in A} e^{-\frac{(A_\pi \Delta(X, Y))_F}{\gamma}}$$

donde:

- A es el conjunto de todas las trayectorias de alineación
- $\Delta(X, Y)$ es la matriz de costos con $\Delta_{ij} = \|x_i - y_j\|^2$
- $\langle \cdot, \cdot \rangle_F$ es el producto de Frobenius. Esta formulación es diferenciable respecto a X e Y , permitiendo su uso en métodos de optimización basados en gradientes. [19]

2.2.2.3. Implementación del Algoritmo TimeSeriesKMeans

El algoritmo sigue un esquema iterativo similar al K-Means, con adaptaciones clave:

Configuración Inicial

El algoritmo TimeSeriesKMeans requiere:

- Entrada de datos: Tensor 3D de forma (n_series , n_puntos , $n_características$)
- Preprocesamiento: Escalado de series con `TimeSeriesScalerMeanVariance()` para normalizar media y varianza [21].

Parámetros Clave

Parámetro	Descripción	Valor por defecto
<code>n_clústeres</code>	Número de clústeres a formar (k)	3

Max_iter	Número máximo de iteraciones	50
tol	Tolerancia; criterio de convergencia	1e-6
N_init	Veces que se ejecuta con distintas semillas	1
metric	Métrica de distancia	euclidean
Max_iter_barycenter	Iteraciones máximas para calcular el centroide	100
init	Método de inicialización de centroides	k-means++
Random_state	Semilla para generador aleatorio	None

Tabla 1: Parámetros Clave TimeSeriesKMeans. Fuente: Propia

Flujo del algoritmo:

1. Inicialización de centroides:

- Selecciona k series aleatorias como centroides iniciales usando k-means++. [22]

2. Asignación a clústeres:

- Calcula la distancia entre cada serie y los centroides usando DTW/softDTW. [22]
- Asigna cada serie al clúster con centroide más cercano.

3. Actualización de centroides:

- Recalcula centroides mediante **DTW Barycenter Averaging (DBA)** [20]:

$$\mu_k = \operatorname{argmin} \sum_{X_i \in C_k} DTW(\mu, X_i)^2$$

4. Convergencia:

- Repite pasos 2-3 hasta que los centroides se estabilicen o se alcance max_iter.

Aplicaciones destacadas

TimeSeriesKMeans se aplica en diversas áreas donde las series temporales presentan *desplazamientos o distorsiones temporales*. Por ejemplo, en finanzas se usa para identificar mecanismos subyacentes en datos de mercado agrupando series según sus patrones de retorno (mejorando la inferencia causal) [12]. En entornos industriales, puede detectar anomalías en procesos a partir de sensores

agrupando comportamientos normales y resaltando desviaciones atípicas [16]. También se emplea para el análisis de series multimodales (varias señales simultáneas), ajustando diferencias de fase; por ejemplo, en hidrología se han alineado perfiles de contaminantes con datos de caudal mediante DTW para agrupar eventos similares [13].

En general, TimeSeriesKMeans mejora la calidad del agrupamiento frente a k-means estándar al capturar la dependencia temporal de las series. Al permitir la deformación temporal (DTW), genera clústeres más coherentes cuando las series están desfasadas o distorsionadas. Sin embargo, su uso implica un mayor costo computacional debido a la complejidad de DTW. A continuación, se resumen ventajas e inconvenientes clave.

Ventajas

- **Manejo de alineamiento temporal:** Al usar DTW/Soft-DTW, puede agrupar series de distinta longitud y compensar desplazamientos de fase. Captura similitudes de forma independientemente del desfase temporal.
- **Resultados más robustos:** Supera al k-means clásico con distancia euclídea en series donde los patrones están desplazados en el tiempo (p.ej. financieros). Mejora la agrupación e interpretabilidad de los resultados en datos temporales.
- **Escalable y convergente:** Hereda las propiedades de k-means: es relativamente simple de implementar y converge (alcanzando un mínimo local) tras unas iteraciones. Al ser iterativo, permite paralelizar ciertas operaciones (p.ej. cálculo de distancias).
- **Flexibilidad:** Puede trabajar con datos multivariantes (cada serie puede tener múltiples características) y opciones de inicialización (k-means++ o aleatoria). Además, Soft-DTW es diferenciable, lo cual facilita su uso en optimización avanzada.

Inconvenientes

- **Costo computacional elevado:** DTW tiene complejidad $O(n^2)$ en la longitud de la serie, por lo que TimeSeriesKMeans es más lento que k-means clásico (que usa $O(n)$). El cálculo iterativo del barycenter DTW es especialmente costoso en series largas.
- **Elección de k e inicialización:** Al igual que k-means, hay que fijar el número de clústeres a priori y los resultados dependen de los centroides iniciales. A menudo se ejecuta varias veces ($n_init > 1$) para mitigar este efecto.
- **Sensibilidad a valores atípicos:** K-means puede verse afectado por outliers (un outlier puede arrastrar el centroide o formar su propio clúster). Se recomienda preprocessar o eliminar atípicos si existen.
- **Limitaciones de la métrica:** DTW no es una métrica de distancia en sentido estricto (no cumple triangular), y en algunos casos deformar excesivamente las series puede no ser deseable. Además, requiere series del mismo tamaño

cuando se usa métrica euclíadiana (para esa métrica alternativa); con DTW esto no aplica.

La técnica TimeSeriesKMeans ha sido aplicada con éxito en el análisis de perfiles de atornillado en líneas de ensamblaje automotriz, donde la identificación de patrones anómalos en curvas par-tiempo es fundamental para garantizar la calidad del proceso. En el estudio de Carlos Betancort [23], se emplearon algoritmos de clustering como TimeSeriesKMeans, implementados a través de la librería tslearn, para agrupar series temporales correspondientes a operaciones de atornillado.

El objetivo era detectar perfiles de apriete desviados respecto a los patrones normales mediante la comparación de formas temporales. Los autores subrayan que este tipo de algoritmos permite diferenciar operaciones correctas de posibles defectos o errores, gracias a su capacidad para capturar la morfología de la señal sin requerir un alineamiento explícito en el tiempo. Este enfoque resulta especialmente útil en entornos de producción donde las señales pueden presentar ligeras variaciones temporales que no necesariamente implican una pérdida de calidad.

2.2.3. KShape

En este contexto, el algoritmo **K-Shape**, propuesto por [24], representa un avance significativo en el clustering de series temporales. K-Shape agrupa las series no solo considerando la proximidad en términos de valores absolutos, sino evaluando la similitud en la forma de las series, lo que permite identificar patrones comunes incluso cuando las series presentan variaciones en escala o están desplazadas en el tiempo. Esta característica lo hace especialmente útil en aplicaciones donde la forma de la señal es más relevante que su valor absoluto, como en el análisis de señales biomédicas, vibraciones mecánicas o patrones de consumo energético.

El núcleo del algoritmo K-Shape es la métrica de distancia denominada **Shape-Based Distance (SBD)**, que mide la similitud entre dos series temporales a través de la correlación cruzada normalizada [8]. Esta métrica es invariante a cambios en la escala y al desplazamiento temporal, lo que permite comparar series que tienen la misma forma, pero pueden estar desfasadas o presentar diferentes amplitudes.

Matemáticamente, para dos series temporales X e Y , la distancia SBD se define como:

$$SBD(X, Y) = 1 - \max_w \frac{CC_w(X, Y)}{\sqrt{R_0(X, X) \cdot R_0(Y, Y)}}$$

donde:

- $CC_w(X, Y)$ es la correlación cruzada entre X e Y con desplazamiento temporal w .
- $R_0(X, X) \cdot R_0(Y, Y)$ son las autocorrelaciones de X e Y en desplazamiento cero.

La maximización sobre w permite encontrar el mejor alineamiento temporal entre las dos series, capturando así la máxima similitud en forma. Esta propiedad es esencial para aplicaciones donde los eventos pueden ocurrir en diferentes momentos, pero mantienen patrones similares.

A diferencia del K-Means tradicional, donde los centroides se calculan como la media aritmética de los puntos asignados a cada clúster, en K-Shape los centroides representan la forma promedio de las series temporales agrupadas. Para ello, se resuelve un problema de optimización que busca maximizar la suma de correlaciones cuadráticas entre el centroide y las series del clúster:

$$\mu_k = \arg \max_{\mu} \sum_{X_i \in C_k} \frac{(X_i \cdot \mu)^2}{\|\mu\|^2}$$

donde:

- μ : candidato a centroide.
- C_k : conjunto de series asignadas al clúster k .
- X_i : serie temporal i -ésima en el clúster k .

Este problema se traduce en obtener el primer vector singular derecho (primer autovector) de la matriz que contiene las series temporales del clúster, utilizando la descomposición en valores singulares (SVD). De esta forma, el centroide refleja la forma dominante del clúster, lo que facilita la interpretación y el análisis posterior.

El algoritmo **K-Shape** es una variante de K-Means especialmente diseñada para trabajar con series temporales. Su funcionamiento se basa en un proceso iterativo en el que se alternan dos etapas principales: asignación de series a grupos y actualización de los centroides, respetando la estructura temporal de los datos. El procedimiento sigue los siguientes pasos:

1. **Inicialización:** Se seleccionan aleatoriamente k series como centroides iniciales. Para mejorar la estabilidad y evitar resultados locales pobres, es aconsejable utilizar técnicas como k -means++.
2. **Asignación:** Cada serie se asigna al clúster cuyo centroide presenta la mayor correlación cruzada normalizada, utilizando como métrica la distancia SBD (*Shape-Based Distance*) [8].
3. **Actualización de centroides:** Se recalcula el centroide de cada grupo aplicando descomposición en valores singulares (SVD), con el fin de obtener una forma promedio representativa del conjunto de series asignadas.

4. **Criterio de parada:** El proceso se repite hasta que las asignaciones no cambian o se alcanza un número máximo de iteraciones.

Este enfoque permite agrupar series temporales respetando tanto la forma como la dinámica de las señales, incluso si existen diferencias en duración, desplazamientos temporales o presencia de ruido.

Este algoritmo trae consigo una serie de ventajas e inconvenientes:

Ventajas:

- **Robustez ante escalas y desplazamientos:** La métrica SBD permite que series similares pero desfasadas o con diferentes amplitudes puedan agruparse correctamente.
- **Eficiencia computacional:** K-Shape es significativamente más rápido que métodos como DTW, lo que lo hace adecuado para conjuntos de datos grandes.
- **Interpretabilidad:** Los centroides representan formas promedio, lo que facilita el análisis y la visualización de los resultados.
- **Resistencia al ruido:** El enfoque centrado en la forma es menos sensible a valores atípicos y fluctuaciones aleatorias.

Limitaciones:

- **Dependencia de la inicialización:** Al igual que otros algoritmos basados en K-Means, los resultados pueden variar en función de la elección inicial de los centroides.
- **Diseño univariado:** K-Shape está enfocado a series de una sola variable. Su uso con series multivariadas requiere adaptaciones o el uso de técnicas complementarias.
- **Falta de modelado temporal profundo:** No capta relaciones complejas ni no lineales como lo hacen los modelos basados en aprendizaje profundo.

El algoritmo K-Shape se ha aplicado con éxito en múltiples dominios gracias a su capacidad para capturar la forma de las series temporales. En ingeniería civil, se ha empleado para clasificar señales de deformación en infraestructuras como presas, identificando zonas de comportamiento similar y posibles riesgos estructurales [25]. En el ámbito del consumo energético, ha permitido analizar perfiles eléctricos en edificios, facilitando una gestión más eficiente y la optimización de tarifas [14]. En procesos industriales, como el control de calidad del atornillado, K-Shape agrupa curvas de par-ángulo y detecta desviaciones que indican errores como sub-aprietas o defectos mecánicos [24]. También se ha utilizado en minería de datos para agrupar series derivadas de redes sociales o análisis de videos, apoyando tareas de análisis de sentimiento y multimedia [8].

Para potenciar su rendimiento, se han desarrollado varias extensiones. Herramientas paralelas y basadas en GPU, como Times-C, han incrementado la velocidad de cálculo hasta 345 veces, habilitando análisis en tiempo real [25]. Métodos como Adaptive Piecewise Aggregate Approximation (APAA) simplifican las series preservando las características clave de la forma [26]. Finalmente, para datos multivariados, se han propuesto modelos como T-GMRF (Time-varying Gaussian Markov Random Fields), que extienden los principios de K-Shape a series con múltiples variables [27].

El algoritmo K-Shape también ha demostrado su utilidad en la industria, particularmente en la monitorización de procesos como la soldadura por arco. Giordano et al. [17] utilizaron K-Shape como método de referencia para agrupar señales de corriente recogidas durante operaciones de soldadura automatizada. Estas series temporales presentan patrones morfológicos específicos que reflejan el estado del proceso y el desgaste del electrodo.

El estudio mostró que K-Shape era capaz de identificar distintos grupos de señales con una elevada coherencia interna, diferenciando entre soldaduras normales y aquellas con variaciones potencialmente críticas. A través del análisis de forma que caracteriza a K-Shape, los investigadores lograron extraer perfiles prototípicos representativos, lo que facilitó el diagnóstico del proceso y la detección de posibles anomalías de manera no invasiva y sin necesidad de etiquetas previas.

El algoritmo K-Shape es una herramienta eficaz y eficiente para el análisis de series temporales, especialmente cuando el objetivo es agrupar señales en función de su forma. Su tolerancia a desplazamientos, escalas y ruido, unida a su bajo coste computacional y facilidad de interpretación, lo convierte en una opción atractiva en entornos industriales y científicos.

Las recientes mejoras en rendimiento y su extensión a datos multivariados amplían aún más sus posibilidades. Como línea futura, sería interesante explorar su combinación con técnicas de aprendizaje profundo para abordar problemas de clustering en línea y el análisis de series complejas.

2.2.4. Índices de valoración del clustering

En el análisis de clustering, después de agrupar los datos, resulta esencial evaluar la calidad de los clústeres formados. Para este fin, se utilizan tres índices comunes: el Silhouette, el Calinski-Harabasz (CH) y el Davies-Bouldin (DB). Cada uno de ellos mide aspectos como la cohesión interna de los clústeres y la separación entre ellos, lo que permite determinar el número óptimo de clústeres o comparar diferentes algoritmos.

El índice Silhouette evalúa cómo se ajusta cada punto a su clúster en comparación con los clústeres cercanos. Calcula una puntuación entre -1 y 1 para cada punto: valores altos indican que el punto está bien colocado y separado de otros clústeres, mientras que valores negativos sugieren que podría pertenecer a otro clúster. Este índice sirve para validar la calidad general del clustering y seleccionar el valor óptimo del número de clústeres. [28]

La fórmula que describe la métrica:

$$Silhouette = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Donde:

- $a(i)$ es la distancia media del punto i a todos los demás puntos dentro de su mismo clúster (mide la cohesión interna).
- $b(i)$ es la mínima distancia media del punto i a todos los puntos del clúster más cercano (mide la separación).

El índice Calinski-Harabasz, conocido también como criterio de razón de varianza, compara la dispersión dentro de los clústeres con la dispersión entre ellos: un valor más alto indica un clustering mejor, con clústeres compactos y bien separados. Se emplea para comparar particiones y encontrar el número ideal de clústeres. [29]

La fórmula que describe la métrica:

$$CH = \frac{BCSS/(k-1)}{WCSS/(n-k)}$$

Donde:

- BCSS es la suma de cuadrados entre clústeres (separación)
- WCSS es la suma de cuadrados dentro de clústeres (compacidad)
- k es el número de clústeres.
- n es el número total de puntos.

El índice Davies-Bouldin mide la similitud entre clústeres a partir de la distancia entre sus centros y el tamaño de los clústeres. Valores bajos señalan un clustering de mayor calidad, con grupos distintos y sin solapamientos. Ayuda a validar la distinción de los grupos y a optimizar parámetros. [30]

La fórmula que describe la métrica:

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} R_{ij}$$

Donde:

- R_{ij} representa la similitud entre los clústeres i y j, definido como la relación entre la dispersión promedio de ambos clústeres y la distancia entre sus centroides.

Estos índices permiten justificar los resultados de manera objetiva, más allá de las visualizaciones.

2.2.5. Criterio de valoración

En el contexto de la evaluación de las configuraciones del clustering, se emplea el operador Weighted Ordered Weighted Averaging (WOWA) para combinar varios de los índices propuestos, asignando pesos a cada uno. Este operador permite obtener una medida única y ponderada que indica si una configuración es adecuada, considerando tanto la importancia relativa de cada índice como su orden en términos de rendimiento.

El operador WOWA es una extensión del **Ordered Weighted Averaging (OWA)**, propuesto por Yager en 1988, que integra dos conjuntos de pesos:

- **Pesos de criterio (p_i)**: reflejan la relevancia intrínseca de cada índice (por ejemplo, mayor peso a la homogeneidad si es crítica).
- **Pesos de orden (w_i)**: ponderan la posición de los valores una vez ordenados (por ejemplo, priorizar los mejores o peores desempeños).

La combinación de ambos permite modelar comportamientos complejos, desde el optimismo (enfocarse en los mejores valores) hasta el pesimismo (atender los peores), o un equilibrio intermedio. En clustering, WOWA genera una puntuación global que no solo promedia, sino que ajusta estratégicamente según el objetivo: maximizar separación, compactación o detección de anomalías.

Matemáticamente, para un conjunto de índices x_1, x_2, \dots, x_m ordenados de mayor a menor, WOWA se define como:

$$WOWA(x_1, \dots, x_m) = \sum_{i=1}^m v_i \cdot x_i$$

donde los pesos agregados v_i se construyen a partir de p y w mediante una función de cuantificación. Esto asegura que el resultado sea monótono y compensatorio controlado, evitando que un índice dominante oculte fallos en otros.

En la práctica, WOWA sirve para agregar valores de múltiples indicadores de manera equilibrada, priorizando aquellos con mayor relevancia o ajustando según el contexto, lo que facilita la comparación de diferentes configuraciones y la selección

de la más óptima. En aplicaciones de clustering, ayuda a resolver problemas de multicriterio al proporcionar una puntuación global que refleja la calidad general sin ignorar las diferencias entre los índices. Por ejemplo, si se desea una configuración robusta ante ruido, se puede asignar mayor peso al índice que mida homogeneidad interna; si el objetivo es detección de anomalías, se prioriza la separación entre clústeres.

La principal ventaja de WOWA frente a promedios ponderados simples radica en su capacidad para modelar criterios de decisión más complejos. Permite, por ejemplo, priorizar configuraciones donde al menos dos índices superen un umbral mínimo o garantizar que el índice de peor rendimiento no sea excesivamente bajo, integrando tanto la relevancia de cada métrica como su posición relativa tras ordenar los resultados. Esta propiedad resulta especialmente útil en entornos industriales, donde la calidad del clustering debe cumplir requisitos operativos específicos, como alta tolerancia a ruido, sensibilidad a defectos críticos o equilibrio entre compactación y separación de clústeres. [31]

3. PROCESO PARA LA DETECCIÓN DE ANOMALÍAS

El objetivo principal de esta propuesta es desarrollar una herramienta informática que ayude a detectar de manera eficiente anomalías en los procesos de atornillado de motores industriales, analizando de forma inteligente las curvas de par y ángulo que se generan en cada operación de ensamblaje. Este reto, considerado en el sector automotriz como un aspecto clave de calidad y seguridad, requiere un enfoque estructurado que combine técnicas avanzadas de análisis de datos con la experiencia práctica en procesos productivos.

La propuesta se fundamenta en la aplicación de algoritmos de clustering especializados en series temporales, para identificar patrones anómalos en el comportamiento de las uniones atornilladas. El enfoque metodológico se estructura en cuatro tareas principales que cubren desde la adquisición de datos hasta la evaluación de resultados (**Figura 9**):

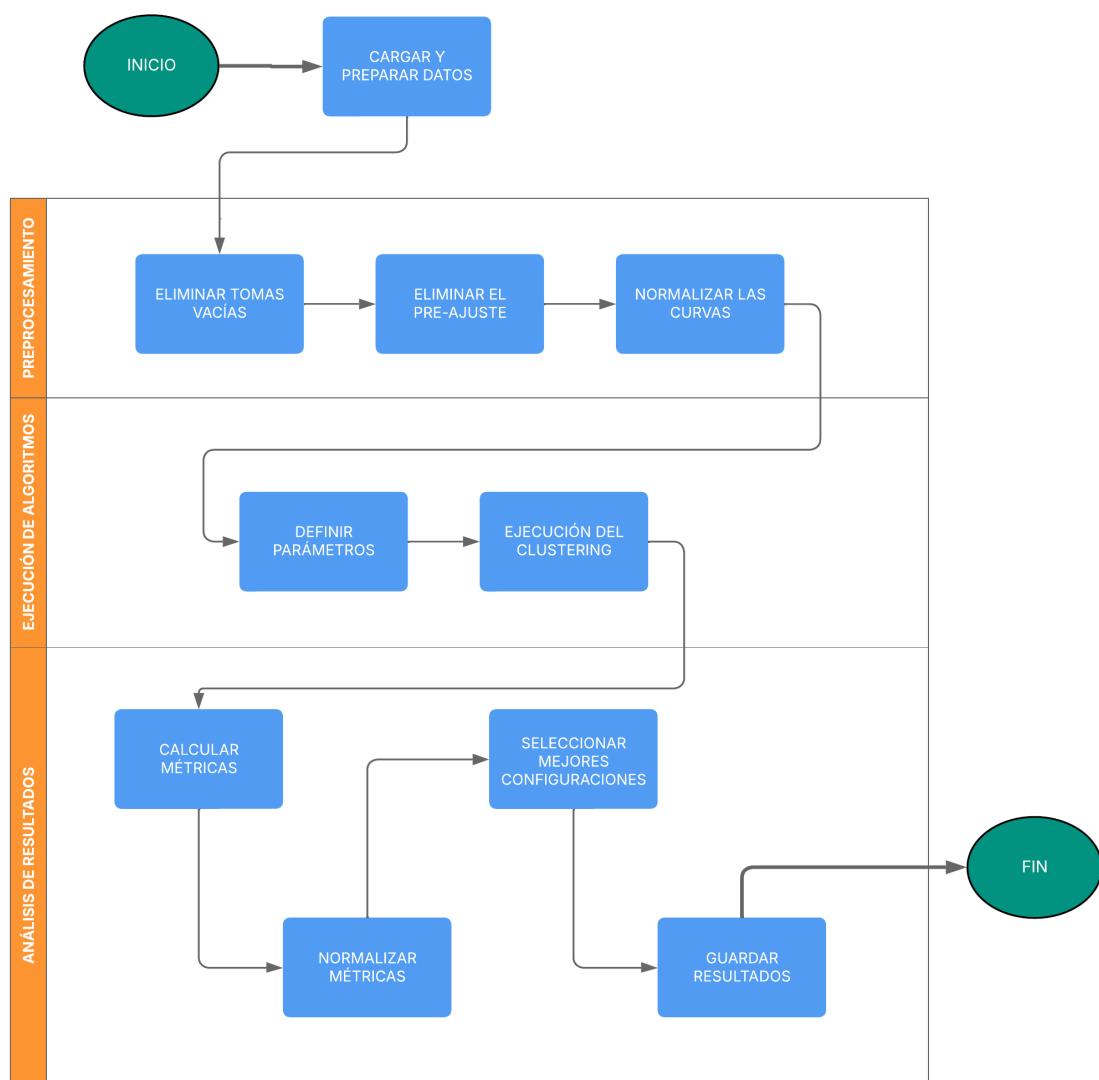


Figura 9: Diagrama de flujo. Fuente: Propia (LucidChart)

1. Captura de datos (datos de series temporales).
2. Preprocesamiento: Procesamiento de imágenes (acondicionamiento y normalización de series temporales).
3. Ejecución de los algoritmos de clustering y criterios de validación
4. Análisis y criterios de evaluación de los resultados.

La razón de adoptar este enfoque está en superar las limitaciones de los métodos tradicionales de inspección manual y de control basados en umbrales fijos, que resultan poco efectivos ante la variabilidad natural de los procesos industriales modernos y frente al gran volumen de datos que genera la digitalización de las líneas de producción.

3.1. Adquisición de Datos de Series Temporales

La primera fase se centra en la recopilación sistemática y estructurada de los datos de entrada, que constituyen la base fundamental para el análisis posterior. Los datos corresponden a series temporales que registran variables dinámicas clave durante el proceso industrial, como la evolución de magnitudes físicas en función del tiempo. Por ejemplo, en un proceso de ensamblaje, estas series pueden capturar el comportamiento de variables críticas, como se observa en la **Figura 10** (curva que muestra una aceleración inicial seguida de una estabilización) y la **Figura 11** (curva que refleja un incremento progresivo, estabilización y un valor máximo alcanzado). Estos datos son capturados mediante sensores especializados integrados en los sistemas de producción, asegurando alta precisión y resolución temporal.

Los datos se organizan en conjuntos que representan casos normales y anómalos, permitiendo una representación completa de las condiciones del proceso. Esta etapa garantiza que los datos sean representativos, capturando tanto comportamientos típicos como posibles anomalías.



Figura 10: Diagrama del proceso de atornillado. Ángulo-tiempo. Fuente: Propia (Python)

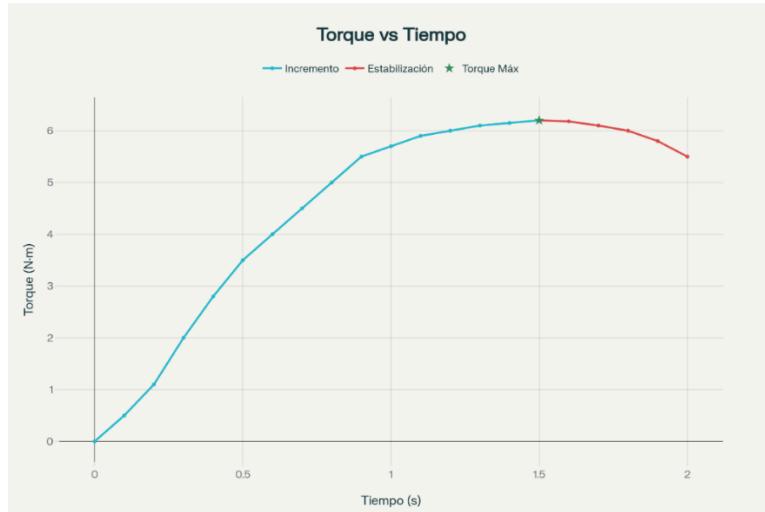


Figura 11: Diagrama del proceso de atornillado. Par-tiempo. Fuente: Propia (Python)

3.2. Preprocesamiento

En esta segunda etapa se realiza el preprocesamiento de los datos, que constituye una fase crítica en el análisis de procesos industriales, particularmente en operaciones de atornillado donde la calidad y precisión son fundamentales [32]. Esta etapa preliminar transforma los datos brutos capturados por sensores en información estructurada y depurada, eliminando ruido, inconsistencias y valores atípicos que podrían comprometer la validez de los análisis posteriores [25]. En el contexto específico de procesos de atornillado, donde se registran variables como par, ángulo y tiempo, el preprocesamiento adquiere especial relevancia para identificar con precisión las fases operativas relevantes y descartar información no significativa [10].

Los objetivos fundamentales del preprocesamiento incluyen:

1. Mejorar la calidad y consistencia de los datos mediante la eliminación de información redundante o errónea.
2. Reducir la dimensionalidad y complejidad de los conjuntos de datos para optimizar el rendimiento computacional [25].
3. Extraer características relevantes que faciliten la identificación de patrones y anomalías.
4. Segmentar adecuadamente las series temporales para aislar fases operativas específicas [25].

Por estas razones se aplica al inicio un procesamiento de las series temporales para eliminar la parte del preajuste y quedarse únicamente con la parte de la gráfica que

corresponda con la información del proceso. Primero se parte de los datos en bruto y se eliminan los datos que no interesan para quedarse solo con el tiempo, par y ángulo, que son las características que queremos analizar.

La propuesta incluye mecanismos robustos de trazabilidad. Cada registro viene con el número de identificación del elemento para poder hacer un seguimiento de este mismo e identificarlo en caso de anomalías. Cada curva de par-ángulo tiene su propio "nombre único" (un identificador), que nos permite seguirla hasta el elemento exacto, dejando un rastro claro que permite analizar mejor las causas.

El resultado de esta fase es un conjunto homogéneo de series temporales normalizadas, listas para el análisis mediante técnicas de clustering, manteniendo la información esencial del proceso.

3.2.1. Limpieza de datos

Cada etapa del procesamiento integra un control automático de calidad orientado a la limpieza de datos. De esta forma, se eliminan las tomas vacías o aquellas con menos de 10 puntos, previniendo la inclusión de registros falsos o erróneos que no contribuyan al experimento.

Esta medida asegura la solidez del conjunto de datos, minimizando errores y garantizando su idoneidad para el análisis de clustering.

3.2.2. Acondicionamiento de Series Temporales

Esta etapa constituye el núcleo esencial del tratamiento de los datos, un paso fundamental que determina la calidad de los resultados en las fases posteriores de análisis. En esta etapa se procesan las series temporales de variables dinámicas, aplicando técnicas específicas para preservar las características relevantes del proceso mientras se eliminan variaciones no deseadas.

3.2.2.1. Segmentación y Extracción de Regiones de Interés

Antes de proceder con cualquier transformación, se realiza un análisis exploratorio inicial del comportamiento temporal. Las curvas presentan duraciones variables, influenciadas por configuraciones de equipos y características mecánicas, por lo que se segmentan y estandarizan temporalmente, preservando la dinámica del proceso.

Una de las aportaciones clave es un algoritmo de segmentación que identifica automáticamente las regiones relevantes dentro de cada serie temporal:

- **Detección del punto de inflexión:** Se utiliza el valor máximo de las variables clave y su punto medio como referencia temporal para la segmentación.
- **Ventana adaptativa:** Se define un intervalo centrado en el punto medio de los valores máximos para optimizar el enfoque en la región de interés.
- **Eliminación del pre-ajuste:** La creación de la ventana con la región de interés descarta la fase inicial de la curva, donde los valores son bajos e inestables, enfocando el análisis en la región más representativa.

3.2.2.2. Normalización y Estandarización

El proceso de normalización se centra en la variación en amplitud: Los valores varían según condiciones operativas, por lo que se aplica un escalado para normalizarlos a un rango estándar, lo que permite una comparación objetiva. Adicionalmente, este ajuste contribuye a la protección de datos sensibles que no pueden divulgarse.

El propósito es equilibrar las series temporales para evitar que variaciones en escala afecten desproporcionadamente el agrupamiento, mejorando así la precisión y la equidad de los resultados.

3.3. Ejecución de algoritmos

La tercera etapa representa el núcleo analítico de la propuesta y constituye la aplicación de técnicas avanzadas de clustering específicamente adaptadas para series temporales. Esta fase es crucial para descubrir patrones latentes en los datos y establecer una clasificación automática que permita distinguir entre comportamientos normales y anómalos en procesos industriales con variables dinámicas.

3.3.1. Selección y Configuración de Algoritmos

El enfoque implementa técnicas de clustering especializadas en series temporales, cada una adaptada para capturar aspectos únicos del análisis:

- **Técnicas basadas en alineamiento dinámico:** Utilizan medidas de distancia que ajustan de manera flexible el tiempo entre secuencias que difieren en ritmo o fase, conservando la estructura principal de las curvas. Esto es

particularmente útil para reconocer patrones que siguen órdenes similares, pero con variaciones en la duración debido a diferencias en los procedimientos o en los elementos involucrados. TimeSeriesKMeans con DTW es un ejemplo de esta técnica. ([Sección 2.2.2.1](#))

- **Técnicas con suavizado:** Son mejoras que incorporan factores de ajuste para hacer la comparación más resistente al ruido, lo cual es clave en contextos industriales donde las variaciones pueden provenir de condiciones externas o de medición, en vez de indicar problemas reales. TimeSeriesKMeans con SoftDTW es un ejemplo de esta técnica. ([Sección 2.2.2.2](#))
- **Técnicas basadas en correlación:** Creadas especialmente para datos secuenciales en el tiempo, aplican indicadores de semejanza que evalúan la configuración general de las curvas sin necesidad de sincronizarlas paso a paso. Resultan ideales para captar irregularidades que aparecen como alteraciones en la silueta de las series, sin importar su escala o intensidad. El algoritmo KShape es un ejemplo de esta técnica. ([Sección 2.2.3](#))

3.3.2. Optimización de Parámetros

El enfoque explora de forma sistemática diferentes configuraciones para optimizar las técnicas de clustering:

- **Número de grupos:** Se prueba un rango equilibrado de grupos, entre una granularidad mínima que distinga los principales tipos de comportamiento y evitando una fragmentación excesiva que complique la interpretación.
- **Iteraciones máximas:** Se evalúan varios niveles de iteraciones, buscando un compromiso entre la convergencia del algoritmo y el rendimiento computacional.
- **Método de inicialización:** Se utilizan métodos para seleccionar los centroides iniciales, distribuyéndolos inteligentemente en el espacio de datos para mejorar la convergencia y reducir la probabilidad de óptimos locales.

3.4. Análisis de resultados

Esta cuarta y última etapa establece un marco de evaluación que asegura la aplicabilidad práctica y la fiabilidad de los resultados obtenidos. Va más allá de la verificación técnica para enfocarse en aspectos clave como la usabilidad y el impacto real en el entorno industrial.

El análisis de resultados integra las salidas de las fases previas en un proceso de interpretación global, que no solo verifica la calidad técnica del clustering, sino que también examina su capacidad para generar conocimientos accionables en contextos reales de operación. En el ámbito de procesos industriales con series temporales (donde los datos capturan dinámicas complejas, tales como variaciones en par o ángulo), este marco enfatiza la habilidad de los clústeres para identificar patrones anómalos de forma interpretable, reduciendo la complejidad y maximizando su relevancia práctica.

Se fundamenta en un enfoque multicriterio que combina métricas cuantitativas, como varios índices de calidad del clustering, que se agregan mediante un operador, para obtener una puntuación única y objetiva, con evaluaciones cualitativas centradas en elementos visuales como gráficas de clústeres, representaciones de series temporales agrupadas. Estos elementos permiten una inspección intuitiva de la coherencia y el significado de los grupos formados. De esta manera, se selecciona la configuración óptima de forma equilibrada y fundamentada.

3.4.1. Evaluación Cuantitativa

El análisis cuantitativo ofrece una evaluación objetiva del rendimiento del clustering, a través de indicadores que miden la calidad esencial de los grupos obtenidos. Estas métricas analizan elementos como la cohesión interna de los clústeres, la separación entre ellos y la dispersión general de los datos, con el fin de asegurar que los grupos sean compactos y claramente diferenciados. En el contexto de series temporales industriales, esta revisión es fundamental para confirmar que los patrones detectados no surjan simplemente del ruido, sino que representen comportamientos genuinos, como las variaciones en la evolución de variables relevantes a lo largo del tiempo.

La evaluación de los resultados de clustering se basa en un sistema de validación multi-criterio, enfoque que combina métricas internas y externas:

- Métricas internas:
 - **Cohesión y separación:** Miden la compactación dentro de cada grupo y su distancia respecto a otros, con valores que indican mejor agrupamiento cuando son más altos o bajos según la métrica.
Ejemplos de este tipo de métrica son Silhouette y Davies-Bouldin ([Sección 2.2.4](#)).
 - **Dispersión:** Analizan la relación entre la varianza entre grupos y dentro de ellos, favoreciendo configuraciones con grupos bien definidos.

Un ejemplo de este tipo de métrica es Calinski-Harabasz (**Sección 2.2.4**).

- **Número de clústeres:** Se prioriza un valor más pequeño para promover una segmentación más interpretable y generalizable, evitando la sobrefragmentación de los datos.

Este criterio se mide mediante el K_Index, un indicador normalizado que penaliza el aumento excesivo de clústeres y que se detalla en la sección subsiguiente.

- Métricas externas:

- **Homogeneidad:** Es un índice que mide qué tan limpios están los grupos en base a las etiquetas ya conocidas, premiando configuraciones donde cada clúster tiene mayoritariamente un tipo de dato similar. Se explica con más detalle más adelante.

3.4.1.1. Índices propuestos de Evaluación

En el marco de esta propuesta, se incorporan dos métricas propuestas para enriquecer la validación de los clústeres en el análisis de series temporales. Estos índices complementan las métricas internas, priorizando aspectos prácticos como la pureza de los grupos y la simplicidad interpretativa, esenciales para entornos donde la distinción entre comportamientos normales y anómalos debe ser clara y accionable. A continuación, se describen su formulación y relevancia.

3.4.1.1.1. Homogeneidad: Métrica de Pureza de Clústeres

La Homogeneidad es un índice desarrollado ad hoc para medir la capacidad del clustering en separar muestras de clases conocidas (normales o anómalas), ponderando la contribución de cada clúster según su tamaño relativo al conjunto total. Su diseño resalta la utilidad práctica en escenarios industriales, donde clústeres "puros" (dominados por una clase) facilitan la identificación rápida de desviaciones, reduciendo falsos positivos y mejorando la toma de decisiones.

La métrica se define como:

$$\text{Homogeneidad} = \sum_{i=1}^k \left(\frac{s_i}{N} * \max \left(\frac{o_i}{s_i}, \frac{n_i}{s_i} \right) \right)$$

donde:

- k : número total de clústeres,
- s_i tamaño del clúster i (número de muestras),
- N : número total de muestras en el conjunto,
- o_i : número de muestras normales (OK) en el clúster i ,
- n_i : número de muestras anómalas (NOK) en el clúster i .

Esta métrica genera un valor entre 0 y 1, donde 1 indica clústeres perfectamente puros (cada uno 100% de una clase) y 0 representa mezclas totales. El término $\frac{s_i}{N}$ actúa como peso, dando mayor relevancia a clústeres grandes, mientras que $\max\left(\frac{o_i}{s_i}, \frac{n_i}{s_i}\right)$ captura la dominancia de la clase mayoritaria en cada clúster. Su integración con los demás índices permite equilibrar la pureza con otras métricas, haciendo que la propuesta sea particularmente efectiva para procesos donde la segregación de anomalías es prioritaria, como en la monitorización de uniones mecánicas.

3.4.1.1.2. K_Index: Métrica de Simplicidad de Configuración

El K_Index es un índice complementario que penaliza configuraciones con un número excesivo de clústeres, promoviendo soluciones parsimoniosas y fáciles de interpretar en contextos operativos. En procesos industriales, donde la complejidad puede complicar la aplicación práctica, este índice favorece valores bajos de k (cerca del mínimo razonable, e.g., 5) para priorizar interpretabilidad sobre granularidad excesiva.

La fórmula se define como:

$$K_{Index} = \begin{cases} 1 - \frac{k - k_{min}}{k_{max} - k_{min}} & \text{si } k \leq k_{max} \\ 0 & \text{si } k > k_{max} \end{cases}$$

donde:

- k : número de clústeres en la configuración,
- k_{min} : mínimo número de clústeres considerado (5, para capturar patrones básicos),
- k_{max} : máximo número evaluado (24, para evitar fragmentación).

Esta normalización lineal asigna 1 al mínimo ($k = k_{min}$), decrece gradualmente hasta 0 en k_{max} y penaliza valores superiores con 0. Por ejemplo, con $k_{min} = 5$ y

$k_{max} = 24$, para $k = 5$: K_Index = 1; para $k = 10$, K_Index = 0,74; y para $k = 24$, K_Index = 0. (Figura 12).

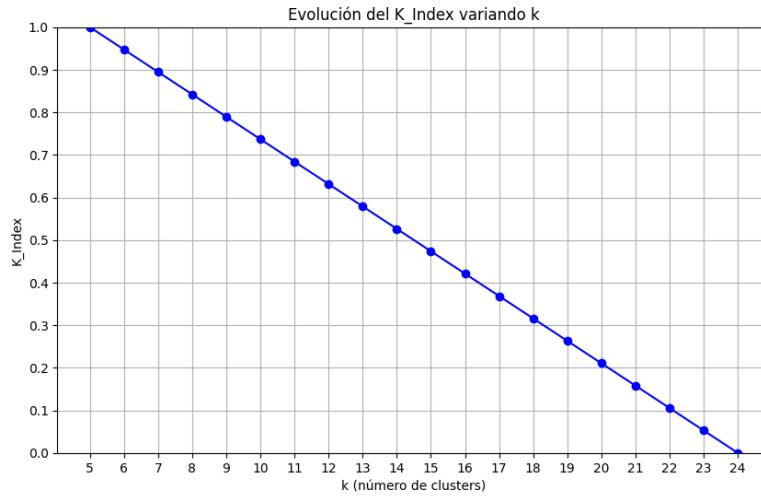


Figura 12: Evolución de K_Index en función de k. Fuente: Propia (Python)

3.4.1.2. Operador WOWA (Weighted Ordered Weighted Averaging) y Ranking

Para unir todas las métricas en una sola decisión clara, el método usa un operador de agregación que prueba varias combinaciones de pesos, para ver cómo priorizar de distintas maneras. Estos pesos se adaptan según lo que se quiera resaltar en cada caso, con el fin de mejorar la detección de anomalías. Además, se hace un ranking de los resultados basados en esos puntajes, ordenando las configuraciones de mejor a peor, tanto para cada técnica por separado como en el total. Esto simplifica la elección de las opciones más útiles para el análisis. (Sección 2.2.5)

3.4.1.3. Tablas de composición de clústeres

Para complementar las métricas numéricas, se incluyen tablas que permiten una revisión detallada de la composición de los clústeres. Estas herramientas apoyan la evaluación objetiva al resumir distribuciones y patrones de forma clara y medible.

Estas tablas detallan, para cada clúster, el tamaño total, el conteo y porcentaje de elementos correctos y defectuosos, junto con la distribución de conteos y porcentajes por atributos relevantes. Esto ofrece una vista tabular precisa para identificar clústeres puros (donde un alto porcentaje de un tipo domina) o mixtos (con distribuciones equilibradas entre tipos). Se aplican formatos condicionales, como rojo para clústeres con separaciones mixtas o insuficientes (por ejemplo, porcentajes equilibrados) y verde para grupos puros (porcentajes dominantes de un tipo), lo que

resalta cuantitativamente la homogeneidad y permite una evaluación objetiva de la pureza y efectividad de cada clúster.

3.4.2. Análisis Cualitativo y Visualización

Complementando la cuantificación, el análisis cualitativo incorpora herramientas visuales para facilitar la interpretación humana de los clústeres. Estas representaciones permiten explorar la estructura subyacente de los datos, identificando solapamientos o características distintivas que no emergen de números aislados. En entornos industriales, donde la toma de decisiones depende de la comprensión intuitiva de anomalías, la visualización actúa como puente entre los resultados algorítmicos y la acción operativa.

Para potenciar esta interpretación, la propuesta integra representaciones esquemáticas y gráficas adaptadas a la complejidad de las series temporales:

- **Imágenes de superposición de series en clústeres:** Se generan gráficas que superponen las curvas de las series temporales asignadas a cada clúster, utilizando colores distintivos para diferenciar clases. Estas imágenes revelan patrones visuales, como alineamientos en fases elásticas o desviaciones en picos anómalos, permitiendo una inspección intuitiva de la homogeneidad y separabilidad, esencial para validar la relevancia práctica de los clústeres en procesos dinámicos.

De esta forma, el análisis cualitativo enriquece la comprensión general, asegurando que los hallazgos sean no solo precisos, sino también accesibles para su aplicación en contextos reales.

4. EXPERIMENTACIÓN

El Capítulo 4 representa la validación práctica de la propuesta metodológica delineada en el Capítulo 3, trasladando los conceptos abstractos a un contexto real mediante la aplicación de técnicas de clustering a datos industriales específicos.

El capítulo se estructura en tres subsecciones principales. En primer lugar, el caso de uso describe el problema aplicado a un entorno industrial real, caracterizando el conjunto de datos y sus desafíos inherentes. A continuación, la metodología experimental expone los detalles operativos de la implementación, incluyendo el preprocesamiento adaptado, las configuraciones de algoritmos y el proceso de evaluación. Finalmente, los resultados obtenidos presentan los hallazgos cuantitativos y cualitativos, respaldados por tablas, gráficos y visualizaciones de clústeres.

4.1. Caso de uso

El caso de uso de este trabajo se centra en el análisis de anomalías en el proceso de atornillado de motores, un componente crítico en la industria automotriz donde la precisión del par y el ángulo influye directamente en la calidad y seguridad del ensamblaje. Los datos disponibles provienen de mediciones de sensores durante el atornillado, normalizados entre 0 y 1 para las variables Par y Ángulo, que reflejan la evolución temporal del proceso de manera estandarizada. Estos datos se dividen en dos categorías: aprietas en estado OK (funcionales) y NOK (con anomalías), con un total de 200 conjuntos de aprietas procesados, de los cuales 150 están en estado OK y 50 en estado NOK, seleccionados para garantizar una representatividad adecuada y un enfoque especial en la detección de anomalías.

Para caracterizar mejor el conjunto de datos, se analizan estadísticas descriptivas básicas de las series temporales, como el número de tomas por categoría, la duración en segundos (mínima, máxima y media) y el número de puntos por toma (mínimo, máximo y medio). Estas métricas ayudan a entender la variabilidad inherente en las mediciones, que oscila entre duraciones de 2,6 y 3,4 segundos y un número de puntos que varía según la resolución de los sensores. La **Tabla 2** resume estos valores, destacando diferencias sutiles entre categorías que justifican el uso de técnicas robustas de clustering.

Categoría	Número de tornillos	Duración mínima (s)	Duración máxima (s)	Duración media (s)	Nº puntos mínimo	Nº puntos máximo	Nº puntos medio
OK	800	2.60	3.39	2.96	348	453	395.9
NOK	400	2.61	3.10	2.96	349	415	395.7
Total	1200	2.60	3.39	2.96	348	453	395.8

Tabla 2: Estadísticas descriptivas de las series temporales. Fuente: Propia (Python)

La caracterización de las series temporales revela un conjunto rico pero desafiante, con duraciones originales entre 2,6 y 3,4 segundos uniformadas a una escala estándar para facilitar el análisis. Cada operación de apriete, se agrupa en un conjunto de 8 aprietes síncronos (8 tomas o tornillos), lo que genera un volumen significativo de datos que presentan ruido, valores atípicos y desfases temporales, aspectos que se corrigen mediante ajustes previos. Estas curvas estandarizadas muestran picos distintivos que indican fases de preajuste y apriete real, ofreciendo pistas clave para identificar patrones anómalos. Estos picos se observan en las gráficas originales de la **Figura 13**. El objetivo principal es emplear técnicas de clustering sobre estas series procesadas para detectar anomalías como sub-aprietes o cruces de roscas, contribuyendo a mejorar la calidad del control de procesos en la producción automotriz mediante un análisis robusto y reproducible.

4.2. Metodología experimental

La metodología diseñada para identificar las mejores configuraciones de agrupamiento de series temporales derivadas de las mediciones de par y ángulo en motores se basa en un enfoque meticuloso y estructurado, pensado para extraer patrones significativos de forma sistemática. El desarrollo del programa para analizar las curvas par-ángulo y detectar anomalías en el atornillado, se identificó que los parámetros configurables eran fundamentales para obtener resultados fiables. Estos valores no solo determinan cómo se procesan los datos y se aplican los algoritmos, sino que también reflejan las decisiones que se toman para adaptar el análisis a las características de las series temporales.

A continuación, se describen las etapas principales del análisis: configuración general, preprocessamiento y estandarización, configuración de algoritmos de clustering, y evaluación con la métrica WOWA, incluyendo los parámetros utilizados

en cada una, su propósito y los ajustes realizados mediante pruebas iterativas. Al final de la sección de algoritmos de clustering se incluye una tabla que los resume, proporcionando una referencia clara para la metodología propuesta.

El proceso arranca con un paso inicial de preparación de los datos, donde se eliminan las secciones que no aportan valor al análisis y se ajustan las duraciones de las series para que todas tengan una escala uniforme, facilitando comparaciones precisas. Este ajuste inicial es clave para homogenizar las mediciones y reducir el impacto de variaciones no deseadas.

A continuación, se aplican diferentes configuraciones de técnicas de agrupamiento, explorando un rango de grupos posibles y ajustando el número de iteraciones para refinar los resultados. Esta etapa busca capturar la diversidad de comportamientos en las series temporales, permitiendo distinguir entre patrones normales y anómalos. Para evaluar la calidad de cada configuración, se emplea un conjunto de indicadores que miden cómo de bien se agrupan los datos dentro de cada clúster, cuánto se separan entre sí los distintos grupos y hasta qué punto se respetan las categorías predefinidas de motores funcionales y con fallos. Estos indicadores se combinan en un criterio ponderado que da más peso a la capacidad de detectar anomalías, ordenando las configuraciones según su rendimiento.

Para complementar el análisis, se crean representaciones visuales que muestran cómo se distribuyen las series temporales dentro de los grupos identificados, ayudando a interpretar los resultados de manera intuitiva. Además, se elaboran reportes detallados que recogen las métricas obtenidas, la distribución de los estados de los motores y las configuraciones más efectivas. Este enfoque asegura que el proceso sea reproducible y que los resultados puedan integrarse fácilmente en estrategias de control de calidad, ofreciendo una base sólida para tomar decisiones en la industria automotriz.

4.2.1. Preprocesamiento

En la fase de preprocesamiento, se probaron varias opciones de intervalos para igualar la duración de las curvas a una longitud fija. Estas pruebas ayudaron a comparar distintos grados de detalle, mostrando que pocos intervalos generaban un suavizado excesivo que borraba aspectos clave, mientras que más intervalos mantenían mejor los picos y cambios notables. Así, se adaptó el nivel de precisión temporal según las características propias de cada curva, mejorando la forma en que se representan los datos para los pasos siguientes del análisis. El algoritmo del preprocesamiento se detalla en la **Tabla 3**:

PARA cada muestra

SELECCIONAR columnas {Tiempo, Par, Ángulo}

SI num_puntos de la toma ≥ 10 ENTONCES

 Eliminar intervalo irrelevante (pre-ajuste) usando punto medio de par/ángulo

 Extraer segmento de interés y normalizar curva a [0,1]

 Generar gráficas y guardar resultados con identificador único

FIN SI

FIN PARA

Tabla 3: Pseudocódigo asociado al preprocessamiento de datos. Fuente: Propia

Para realizar el preprocessamiento, se calcula el punto donde la gráfica toma la mitad del valor máximo y a partir de ese tiempo se aplica un intervalo a cada lado dependiendo de si es el par o el ángulo (**Figura 13**). Para el par, se toma un intervalo de 0.85 segundos antes y 1.6 segundos después de la mitad del valor máximo, mientras que para el ángulo se usa un intervalo de 1.1 segundos antes y 1.1 segundos después. En la **Figura 13**, las dos gráficas de la izquierda muestran el par y ángulo originales, mientras que las dos de la derecha presentan el par y ángulo ya preprocessados.

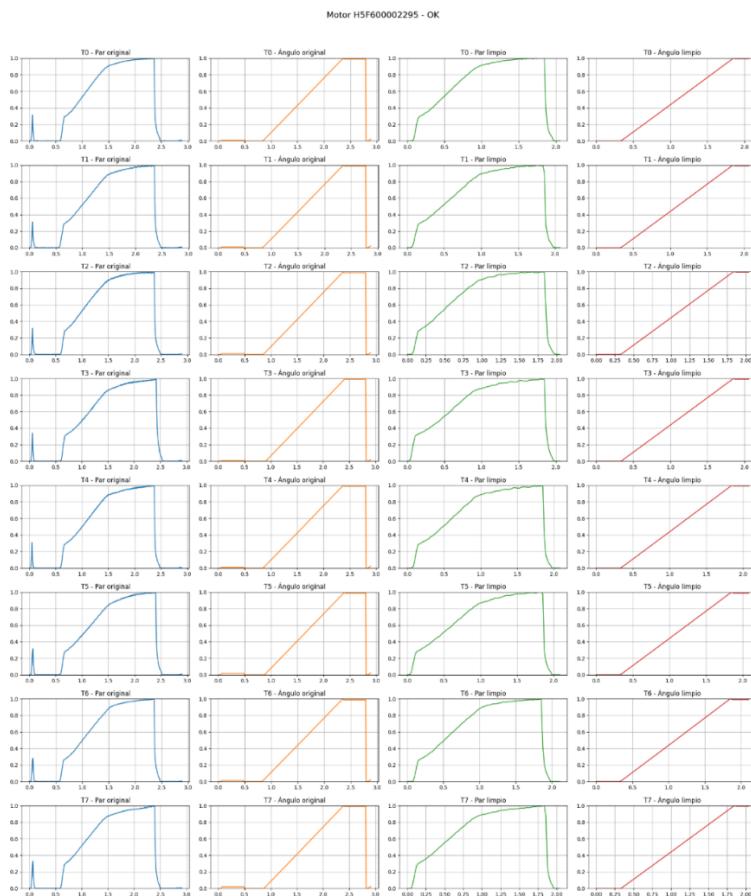


Figura 13: Ejemplo de procesamiento de datos. Fuente: Propia (Python)

Así nos aseguramos de que los datos miden lo mismo en tiempo para poder analizarlos conjuntamente y de que nos quedamos con la información que nos interesa, ya que el preajuste no es válido para nuestro análisis y puede generar anomalías o errores de interpretación a la hora de realizar el clustering.

Además, se genera un archivo .json con los datos preprocesados de cada toma, incluyendo las columnas Tiempo (Seg), Par (N.m) y Ángulo (Deg), lo que permite almacenar y reutilizar la información estructurada para análisis posteriores (**Tabla 3**).

También se definen $n_{ok} = 150$ y $n_{nok} = 50$, que limitan la cantidad de motores en estado "OK" y "NOK" a procesar. Estas cifras se eligieron basándose en la disponibilidad de datos y en un equilibrio entre representatividad: 150 conjuntos de aprietas OK y 50 conjuntos de aprietas NOK para detectar anomalías. Esto asegura que el análisis refleje tanto lo típico como lo atípico sin sobrecargar el procesamiento. Además, en ocasiones variaba esta cantidad de datos de entrada para poder realizar comprobaciones más rápidas cuando realizaba cambios significativos en el programa.

4.2.2. Algoritmos de clustering

Para agrupar las series temporales en función de su forma dinámica se emplean algoritmos de clustering especializados. En este trabajo se utilizan principalmente variantes de **k-means** adaptadas a series temporales: la clase TimeSeriesKMeans con métrica *DTW* o *Soft-DTW*, y el algoritmo **K-Shape**. Las características clave de estos métodos son:

- **TimeSeriesKMeans con DTW (Dynamic Time Warping):** La métrica DTW mide la similitud entre dos series temporales alineando sus puntos de forma no lineal. Esto permite comparar series que pueden tener desfases o variaciones en el tiempo. Al usar `metric="dtw"` en TimeSeriesKMeans, el algoritmo calcula centroides basados en el promedio de DTW. En la práctica, esta variante agrupa series con formas similares independientemente de pequeños desplazamientos temporales.
- **TimeSeriesKMeans con Soft-DTW:** El método Soft-DTW es una versión suavizada y diferenciable de DTW. Al emplear `metric="softdtw"`, TimeSeriesKMeans calcula centroides mediante Soft-DTW, lo que facilita el cómputo de promedios en el espacio DTW. Soft-DTW ha demostrado mejorar la calidad de clustering y de cálculo de centroides frente a DTW. En la práctica se parametriza con un valor de suavizado y para controlar la regularización.
- **K-Shape:** K-Shape es un algoritmo de clustering diseñado para series temporales que utiliza una medida basada en correlación normalizada para comparar formas. A diferencia de DTW, K-Shape no realiza un alineamiento

dinámico, sino que compara las series directamente según su “forma” general. K-Shape es iterativo y construye centroides de forma eficiente a partir de las propiedades de esta distancia de correlación. Se ha observado que K-Shape produce clústeres bien separados y homogéneos para diversos conjuntos de series.

En la tabla 4 se expone el pseudocódigo de la aplicación de TimeSeriesKMeans y KShape:

```
PARA algoritmo EN {TimeSeriesKMeans, KShape}:
    PARA num_clúster EN {5, 25}:
        PARA iteraciones EN {50, 100, 150}:
            PARA método EN {DTW, SoftDTW}:
                Ejecutar clustering para: algoritmo, num_clúster, iteraciones, método
                Calcular índices de evaluación: Silhouette, DB, CH, Homogeneidad, K-Index
                Analizar composición de los clústeres
                Generar representaciones visuales de los clústeres
                Generar archivos de reporte con resultados por configuración
            FIN_PARA
        FIN_PARA
    FIN_PARA
FIN_PARA
```

Tabla 4: Pseudocódigo de evaluación de curvas par-ángulo. Fuente: Propia

En el programa, cada algoritmo se ejecuta para varias configuraciones de k (número de clústeres), en este caso varían de $k = 5$ hasta $k = 25$. Además, existen otros parámetros que son actualizables, como las iteraciones máximas (max_iters), que varían en 50, 100 y 150 y como el intervalo para estandarización (n_init) que varía en 5 y 10. Estas ejecuciones permiten estudiar cómo varían los resultados con distintos niveles de granularidad. Se inicializa el algoritmo con el método *k-means++* (que mejora la selección de centroides iniciales). Esta serie de parámetros van variando, creando las diferentes configuraciones mediante bucles for. En total para este procedimiento con esos parámetros se generan 180 configuraciones, de las cuales las 60 primeras pertenecen a TimeSeriesKMeans con DTW, las siguientes 60 pertenecen a TimeSeriesKMeans con Softdtw y las últimas 60 pertenecen al algoritmo KShape.

El programa cada vez que realiza una configuración genera un archivo de valores o tabla con las características de dicha configuración: Número de configuración,

número de clústeres, iteraciones máximas y el clúster que es junto con sus datos de número de tornillos, número de tornillos OK, porcentaje de tornillos OK, número de tornillos NOK, porcentaje de tornillos NOK y el número de tornillos de cada atornilladora con sus respectivos porcentajes sobre el total. Además en las últimas columnas se genera el archivo de la imagen del clustering realizado de esa configuración y los coeficientes de los índices de medición de las métricas: Silhouette, Calinski-Harabasz, Davies-Bouldin y los índices propios generados (Homogeneidad y K_Index).

Una vez se han generado todas las configuraciones se almacenan los resultados con todas las configuraciones de cada algoritmo (KMeans y KShape). De esta manera podemos tener un archivo completo con todos los datos de las configuraciones y archivos individuales de cada configuración que puedan ser más manejables debido a la gran cantidad de datos que se manejan en el programa.

4.2.3. Mejores configuraciones

Para identificar la configuración óptima de clustering, se analizaron los resultados de las 180 configuraciones generadas mediante el índice WOWA, que integra las métricas normalizadas de Silhouette, Davies-Bouldin (inverso), Calinski-Harabasz, Homogeneidad y K_Index. La normalización escaló todas las métricas entre 0 y 1 usando la fórmula $(\text{valor} - \text{mínimo}) / (\text{máximo} - \text{mínimo})$, con Davies-Bouldin ajustado como $1 - \text{valor_normalizado}$ (ya que valores más bajos son mejores). Esto permite compararlas equitativamente.

Tras evaluar las combinaciones de pesos WOWA (base: [0.1, 0.1, 0.1, 0.5, 0.2], priorizando Homogeneidad en 0.5 y K_Index en 0.2; y variaciones como [0.0, 0.0, 0.0, 0.5, 0.5], [0.0, 0.0, 0.0, 1.0, 0.0], [0.25, 0.25, 0.25, 0.0, 0.25], [0.2, 0.2, 0.2, 0.2, 0.2]), el proceso se realizó en dos niveles: primero, por cada combinación de pesos, agrupando datos por Config_ID, k y Max_Iter, calculando WOWA promedio y seleccionando las 20 mejores únicas por algoritmo (TimeSeriesKMeans y KShape), eliminando duplicados. Segundo, consolidando todas las combinaciones en un ranking global de 20 mejores, combinando resultados sin importar si es TimeSeriesKMeans o KShape, y expresando los pesos utilizados para cada configuración.

En la **Tabla 5** se detalla el pseudocódigo del proceso de selección y generación de resultados:

```
PARA cada métrica EN {Silhouette, DB, CH, Homogeneidad, K_Index}:
    CALCULAR min/max globales
    NORMALIZAR a [0,1]
```

```

    para Davies-Bouldin: 1 - normalizado

FIN_PARA

PARA cada combinación de pesos:

PARA cada configuración:

    ORDENAR métricas normalizadas descendente

    CALCULAR WOWA

FIN_PARA

PARA algoritmo EN {TimeSeriesKMeans, KShape}:

    AGRUPAR por Config_ID, k, Max_Iter

    SELECCIONAR 20 mejores

FIN_PARA

ALMACENAR resultados

FIN_PARA

```

Tabla 5: Pseudocódigo para la generación de resultados. Fuente: Propia

4.3. Resultados obtenidos

La evaluación de los resultados se centra en analizar la configuración de clustering obtenidas para detectar anomalías en las series temporales de par y ángulo, utilizando el operador de agregación WOWA como criterio principal de selección. Este índice agrega las métricas normalizadas de Silhouette (cohesión y separación), Davies-Bouldin (inverso, para compactidad), Calinski-Harabasz (dispersión), Homogeneidad (pureza de clústeres en términos de OK/NOK) y K_Index (penalización por número de clústeres). La normalización escaló todas las métricas entre 0 y 1 mediante la fórmula $(\text{valor} - \text{mínimo}) / (\text{máximo} - \text{mínimo})$, ajustando Davies-Bouldin como $1 - \text{valor_normalizado}$ para que valores más bajos sean favorables.

Se analizaron las 180 configuraciones generadas (60 por variante: TimeSeriesKMeans con DTW, con SoftDTW, y KShape). El proceso detalla las 20 mejores configuraciones obtenidas para distintas combinaciones de pesos WOWA. La notación utilizada para indicar los pesos asignados a cada índice individualmente es: [Silhouette, Davies-Bouldin, Calinski-Harabasz, Homogeneidad, K_Index]:

4.3.1. [0.0, 0.0, 0.0, 0.5, 0.5]

Esta primera variación de pesos se centra exclusivamente en las métricas propuestas (Homogeneidad y K_Index). Los pesos cero de las primeras métricas internas significan que se ignoran por completo, priorizando los otros dos índices con igual importancia. Esto es útil para centrarnos en la pureza de los clústeres y la adecuación del número de clústeres. (**Tabla 6**)

Rank	Algorithm	Conf_ID	k	Max_It	Silhouette	Davi_Bou	Cal_Har	Homog	K_Ind	WOWA
1	TSKM (SoftDTW)	65	6	100	0,457	0,733	1788,623	0,783	0,947	0,916
2	TSKM (SoftDTW)	69	7	150	0,348	1,309	1407,04	0,779	0,895	0,87
3	TSKM (SoftDTW)	68	7	100	0,396	0,805	1672,574	0,779	0,895	0,87
4	TSKM (SoftDTW)	71	8	100	0,357	1,03	1512,902	0,781	0,842	0,851
5	TSKM (SoftDTW)	67	7	50	0,407	0,748	1572,921	0,774	0,895	0,846
6	TSKM (SoftDTW)	72	8	150	0,421	0,813	1713,707	0,779	0,842	0,842
7	TSKM (SoftDTW)	66	6	150	0,431	0,758	1680,987	0,766	0,947	0,837
8	TSKM (DTW)	2	5	100	0,059	1,658	457,87	0,76	1	0,836
9	TSKM (SoftDTW)	61	5	50	0,433	0,752	1625,621	0,759	1	0,832
10	TSKM (SoftDTW)	62	5	100	0,433	0,752	1625,621	0,759	1	0,832
11	TSKM (SoftDTW)	63	5	150	0,433	0,752	1625,621	0,759	1	0,832
12	TSKM (SoftDTW)	64	6	50	0,429	0,759	1677,112	0,764	0,947	0,827
13	TSKM (DTW)	15	9	150	-0,032	3,571	338,353	0,779	0,789	0,814
14	TSKM (DTW)	17	10	100	-0,024	6,298	326,73	0,783	0,737	0,805
15	TSKM (DTW)	7	7	50	0	2,524	373,463	0,764	0,895	0,8
16	TSKM (SoftDTW)	70	8	50	0,307	1,053	1230,446	0,769	0,842	0,795
17	TSKM (SoftDTW)	73	9	50	0,311	1,277	1326,824	0,774	0,789	0,79
18	TSKM (DTW)	14	9	100	-0,026	6,188	358,212	0,774	0,789	0,79
19	TSKM (SoftDTW)	77	10	100	0,276	1,206	1316,595	0,779	0,737	0,786
20	TSKM (DTW)	16	10	50	-0,03	5,653	327,789	0,779	0,737	0,786

Tabla 6: Mejores configuraciones [0.0,0.0,0.0,0.5,0.5]. Fuente: Propia

Ahora se profundiza en las dos mejores configuraciones de esta variación de pesos:

1. Configuración 65: k = 6. Max_It = 100 (WOWA = 0.916):

Con k=6, divide los datos en grupos medianos, logrando una homogeneidad sólida con dos clústeres casi perfectos en división, dos clústeres de mayor tamaño con una composición óptima en porcentaje entre aprietes buenos y malos, y dos clústeres de tamaño mediano cuya composición no es tan adecuada, cercana al 60/40. (**Tabla 7**)

Clúster	Clúster_Size	OK_Count	OK_Percent	NOK_Count	NOK_Percent
0	252	156	61.9	96	38.1
1	206	74	35.9	132	64.1
2	275	240	87.3	35	12.7
3	263	214	81.4	49	18.6
4	109	108	99.1	1	0.9
5	92	5	5.4	87	94.6

Tabla 7: Configuración 65. Fuente: Propia

2. Configuración 69: k=7. Max_It = 150 (WOWA = 0.870):

En este caso el número de clústeres es 7, lo que refina la división, pero baja ligeramente el K_Index. La división de homogeneidad se mantiene alta, pero el clúster extra introduce más variabilidad. Esta división contiene dos clústeres que rozan la perfección en la división, de casi un 100/0, tres clústeres bastante óptimos que se acercan a un valor del 90/10 y un clúster con una división inadecuada cercana al 60/40. (**Tabla 8**)

Clúster	Clúster_Size	OK_Count	OK_Percent	NOK_Count	NOK_Percent
0	302	212	70.2	90	29.8
1	115	7	6.1	108	93.9
2	152	132	86.8	20	13.2
3	102	101	99.0	1	1.0
4	236	101	42.8	135	57.2
5	142	126	88.7	16	11.3
6	148	118	79.7	30	20.3

Tabla 8: Configuración 69. Fuente: Propia

4.3.2. [0.0, 0.0, 0.0, 1.0, 0.0]

Esta variación de pesos es aún más focalizada que la anterior, ignora completamente todas las métricas internas y se centra únicamente en la métrica externa propuesta, la Homogeneidad. (**Tabla 9**)

Rank	Algorithm	Conf_ID	k	Max_It	Silhouette	Davi_Bou	Cal_Har	Homog	K_Ind	WOWA
1	TSKM (SoftDTW)	91	15	50	0,206	1,523	735,43	0,795	0,474	1
2	TSKM (DTW)	55	23	50	-0,067	5,287	225,637	0,789	0,053	0,944
3	TSKM (SoftDTW)	105	19	150	0,053	1,738	532,275	0,789	0,263	0,944
4	TSKM (SoftDTW)	102	18	150	-0,059	1,533	494,631	0,786	0,316	0,916
5	TSKM (SoftDTW)	118	24	50	-0,003	1,608	482,35	0,786	0	0,916
6	TSKM (SoftDTW)	37	17	50	-0,051	6,395	259,415	0,786	0,368	0,916
7	TSKM (SoftDTW)	92	15	100	0,164	1,819	880,566	0,785	0,474	0,907
8	TSKM (DTW)	60	24	150	-0,082	6,211	206,935	0,784	0	0,897
9	TSKM (SoftDTW)	97	17	50	0,004	1,402	555,178	0,784	0,368	0,897
10	TSKM (SoftDTW)	86	13	100	0,201	1,55	1152,61	0,784	0,579	0,897
11	TSKM (SoftDTW)	65	6	100	0,457	0,733	1788,623	0,783	0,947	0,888
12	TSKM (DTW)	17	10	100	-0,024	6,298	326,73	0,783	0,737	0,888
13	TSKM (DTW)	45	19	150	-0,079	4,384	242,275	0,783	0,263	0,888
14	TSKM (SoftDTW)	117	23	150	0,001	1,707	569,118	0,782	0,053	0,879
15	TSKM (SoftDTW)	114	22	150	0,025	1,844	471,444	0,782	0,105	0,879
16	TSKM (SoftDTW)	120	24	150	0,098	1,278	648,946	0,782	0	0,879
17	TSKM (DTW)	19	11	50	-0,074	4,918	314,143	0,782	0,684	0,879
18	TSKM (DTW)	22	12	50	-0,054	8,445	296,421	0,782	0,632	0,879
19	TSKM (SoftDTW)	90	14	150	0,139	1,267	781,292	0,782	0,526	0,879
20	TSKM (DTW)	21	11	150	-0,039	7,754	304,379	0,781	0,684	0,869

Tabla 9: Mejores configuraciones [0,0,0,0,0,0,1,0,0,0]. Fuente: Propia

Ahora se profundiza en las dos mejores configuraciones:

1. Configuración 91: k=15, Max_It=150 (WOWA = 0.944):

Contiene 3 clústeres vacíos (11, 12 y 13), a que los centros iniciales caen en baja densidad, dejando regiones sin asignación. Esto perjudica ya que aumenta artificialmente k, el cual debería ser 12. En este caso al solo tener en cuenta la Homogeneidad, la mayoría de los clústeres son óptimos (con 6 clústeres realmente buenos), salvo dos clústeres malos que no separan bien las tomas buenas de las malas. (Tabla 10)

Clúster	Clúster_Size	OK_Count	OK_Percent	NOK_Count	NOK_Percent
0	230	77	33.5	153	66.5
1	362	302	83.4	60	16.6
2	89	5	5.6	84	94.4
3	48	24	50.0	24	50.0
4	31	29	93.5	2	6.5
5	12	10	83.3	2	16.7
6	27	24	88.9	3	11.1
7	248	183	73.8	65	26.2
8	12	12	100.0	0	0.0
9	71	64	90.1	7	9.9
10	1	1	100.0	0	0.0
14	66	66	100.0	0	0.0

Tabla 10: Configuración 71. Fuente: Propia

2. Configuración 55: k=23, Max_It=100 (WOWA=0.938):

No presenta clústeres vacíos, gracias a una inicialización efectiva que distribuye centros en zonas de densidad media, evitando regiones desérticas. Esto beneficia la Homogeneidad al maximizar el uso de todos los k, aunque k alto fragmenta datos en subgrupos finos, potencialmente diluyendo la pureza global. Al enfocarnos solo en Homogeneidad, la mayoría de los clústeres son óptimos con alto % de división (cercano al 80/20), salvo seis clústeres problemáticos que no separan bien las tomas y bajan el promedio general. (**Tabla 11**)

Clúster	Clúster_Size	OK_Count	OK_Percent	NOK_Count	NOK_Percent
0	25	24	96.0	1	4.0
1	44	16	36.4	28	63.6
2	63	37	58.7	26	41.3
3	72	65	90.3	7	9.7
4	32	32	100.0	0	0.0
5	73	28	38.4	45	61.6
6	66	57	86.4	9	13.6
7	71	30	42.3	41	57.7
8	48	41	85.4	7	14.6
9	31	1	3.2	30	96.8
10	76	56	73.7	20	26.3
11	53	47	88.7	6	11.3
12	49	37	75.5	12	24.5
13	48	43	89.6	5	10.4
14	55	53	96.4	2	3.6
15	42	31	73.8	11	26.2
16	52	3	5.8	49	94.2
17	53	10	18.9	43	81.1
18	27	24	88.9	3	11.1
19	25	21	84.0	4	16.0
20	80	67	83.8	13	16.2
21	62	42	67.7	20	32.3
22	50	32	64.0	18	36.0

Tabla 11: Configuración 55. Fuente: Propia

4.3.3. [0.1, 0.1, 0.1, 0.5, 0.2]

Esta variación de pesos equilibra las métricas, dando un peso débil a Silhouette, Davies-Bouldin y Calinski-Harabasz, un peso moderado a K_Index y un mayor peso a la Homogeneidad. Priorizando la pureza y estabilidad, pero teniendo en cuenta la separación global de los clústeres. (**Tabla 12**)

Rank	Algorithm	Conf_ID	k	Max_It	Silhouette	Davi_Bou	Cal_Har	Homog	K_Ind	WOWA
1	TSKM (SoftDTW)	65	6	100	0,457	0,733	1788,623	0,783	0,947	0,933
2	TSKM (SoftDTW)	68	7	100	0,396	0,805	1672,574	0,779	0,895	0,886
3	TSKM (SoftDTW)	72	8	150	0,421	0,813	1713,707	0,779	0,842	0,881
4	TSKM (SoftDTW)	71	8	100	0,357	1,03	1512,902	0,781	0,842	0,866
5	TSKM (SoftDTW)	67	7	50	0,407	0,748	1572,921	0,774	0,895	0,859
6	TSKM (SoftDTW)	69	7	150	0,348	1,309	1407,04	0,779	0,895	0,858
7	TSKM (SoftDTW)	66	6	150	0,431	0,758	1680,987	0,766	0,947	0,843
8	TSKM (SoftDTW)	64	6	50	0,429	0,759	1677,112	0,764	0,947	0,833
9	TSKM (SoftDTW)	61	5	50	0,433	0,752	1625,621	0,759	1	0,818
10	TSKM (SoftDTW)	62	5	100	0,433	0,752	1625,621	0,759	1	0,818
11	TSKM (SoftDTW)	63	5	150	0,433	0,752	1625,621	0,759	1	0,818
12	TSKM (SoftDTW)	76	10	50	0,312	1,283	1369,489	0,778	0,737	0,812
13	TSKM (SoftDTW)	77	10	100	0,276	1,206	1316,595	0,779	0,737	0,809
14	TSKM (SoftDTW)	73	9	50	0,311	1,277	1326,824	0,774	0,789	0,802
15	TSKM (SoftDTW)	84	12	150	0,285	1,607	1366,09	0,781	0,632	0,797
16	TSKM (SoftDTW)	79	11	50	0,256	1,305	1349,334	0,779	0,684	0,795
17	TSKM (SoftDTW)	70	8	50	0,307	1,053	1230,446	0,769	0,842	0,786
18	TSKM (SoftDTW)	91	15	50	0,206	1,523	735,43	0,795	0,474	0,78
19	TSKM (SoftDTW)	75	9	150	0,278	1,284	1296,89	0,77	0,789	0,776
20	TSKM (SoftDTW)	88	14	50	0,254	1,385	1423,313	0,781	0,526	0,775

Tabla 12: Mejores configuraciones [0,1,0,1,0,1,0,5,0,2]. Fuente: Propia

A continuación, se desarrollan las mejores configuraciones:

1. **Configuración 65:** Explicada en la sección 4.3.1.
2. **Configuración 68:** k=7. Max_It = 100 (WOWA = 0.886):

Esta configuración refina con un k=7, manteniendo unas iteraciones moderadas (Max_It=100). Contiene una homogeneidad sólida, una Silhouette decente y un K_Index estable, aunque Davies-Bouldin sugiere algo de solapamiento. La composición de clústeres destaca el 0 y 1 por su casi perfecta separación, pero penaliza con dos clústeres de gran tamaño que realizan una separación mala de los tornillos OK y NOK. (Tabla 13)

Clúster	Clúster_Size	OK_Count	OK_Percent	NOK_Count	NOK_Percent
0	113	7	6.2	106	93.8
1	94	93	98.9	1	1.1
2	191	150	78.5	41	21.5
3	209	86	41.1	123	58.9
4	198	172	86.9	26	13.1
5	125	112	89.6	13	10.4
6	267	177	66.3	90	33.7

Tabla 13: Configuración 68. Fuente: Propia

4.3.4. [0.25, 0.25, 0.25, 0.0, 0.25]

Esta variación enfatiza las métricas internas del clustering con un mismo peso e ignorando completamente la métrica externa de la Homogeneidad. Esto es ideal para priorizar la compactación y separación global sin preocuparse por la pureza externa, útil para comparar la métrica propuesta. (Tabla 14)

Rank	Algorithm	Conf_ID	k	Max_It	Silhouette	Davi_Bou	Cal_Har	Homog	K_Ind	WOWA
1	TSKM (SoftDTW)	65	6	100	0,457	0,733	1788,623	0,783	0,947	0,986
2	TSKM (SoftDTW)	63	5	150	0,433	0,752	1625,621	0,759	1	0,966
3	TSKM (SoftDTW)	62	5	100	0,433	0,752	1625,621	0,759	1	0,966
4	TSKM (SoftDTW)	61	5	50	0,433	0,752	1625,621	0,759	1	0,966
5	TSKM (SoftDTW)	66	6	150	0,431	0,758	1680,987	0,766	0,947	0,959
6	TSKM (SoftDTW)	64	6	50	0,429	0,759	1677,112	0,764	0,947	0,958
7	TSKM (SoftDTW)	72	8	150	0,421	0,813	1713,707	0,779	0,842	0,931
8	TSKM (SoftDTW)	68	7	100	0,396	0,805	1672,574	0,779	0,895	0,929
9	TSKM (SoftDTW)	67	7	50	0,407	0,748	1572,921	0,774	0,895	0,92
10	TSKM (SoftDTW)	71	8	100	0,357	1,03	1512,902	0,781	0,842	0,871
11	TSKM (SoftDTW)	69	7	150	0,348	1,309	1407,04	0,779	0,895	0,861
12	TSKM (SoftDTW)	70	8	50	0,307	1,053	1230,446	0,769	0,842	0,809
13	TSKM (SoftDTW)	73	9	50	0,311	1,277	1326,824	0,774	0,789	0,806
14	TSKM (SoftDTW)	76	10	50	0,312	1,283	1369,489	0,778	0,737	0,799
15	TSKM (SoftDTW)	75	9	150	0,278	1,284	1296,89	0,77	0,789	0,788
16	TSKM (SoftDTW)	78	10	150	0,28	1,222	1332,461	0,77	0,737	0,782
17	TSKM (SoftDTW)	80	11	100	0,275	1,367	1455,732	0,77	0,684	0,78
18	TSKM (SoftDTW)	77	10	100	0,276	1,206	1316,595	0,779	0,737	0,778
19	TSKM (SoftDTW)	74	9	100	0,26	1,382	1223,042	0,771	0,789	0,768
20	TSKM (SoftDTW)	81	11	150	0,285	1,05	1269,594	0,771	0,684	0,764

Tabla 14: Mejores configuraciones [0.25,0.25,0.25,0.0,0.25]. Fuente: Propia

A continuación, se desarrollan las mejores configuraciones:

1. **Configuración 65:** Explicada en la sección 4.3.1.
2. **Configuración 63:** k=5, Max_It=150 (WOWA = 0.966):

Esta configuración usa k=5 mínimo, lo cual hace que los clústeres sean más estables, pero baja la granularidad. Las iteraciones aseguran una convergencia completa que ajusta bien los centros (Silhouette=0,433), aunque el valor de Davies-Bouldin moderado que indica algo de solapamiento entre clústeres y un Calinski-Harabasz sólido que refleja una cohesión interna fuerte. Los detalles de la configuración reflejan 4 clústeres bastante óptimos con una separación superior al 75/25%, pero con un clúster mal separado y con un volumen alto de tamaño. (**Tabla 15**)

Clúster	Clúster_Size	OK_Count	OK_Percent	NOK_Count	NOK_Percent
0	129	11	8.5	118	91.5
1	299	260	87.0	39	13.0
2	324	252	77.8	72	22.2
3	109	108	99.1	1	0.9
4	336	166	49.4	170	50.6

Tabla 15: Configuración 63. Fuente: Propia

4.3.5. [0.2, 0.2, 0.2, 0.2, 0.2]

Esta variación asigna un valor uniforme a cada métrica, equilibrándolas en el análisis. Esto permite tener en cuenta todas las métricas por igual para dar un análisis integral y sin sesgos, capturando tanto la validez interna de los clústeres como su utilidad externa, lo que favorece configuraciones más versátiles. (**Tabla 16**)

Rank	Algorithm	Conf_ID	k	Max_It	Silhouette	Davi_Bou	Cal_Har	Homog	K_Ind	WOWA
1	TSKM (SoftDTW)	65	6	100	0,457	0,733	1788,623	0,783	0,947	0,966
2	TSKM (SoftDTW)	72	8	150	0,421	0,813	1713,707	0,779	0,842	0,915
3	TSKM (SoftDTW)	66	6	150	0,431	0,758	1680,987	0,766	0,947	0,913
4	TSKM (SoftDTW)	68	7	100	0,396	0,805	1672,574	0,779	0,895	0,913
5	TSKM (SoftDTW)	64	6	50	0,429	0,759	1677,112	0,764	0,947	0,908
6	TSKM (SoftDTW)	61	5	50	0,433	0,752	1625,621	0,759	1	0,906
7	TSKM (SoftDTW)	63	5	150	0,433	0,752	1625,621	0,759	1	0,906
8	TSKM (SoftDTW)	62	5	100	0,433	0,752	1625,621	0,759	1	0,906
9	TSKM (SoftDTW)	67	7	50	0,407	0,748	1572,921	0,774	0,895	0,897
10	TSKM (SoftDTW)	71	8	100	0,357	1,03	1512,902	0,781	0,842	0,871
11	TSKM (SoftDTW)	69	7	150	0,348	1,309	1407,04	0,779	0,895	0,859
12	TSKM (SoftDTW)	76	10	50	0,312	1,283	1369,489	0,778	0,737	0,808
13	TSKM (SoftDTW)	73	9	50	0,311	1,277	1326,824	0,774	0,789	0,806
14	TSKM (SoftDTW)	70	8	50	0,307	1,053	1230,446	0,769	0,842	0,799

15	TSKM (SoftDTW)	77	10	100	0,276	1,206	1316,595	0,779	0,737	0,793
16	TSKM (SoftDTW)	75	9	150	0,278	1,284	1296,89	0,77	0,789	0,784
17	TSKM (SoftDTW)	78	10	150	0,28	1,222	1332,461	0,77	0,737	0,779
18	TSKM (SoftDTW)	80	11	100	0,275	1,367	1455,732	0,77	0,684	0,778
19	TSKM (SoftDTW)	79	11	50	0,256	1,305	1349,334	0,779	0,684	0,777
20	TSKM (SoftDTW)	84	12	150	0,285	1,607	1366,09	0,781	0,632	0,776

Tabla 16: Mejores configuraciones [0,2,0,2,0,2,0,2]. Fuente: Propia

A continuación, se desarrollan las mejores configuraciones:

1. **Configuración 65:** Explicada en la sección 4.3.1
2. **Configuración 72:** k=8, Max_It=150 (WOWA = 0.915):

Esta configuración tiene un k=8 medio para una granularidad equilibrada y un Max_It alto para una convergencia estable, destacando una buena separación y cohesión fuerte, aunque indica algo de solapamiento. La separación según el desglose revela unos clústeres muy bien separados, sobre todo el 1,3 y 6. Pero un clúster de gran tamaño mal separado (Clúster 0). (**Tabla 17**)

Clúster	Clúster_Size	OK_Count	OK_Percent	NOK_Count	NOK_Percent
0	263	155	58.9	108	41.1
1	60	59	98.3	1	1.7
2	214	64	29.9	150	70.1
3	51	1	2.0	50	98.0
4	134	120	89.6	14	10.4
5	230	182	79.1	48	20.9
6	50	50	100.0	0	0.0
7	195	166	85.1	29	14.9

Tabla 17: Configuración 72. Fuente: Propia

4.3.6. Conclusión sobre los resultados

Tras explorar las diferentes ponderaciones de métricas en el análisis de series temporales de motores, se observa que el algoritmo TSKM (TimeSeriesKMeans) emerge como el más efectivo y consistente, apareciendo en todas las configuraciones top 20 de los vectores de pesos evaluados. Esto se debe a su capacidad para manejar distancias temporales (DTW/softDTW) y generar clústeres estables, superando a KShape en WOWA máximo. (**Figura 14**)

TimeSeriesKMeans destaca por su adaptabilidad a datos como los de tornillos, minimizando solapamientos y maximizando homogeneidad sin requerir ajustes complejos.

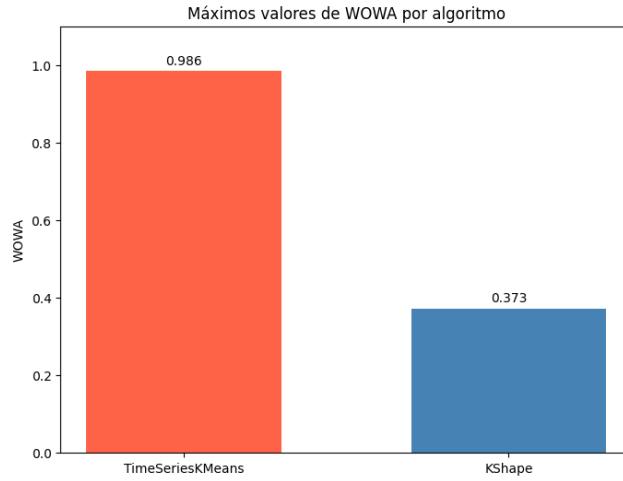


Figura 14: Máximos valores por algoritmo. Fuente: Propia (Python)

Para sintetizar, la **Tabla 18** resume las 20 mejores configuraciones globales, ordenadas por WOWA descendente, incluyendo la columna de pesos asociada a cada una:

Rank	Algorithm	Conf_ID	k	Max_It	Silho	Dav_Bou	Cal_Har	Homog	K_Ind	WOWA	Pesos
1	TSKM (SoftDTW)	65	6	100	0,457	0,733	1788,623	0,783	0,947	0,986	[0.25, 0.25, 0.25, 0, 0.25]
2	TSKM (SoftDTW)	65	6	100	0,457	0,733	1788,623	0,783	0,947	0,966	[0.2, 0.2, 0.2, 0.2, 0.2]
3	TSKM (SoftDTW)	63	5	150	0,433	0,752	1625,621	0,759	1	0,966	[0.25, 0.25, 0.25, 0, 0.25]
4	TSKM (SoftDTW)	62	5	100	0,433	0,752	1625,621	0,759	1	0,966	[0.25, 0.25, 0.25, 0, 0.25]
5	TSKM (SoftDTW)	61	5	50	0,433	0,752	1625,621	0,759	1	0,966	[0.25, 0.25, 0.25, 0, 0.25]
6	TSKM (SoftDTW)	65	6	100	0,457	0,733	1788,623	0,783	0,947	0,933	[0.1, 0.1, 0.1, 0.5, 0.2]
7	TSKM (SoftDTW)	91	15	50	0,206	1,523	735,43	0,795	0,474	0,930	[0, 0, 0, 1.0, 0]
8	TSKM (SoftDTW)	72	8	150	0,421	0,813	1713,707	0,779	0,842	0,931	[0.25, 0.25, 0.25, 0, 0.25]
9	TSKM (SoftDTW)	68	7	100	0,396	0,805	1672,574	0,779	0,895	0,929	[0.25, 0.25, 0.25, 0, 0.25]
10	TSKM (SoftDTW)	72	8	150	0,421	0,813	1713,707	0,779	0,842	0,915	[0.2, 0.2, 0.2, 0.2, 0.2]
11	TSKM (SoftDTW)	66	6	150	0,431	0,758	1680,987	0,766	0,947	0,913	[0.2, 0.2, 0.2, 0.2, 0.2]
12	TSKM (SoftDTW)	68	7	100	0,396	0,805	1672,574	0,779	0,895	0,913	[0.2, 0.2, 0.2, 0.2, 0.2]
13	TSKM (SoftDTW)	64	6	50	0,429	0,759	1677,112	0,764	0,947	0,908	[0.2, 0.2, 0.2, 0.2, 0.2]
14	TSKM (SoftDTW)	61	5	50	0,433	0,752	1625,621	0,759	1	0,906	[0.2, 0.2, 0.2, 0.2, 0.2]
15	TSKM (SoftDTW)	63	5	150	0,433	0,752	1625,621	0,759	1	0,906	[0.2, 0.2, 0.2, 0.2, 0.2]
16	TSKM (SoftDTW)	62	5	100	0,433	0,752	1625,621	0,759	1	0,906	[0.2, 0.2, 0.2, 0.2, 0.2]
17	TSKM (SoftDTW)	67	7	50	0,407	0,748	1572,921	0,774	0,895	0,92	[0.25, 0.25, 0.25, 0, 0.25]

18	TSKM (SoftDTW)	67	7	50	0,407	0,748	1572,921	0,774	0,895	0,897	[0.2, 0.2, 0.2, 0.2, 0.2]
19	TSKM (SoftDTW)	68	7	100	0,396	0,805	1672,574	0,779	0,895	0,886	[0.1, 0.1, 0.1, 0.5, 0.2]
20	TSKM (SoftDTW)	71	8	100	0,357	1,03	1512,902	0,781	0,842	0,871	[0.1, 0.1, 0.1, 0.5, 0.2]

Tabla 18: : Mejores configuraciones globales. Fuente: Propia

La configuración óptima global es la 65 (k=6, Max_It=100), apareciendo en todas las mejores, con un valor de WOVA medio ~0.95 una Silhouette alta (separación buena), Homogeneidad sólida (pureza ~78%), K_Index óptimo (estabilidad sin extremos), y métricas internas compactas (Davi_Bou bajo, Cal_Har alto). Es superior porque integra todo sin sesgos, superando configuraciones con k alto (sobreajuste) o muy bajo (poca granularidad).

A continuación, en la **Tabla 19** y la **Figura 15** se detalla el desglose completo de la configuración 65:

Clúster	0	1	2	3	4	5
Clúster_Size	252	206	275	263	109	92
OK_Count	156	74	240	214	108	5
OK_Percent	61.9	35.9	87.3	81.4	99.1	5.4
NOK_Count	96	132	35	49	1	87
NOK_Percent	38.1	64.1	12.7	18.6	0.9	94.6
Tornillo_0	26	30	36	31	15	12
Tornillo_1	31	29	32	27	14	17
Tornillo_2	29	24	35	45	15	2
Tornillo_3	47	25	21	22	13	22
Tornillo_4	32	23	44	35	14	2
Tornillo_5	29	15	45	40	17	3
Tornillo_6	30	33	34	26	11	15
Tornillo_7	28	27	28	37	10	19
Tornillo_0_Percent	10.3	14.6	13.1	11.8	13.8	13.0
Tornillo_1_Percent	12.3	14.1	11.6	10.3	12.8	18.5
Tornillo_2_Percent	11.5	11.7	12.7	17.1	13.8	2.2
Tornillo_3_Percent	18.7	12.1	7.6	8.4	11.9	23.9
Tornillo_4_Percent	12.7	11.2	16.0	13.3	12.8	2.2
Tornillo_5_Percent	11.5	7.3	16.4	15.2	15.6	3.3
Tornillo_6_Percent	11.9	16.0	12.4	9.9	10.1	16.3
Tornillo_7_Percent	11.1	13.1	10.2	14.1	9.2	20.7

Tabla 19: Desglose completo Configuración 65. Fuente: Propia

Config 65: TimeSeriesKMeans, Metric=softdtw, k=6, Max_Iter=100 (Clusters 0-4)

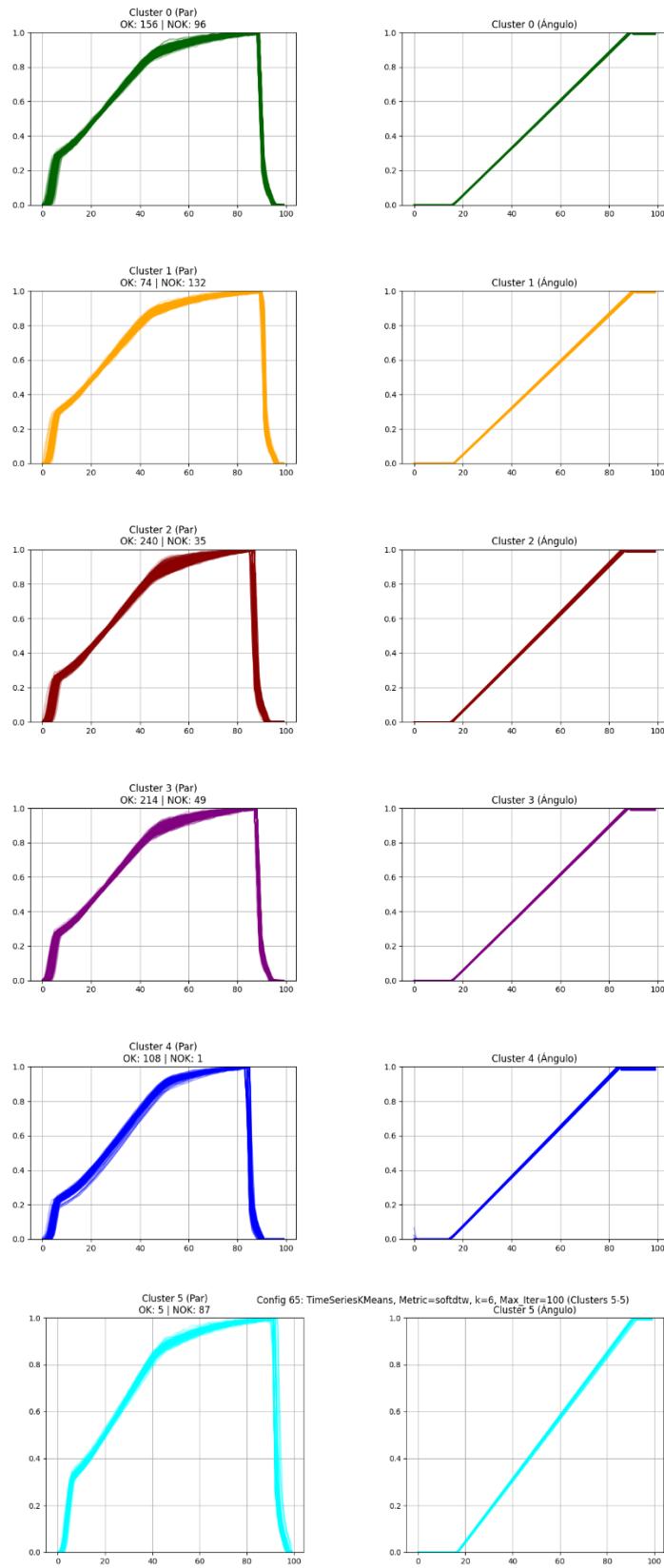


Figura 15: Clustering Configuración 65. Fuente: Propia (Python)

El análisis de la distribución de los tornillos (**Tabla 19**) a lo largo de las configuraciones de clustering revela que la separación de los clústeres no se basa de manera significativa en la atornilladora utilizada. A pesar de evaluar las proporciones de tornillos OK y NOK por clúster, los datos muestran una distribución homogénea entre las diferentes atornilladoras, con porcentajes de tornillos asignados a cada clúster que varían de forma aleatoria y no presentan patrones distintivos asociados a un equipo específico (por ejemplo, clústeres con un 10-20% de tornillos por atornilladora en todas las configuraciones). Esto sugiere que las diferencias en las curvas par-ángulo están más influenciadas por factores inherentes al proceso de atornillado (como par o ángulo) que, por las características específicas de las atornilladoras, lo que indica que el clustering no logra segregar los datos según el equipo utilizado.

La **Figura 15** muestra la visualización de los 5 clústeres generados por la configuración óptima, presentando las curvas de par en la parte izquierda y las curvas de ángulo en la parte derecha. Se puede observar un excelente solapamiento de las curvas dentro de cada clúster, evidenciando alta cohesión interna y buena separación entre clústeres. Esto confirma la calidad del clustering, demostrando que la configuración óptima genera agrupaciones interpretables, compactas y efectivas.

Como conclusión, este análisis ha abordado las curvas par-ángulo para detectar anomalías en el atornillado de motores, un desafío clave en la industria automotriz donde la precisión del par y el ángulo marca la diferencia. Partiendo de un caso de uso práctico, se caracterizaron datos de 200 conjuntos de aprietas (150 OK y 50 NOK), procesándolos para extraer patrones relevantes mediante clustering. La metodología incluyó un preprocessamiento que eliminó intervalos irrelevantes, normalizó curvas y generó gráficas, seguido de la aplicación de algoritmos como TimeSeriesKMeans y KShape con configuraciones variadas (5 a 25 clústeres, 50 a 150 iteraciones, DTW y SoftDTW). Esto permitió evaluar los clústeres con las métricas Silhouette, Davies-Bouldin, Calinski-Harabasz, Homogeneidad y K_Index, combinados en un criterio WOWA con pesos variados para poder saber más acerca de los clústeres y la separación de estos.

La Homogeneidad, métrica desarrollada ad hoc, cuantifica la pureza de cada clúster ponderando su tamaño relativo por el máximo porcentaje de una clase (OK o NOK). Su rol es crucial para priorizar clústeres "puros" ($>78\%$ de una clase), facilitando la detección de desviaciones claras. El K_Index, también propuesto, penaliza configuraciones complejas favoreciendo k bajos ($k=5=1$, $k=10 \approx 0.74$) para una simplicidad interpretativa en entornos industriales.

El análisis exhaustivo de las variaciones en los pesos de las métricas de evaluación, desde una configuración inclinada al índice de Homogeneidad propuesto [0.0, 0.0, 0.0, 1.0, 0.0] hasta los extremos dominados por las métricas internas [0.25, 0.25, 0.25, 0.0, 0.25], revela patrones consistentes en el rendimiento de los algoritmos TimeSeriesKMeans y KShape para el clustering de series temporales orientado a la detección de anomalías (OK/NOK). En todas las combinaciones, TimeSeriesKMeans

demuestra una superioridad clara y robusta, ocupando invariablemente las posiciones superiores de los rankings con valores de WOWA promedio superiores a 0.80, impulsados por una cohesión interna sólida (Silhouette positivo ~0.4), separación entre clústeres efectiva (Davies-Bouldin ~0.8) y dispersión externa óptima (Calinski-Harabasz ~1600-2100), combinados con una Homogeneidad elevada (~0.77-0.78) que asegura pureza superior al 77% y un K_Index equilibrado (~0.8-0.95) que favorece simplicidad sin sacrificar granularidad. Los resultados reflejan que configuraciones como la Conf_ID 65 (k=6, Max_It=100) se repite de manera estable en los tops a lo largo de las variaciones, destacando su versatilidad para datos temporales variables con alineamientos dinámicos, lo que la posiciona como ideal para implementaciones prácticas en entornos industriales donde la precisión en la clasificación prima sobre la complejidad computacional.

Por el contrario, KShape exhibe un rendimiento consistentemente inferior, con WOWA promedio ~0.35-0.25, limitado por clústeres dispersos (Silhouette negativo ~-0.08), separación pobre (Davies-Bouldin ~3.0) y dispersión externa baja (Calinski-Harabasz ~110-150), resultando en una Homogeneidad moderada (~0.69-0.72) que genera clasificaciones mixtas insuficientes para anomalías complejas. Solo en variaciones extremas con peso total en Homogeneidad y número de clústeres mejora relativamente, pero sin superar la brecha global de ~0.55-0.5 puntos en WOWA respecto a TSKM, cuya fortaleza radica en el manejo superior de correlaciones temporales y alineamientos no lineales. (**Tabla 20**)

Rank	Algorithm	Conf_ID	k	Max_It	Silho	Dav_Bou	Cal_Har	Homog	K_Ind	WOWA	Pesos
1	KShape	130	8	50	-0.078	3.061	112.688	0.716	0.842	0.372993	[0.0, 0.0, 0.0, 0.5, 0.5]
2	KShape	131	8	100	-0.078	3.061	112.688	0.716	0.842	0.372993	[0.0, 0.0, 0.0, 0.5, 0.5]
3	KShape	132	8	150	-0.078	3.061	112.688	0.716	0.842	0.372993	[0.0, 0.0, 0.0, 0.5, 0.5]
4	KShape	121	5	50	-0.053	3.207	149.440	0.688	1.000	0.368940	[0.0, 0.0, 0.0, 1.0, 0.0]
5	KShape	123	5	150	-0.053	3.207	149.440	0.688	1.000	0.368940	[0.0, 0.0, 0.0, 1.0, 0.0]
6	KShape	122	5	100	-0.053	3.207	149.440	0.688	1.000	0.368940	[0.0, 0.0, 0.0, 1.0, 0.0]
7	KShape	127	7	50	-0.078	2.878	119.389	0.700	0.895	0.365697	[0.0, 0.0, 0.0, 0.5, 0.5]
8	KShape	128	7	100	-0.078	2.878	119.389	0.700	0.895	0.365697	[0.0, 0.0, 0.0, 0.5, 0.5]
9	KShape	129	7	150	-0.078	2.878	119.389	0.700	0.895	0.365697	[0.0, 0.0, 0.0, 0.5, 0.5]
10	KShape	124	6	50	-0.076	2.953	131.958	0.689	0.947	0.353659	[0.0, 0.0, 0.0, 0.5, 0.5]
11	KShape	125	6	100	-0.076	2.953	131.958	0.689	0.947	0.353659	[0.0, 0.0, 0.0, 0.5, 0.5]
12	KShape	126	6	150	-0.076	2.953	131.958	0.689	0.947	0.353659	[0.0, 0.0, 0.0, 0.5, 0.5]
13	KShape	133	9	50	-0.086	3.196	95.618	0.712	0.789	0.331205	[0.1, 0.1, 0.1, 0.5, 0.2]
14	KShape	134	9	100	-0.086	3.196	95.618	0.712	0.789	0.331205	[0.1, 0.1, 0.1, 0.5, 0.2]
15	KShape	135	9	150	-0.086	3.196	95.618	0.712	0.789	0.331205	[0.1, 0.1, 0.1, 0.5, 0.2]
16	KShape	138	10	150	-0.094	3.493	87.552	0.713	0.737	0.291489	[0.0, 0.0, 0.0, 0.5, 0.5]
17	KShape	136	10	50	-0.094	3.493	87.552	0.713	0.737	0.291489	[0.0, 0.0, 0.0, 0.5, 0.5]
18	KShape	137	10	100	-0.094	3.493	87.552	0.713	0.737	0.291489	[0.0, 0.0, 0.0, 0.5, 0.5]
19	KShape	139	11	50	-0.100	3.345	84.112	0.712	0.684	0.282868	[0.1, 0.1, 0.1, 0.5, 0.2]
20	KShape	140	11	100	-0.100	3.345	84.112	0.712	0.684	0.282868	[0.1, 0.1, 0.1, 0.5, 0.2]

Tabla 20: Mejores configuraciones KShape. Fuente: Propia

Las variaciones en pesos confirman que equilibrar métricas internas (Silhouette, Davies-Bouldin, Calinski-Harabasz y K_Index) con el índice de Homogeneidad genera rankings más estables y configuraciones robustas, elevando WOWA ~0.85-0.90 sin desestabilizar el orden, y favoreciendo números de clústeres entre 5 y 8 con iteraciones en torno a 100 para una combinación óptima entre pureza y simplicidad. Sin embargo, una variación que pondere totalmente a la Homogeneidad, priorización de pureza extrema permite k más altos (hasta 24) con menor penalización, pero reduce la influencia de la cohesión, lo que podría inducir sobreajuste en datasets ruidosos; en estos casos, WOWA se mantiene alto, pero los rankings se vuelven menos sensibles a métricas internas, resaltando el rol de Homogeneidad como métrica principal. Ignorar completamente las métricas internas en la variación no parece una buena opción para el ajuste correcto del clustering.

En síntesis, estos resultados validan TimeSeriesKMeans como el algoritmo preferente para clustering temporal en detección de anomalías, donde SoftDTW destaca por encima de DTW tradicional gracias a su aproximación suave y diferenciable que facilita la optimización basada en gradientes y reduce la sensibilidad a alineamientos locales ruidosos, mejorando la estabilidad en series variables. La mejor configuración global corresponde a la base equilibrada en métricas internas [0.25, 0.25, 0.25, 0.0, 0.25], aunque la adición del índice de Homogeneidad mejora los resultados desarrollados en las configuraciones.

4.4. Bibliotecas, Herramientas y Software Utilizados

Para llevar a cabo este trabajo, se utilizan una serie de herramientas que me han permitido procesar los datos, realizar los análisis y presentar los resultados de forma clara. El análisis comienza con la configuración general de parámetros que establecen el marco del trabajo, detallados en la **Tabla 21**:

Parámetro	Valor	Propósito
metrics	["dtw", "softdtw"]	Define las métricas de distancia (DTW y Soft-DTW) para TimeSeriesKMeans.
k_values	range(5, 25)	Establece el rango de clústeres a probar (5 a 24).
max_iters	[50, 100, 150]	Define las iteraciones máximas para la convergencia de los algoritmos.

n_int	[5, 10]	Controla los intervalos de interpolación en la estandarización de duración.
n_ok_motors	150	Limita el número de motores en estado "OK" a procesar.
n_nok_moto rs	50	Limita el número de motores en estado "NOK" a procesar.
init_method	"k-means++"	Establece el método de inicialización de centroides en TimeSeriesKMeans.
n_init	1	Define el número de inicializaciones en KShape (implícito).
wowa_weight s	[0.1, 0.1, 0.1, 0.5, 0.2]	Asigna pesos a las métricas para calcular el índice WOVA.

Tabla 21: Configuración de los parámetros: Fuente: Propia

Estos parámetros no son aleatorios; se ajustaron mediante un proceso iterativo basado en las características de las curvas par-ángulo:

Uno de los primeros es metrics, una lista que incluye ["dtw", "softdtw"], que especifica las métricas de distancia utilizadas por TimeSeriesKMeans. Se elige DTW (Dynamic Time Warping) por su capacidad para alinear temporalmente las curvas y capturar variaciones en el par, mientras que Soft-DTW añade flexibilidad con un suavizado implícito, lo que se prueba para mejorar la robustez ante ruido. Este parámetro permite comparar ambos enfoques y evaluar su impacto en la detección de anomalías.

Otro parámetro clave es k_values, definido como `list(range(5, 25))`, que establece el rango de clústeres a probar. Lo fijé entre 5 y 24 porque quería explorar una variedad de agrupaciones que reflejaban tanto patrones normales como anómalos en las curvas, basándome en la diversidad de los datos de entrada. Lo ajusté tras observar que valores menores a 5 generaban clústeres demasiado generales, mientras que más de 24 fragmentaban excesivamente los datos.

El parámetro max_iters, con valores [50, 100, 150], define las iteraciones máximas para la convergencia de TimeSeriesKMeans y KShape. Lo configuré en este rango tras probar que 50 iteraciones a veces no alcanzaban la convergencia en datasets grandes, mientras que 150 ofrecía mejoras marginales, por lo que 100 se convirtió en un punto intermedio efectivo. Este ajuste asegura que los algoritmos se estabilicen sin consumir recursos innecesarios.

Para TimeSeriesKMeans, el parámetro `init_method = "k-means++"` establece el método de inicialización de los centroides. Lo seleccioné porque k-means++ mejora la convergencia inicial al elegir puntos iniciales más representativos, reduciendo el riesgo de caer en óptimos locales, algo que noté al comparar con inicializaciones aleatorias en pruebas previas.

En K-Shape, aunque no se especifica explícitamente un parámetro de inicialización, el valor implícito de `n_init = 1` (definido en el código) indica una sola inicialización. Lo mantuve así por simplicidad y para alinear el tiempo de cómputo con TimeSeriesKMeans, aunque consideré aumentar a 5 en futuros ajustes si la estabilidad lo requiriera.

Todos estos parámetros pueden variarse al inicio del programa para buscar configuraciones óptimas, incluyendo ajustes en `n_ok_motors` y `n_nok_motors` según disponibilidad de datos.

A continuación, se detallan las principales bibliotecas y herramientas utilizadas:

- **Python:** Se utiliza este lenguaje como base para todo el proyecto, eligiendo la versión 3.12 por su estabilidad y la amplia comunidad que lo apoya, ideal para tareas de ciencia de datos. Me ha dado la flexibilidad necesaria para estructurar el flujo de trabajo.
- **pandas:** Esta herramienta ha sido clave para organizar los datos en tablas manejables. Me ha permitido trabajar con las mediciones como si fueran hojas de cálculo, facilitando su análisis y preparación.
- **numpy:** Para los cálculos numéricos, especialmente con matrices y ajustes de series temporales, esta biblioteca ha sido mi aliada. Su rapidez ha hecho que los procesos pesados fueran más eficientes.
- **matplotlib y gridspec:** Con matplotlib se crean los gráficos que muestran las curvas de par y ángulo, mientras que gridspec me ha ayudado a organizar varios subgráficos en una misma figura, mejorando la visualización comparativa.
- **tslearn:** Esta biblioteca especializada en series temporales me ha proporcionado los algoritmos necesarios para el clustering, como TimeSeriesKMeans y KShape, adaptados perfectamente a mi caso de estudio.
- **sklearn:** Se ha recurrido a esta herramienta para evaluar los clústeres con métricas como la puntuación de Silhouette, el índice de Davies-Bouldin y el de Calinski-Harabasz, además de estandarizar los datos antes del análisis.
- **json:** Me ha servido para guardar los datos procesados en un formato ligero, ideal para almacenar las series temporales de forma ordenada y accesible.

- **openpyxl**: Con esta biblioteca se han generado los reportes en Excel, añadiendo tablas estilizadas que hacen que los resultados sean más profesionales y fáciles de compartir.
- **collections (Counter)**: Este módulo me ha ayudado a contar rápidamente la distribución de estados (OK y NOK) entre los motores, dándome una visión inicial útil del conjunto de datos.

Todas estas herramientas se han integrado de manera que el proceso fluya desde la carga de datos hasta la generación de informes y gráficos, asegurando un análisis sólido y resultados comprensibles para su uso en un entorno industrial.

5. CONCLUSIONES Y TRABAJO FUTURO

5.1. Conclusiones

A lo largo de este trabajo se han abordado con éxito los objetivos planteados y se ha conseguido realizar una contribución significativa al campo del control de calidad en procesos de atornillado industrial, tomando como caso de estudio los motores ensamblados en la empresa HORSE.

El primer y principal objetivo era analizar y proponer soluciones basadas en técnicas de analítica de datos, especialmente algoritmos de clustering orientados a la detección y prevención de anomalías en series temporales de par y ángulo generadas durante el proceso de atornillado. Para ello, se ha diseñado una metodología detallada que abarca desde el preprocesamiento de los datos, la normalización de los perfiles de cada tornillo, hasta la aplicación iterativa y sistemática de diferentes configuraciones de clustering que permiten segmentar y analizar de forma objetiva grandes volúmenes de información provenientes de sensores industriales.

Entre las principales aportaciones destacan el desarrollo y puesta en práctica de un sistema de selección de parámetros configurables que facilita la comparación rigurosa entre distintas variantes de algoritmos de agrupamiento, como TimeSeriesKMeans y KShape, además de comparar las métricas DTW y SoftDTW en el análisis de series temporales.

El enfoque propuesto permite no solo identificar patrones normales y anómalos en los datos, sino también discriminar de forma robusta entre motores funcionales y aquellos con defectos, aportando indicadores cuantitativos como la Silhouette, Calinski-Harabasz, Davies-Bouldin, Homogeneidad y K_Index, además del operador de agregación de métricas WOWA, que ha demostrado ser muy útil para sintetizar la calidad de las configuraciones ensayadas.

Cabe subrayar que la solución final adoptada, basada en TimeSeriesKMeans con SoftDTW y un número óptimo de clústeres de seis, ha mostrado una capacidad sobresaliente para agrupar curvas de par-ángulo y detectar anomalías características del proceso de atornillado en los motores de HORSE, superando el rendimiento de otras alternativas estudiadas. Este resultado, validado por la comparación sistemática de cientos de configuraciones, se apoya además en la generación automatizada de informes y visualizaciones, lo cual facilita enormemente su integración en entornos reales de fabricación.

Esta metodología se ha aplicado a un problema real de producción industrial, validada con datos de 200 motores ensamblados en HORSE (150 funcionales y 50 con anomalías), demostrando su efectividad en la detección de desviaciones del

comportamiento nominal y su potencial para reducir rechazos y mejorar la fiabilidad en entornos operativos.

No menos importante es el valor añadido que aporta el desarrollo operativo de todo el proceso en Python, apoyándose en bibliotecas especializadas para la manipulación y visualización de datos, lo que garantiza la reproducibilidad y escalabilidad del análisis en contextos industriales. Se ha logrado así avanzar sustancialmente en la automatización del diagnóstico de fallos en el atornillado y en la mejora continua del control de calidad.

5.2. Líneas de Trabajo Futuro

El trabajo realizado abre numerosas posibilidades de ampliación y mejora en futuras investigaciones y desarrollos, algunas de las cuales se enumeran a continuación:

- **Extensión de la base de datos:** Como evolución natural, resulta fundamental ampliar el número de motores y atornilladoras analizadas, recogiendo más muestras y expandiendo la representatividad tanto en cantidad como en diversidad de casos. Esto permitiría fortalecer la validez estadística de los resultados y abordar variantes o modelos de motores con particularidades distintas, adaptando el análisis a nuevos escenarios industriales.
- **Incorporación de nuevas técnicas analíticas:** La metodología puede enriquecerse probando algoritmos de clustering alternativos (como DBSCAN, espectrales o modelos basados en aprendizaje profundo) que faciliten la visualización y comprensión de patrones ocultos en los datos de par y ángulo. Igualmente, la integración de técnicas de aprendizaje supervisado podría servir para comparar el rendimiento contra los enfoques no supervisados utilizados hasta ahora.
- **Desarrollo de dashboards interactivos:** De cara a la aplicación práctica en planta, se plantea construir paneles de control visuales que permitan a los técnicos y responsables de calidad ver en tiempo real el estado de cada motor y tornillo, facilitando la identificación instantánea de posibles anomalías. Esto requeriría el desarrollo de aplicaciones web o interfaces gráficas que conecten con el sistema de análisis y muestren indicadores de agrupamiento, perfiles de par-ángulo, alertas, y estadísticas clave, todo ello de manera intuitiva y flexible.
- **Implementación en streaming en la planta de HORSE:** Una línea especialmente relevante es la adaptación del sistema para su uso en streaming, integrándolo directamente en el proceso de producción de HORSE. Esto permitiría analizar las curvas de par-ángulo en tiempo real, detectando anomalías durante el propio ensamblaje y facilitando la toma de decisiones inmediata, la trazabilidad automática y la optimización de los tiempos de parada necesarios para revisiones o correcciones. Para ello sería necesaria

una arquitectura capaz de manejar grandes volúmenes de datos en tiempo real, así como mecanismos de alerta y retroalimentación directa para los operarios.

En definitiva, el camino iniciado con este trabajo pone las bases para una mejora sustancial en el análisis predictivo y el control de calidad en procesos de atornillado industrial, no solo contribuyendo a la fiabilidad y seguridad de los motores de HORSE sino también abriendo la puerta a aplicaciones similares en muchas otras áreas de la industria moderna.

Bibliografía

- [1] R. G. Estrada Moyano, «Estudio de los modos de fallo en uniones atornilladas de materiales compuestos biodegradables. Universidad Politécnica de Madrid,» 2019. [En línea]. Available: <https://hdl.handle.net/10016/30331>.
- [2] Y. Ramos-Alfonso, J. A. Acevedo Suárez, F. Ramírez Betancourt y E. García Rodríguez, «Modelo de gestión de la eficiencia basado en los costos de la calidad con enfoque generalizador,» *SCIELO*, vol. 7, nº 1, pp. 47-54, 2016.
- [3] M. P. Astier Peña, J. Á. Maderuelo-Fernández, M. T. Jiménez Julvez, M. D. Martín Rodríguez, J. Palacio Lapuente y G. Oliveras Cañada, «Análisis proactivo del riesgo: el análisis modal de fallos y efectos (AMFE),» Universitat Autónoma de Barcelona, 2010.
- [4] I. 9001, Sistemas de gestión de la calidad, 2015.
- [5] J. Carrasco Castillo, «Algoritmos de inteligencia computacional para abordar problemas de detección de anomalías en entornos big data. Universidad de La Rioja,» 2023. [En línea]. Available: <https://dialnet.unirioja.es/servlet/tesis?codigo=315195>.
- [6] D. P. Bistriceanu, «Influence of tightening screws process in high-pressure injection pump,» de *IOP Conference Series: Materials Science and Engineering*, 2020.
- [7] L. Cheng, H. Zhang, H. Lu y X. Liu, «Automatic Assembly Technology of Dense Small Screws for Flat Panel Parts,» *Applied Sciences*, vol. 13, nº 14, pp. 1-18, 2023.
- [8] R. Zhang y C. Ren, «Sentiment time series clustering of Danmu videos based on BERT fine-tuning and SBD-K-shape,» *Electronic Library*, vol. 42, nº 4, pp. 553-575, 2024.
- [9] D. Bai, «Unsupervised method for detecting surface defects in steel based on joint optimization of pseudo-labeling and clustering,» *Measurement Science and Technology*, p. 10, 2024.
- [10] N. Olivero, «Análisis de series temporales,» Academia EDU, 2007.
- [11] J. F. Marín Blanca, «Automatización de proceso de atornillado mecánico a través de elementos de control electrónico. Universidad Carlos III de Madrid,» 2010. [En línea]. Available: <https://e->

archivo.uc3m.es/entities/publication/0c15eab8-f311-4d71-93a7-eaaa6f91f315.

- [12] J. E. Molina Muñoz, «Análisis bibliométrico del uso de machine learning en finanzas a través de un modelo de K-Means. Universidad del Rosario,» 2021. [En línea]. Available: <https://pure.urosario.edu.co/en/publications/an%C3%A1lisis-bibliometrico-del-uso-de-machine-learning-en-finanzas-a->.
- [13] N. Naranjo Fernández, C. Guardiola Albert, H. Aguilera , C. Serrano Hidalgo y E. Montero González, «Clustering groundwater level time series of the exploited Almonte-Marismas aquifer in Southwest Spain,» *Water*, vol. 12, nº 4, p. 1063, 2020.
- [14] M. A. Chong Fuentes y R. Aguilar, «Proyección de series de tiempo para el consumo de la energía eléctrica a clientes residenciales en Ecuador,» *ESPOL*, pp. 1-66, 2016.
- [15] J. Lorenzo-Ginori y Y. Ruiz, «Algoritmos para la corrección de latidos ectópicos en series temporales de la señal electrocardiográfica,» ResearchGate, 2006.
- [16] A. Manzano Santos, «Técnicas de agrupación sobre series temporales para la detección de anomalías en un entorno industrial. Universidad de Valladolid,» [En línea]. Available: <https://uvadoc.uva.es/handle/10324/43730>.
- [17] D. Giordano, M. Mellia y T. Cerquitelli, «K-MDTSC: K-Multi-Dimensional Time-Series Clustering Algorithm,» *Electronics*, vol. 10, nº 11, p. 1166, 2021.
- [18] X. Huang, Y. Ye, L. Xiong, R. Lau, N. Jiang y S. Wang, «Time series k-means: A new k-means type smooth subspace clustering for time series data,» *ScienceDirect*, Vols. %1 de %2367-368, pp. 1-13, 2016.
- [19] M. Cuturi y M. Blondel, «Soft-DTW: a Differentiable Loss Function for Time-Series,» de *ICML*, 2017.
- [20] S. Dalta, C. Karmakar y M. Palaniswami, «Averaging Methods using Dynamic Time Warping for Time Series Classification,» de *IEEE*, 2020.
- [21] M. Mahdavi, A. Kazemnejad, A. Asosheh y D. Khalili, «Cardiovascular risk patterns through AI-enhanced clustering of longitudinal health data,» *Journal of Diabetes & Metabolic Disorders*, vol. 24, nº 84, pp. 1-10, 2025.
- [22] B. Chong, «K-Means algorithm: a brief review,» Francis Press, 2021.

- [23] C. Betancort Rodríguez, «Clustering Inteligente sobre Series Temporales. Universidad Complutense de Madrid,» 2023. [En línea]. Available: <https://docta.ucm.es/entities/publication/6caf2353-f7b9-429d-994f-aad8d4dfc21a>.
- [24] J. Paparrizos y L. Gravano, «k-Shape: Efficient and Accurate Clustering of Time Series,» de ACM SIGMOD, 2015.
- [25] M. Jiedeerbieke, T. Li, Y. Chao, H. Qi y C. Lin, «Gravity Dam Deformation Prediction Model Based on I-KShape and ZOA-BiLSTM,» de IEEE, 2024.
- [26] Q. Tang, Q. Zhang , Y. Deng, M. Zhu, Y. Su, J. Liu y X. Wang, «Typical Power Load Profiles Shape Clustering Analysis Based on Adaptive Piecewise Aggregate Approximation,» de IOP Conference Series: Materials Science and Engineering, 2018.
- [27] W. Ding, W. Li, Z. Zhang, C. Wan, J. Duan y S. Lu, «Time-Varying Gaussian Markov Random Fields Learning for Multivariate Time Series Clustering,» de IEEE, 2023.
- [28] P. J. Rousseeuw, «Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis,» *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53-65, 1987.
- [29] T. Calinski y J. Harabasz, «A dendrite method for cluster analysis,» *Communications in Statistics*, vol. 3, nº 1, pp. 1-27, 1974.
- [30] D. L. Davies y D. W. Bouldin, «A Cluster Separation Measure,» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vols. %1 de %2PAMI-1, nº 2, pp. 224-227, 1979.
- [31] V. Torra, «The weighted OWA operator,» *International Journal of Intelligent Systems*, vol. 12, nº 2, pp. 153-166, 1998.
- [32] M. Cheng, J. Wang, X. Wang, B. Huang y C. Fan, «A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data,» *Frontiers in Energy Research*, vol. 9, pp. 1-17, 2021.
- [33] L. E. Aik, T. W. Choon y M. S. Abu, «K-means Algorithm Based on Flower Pollination Algorithm and Calinski-Harabasz Index,» de *Journal of Physics: Conference Series*, 2023.
- [34] A. Dudek, «Silhouette Index as Clustering Evaluation Tool,» de *Advances in Data Analysis with Computational Intelligence Methods*, Springer, 2020, pp. 21-37.

- [35] F. Ros, R. Riad y S. Guillaume, «PDBI: A partitioning Davies-Bouldin index for clustering evaluation,» *Neurocomputing*, vol. 527, pp. 71-80, 2023.