

Daniel Ramos

Tesis-v3-Daniel-Ramos-Rivera.pdf

 Tecnológico Nacional de México

Detalles del documento

Identificador de la entrega

trn:oid:::20755:421283390

Fecha de entrega

16 ene 2025, 7:55 a.m. GMT-8

Fecha de descarga

16 ene 2025, 7:58 a.m. GMT-8

Nombre de archivo

Tesis-v3-Daniel-Ramos-Rivera.pdf

Tamaño de archivo

2.7 MB

98 Páginas

22,879 Palabras

119,327 Caracteres

14% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- Fuentes de Internet
- Base de datos de Crossref
- Base de datos de contenido publicado de Crossref

Fuentes principales

- 0%  Fuentes de Internet
- 0%  Publicaciones
- 14%  Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alerta de integridad para revisión

-  **Texto oculto**
96 caracteres sospechosos en N.º de página
El texto es alterado para mezclarse con el fondo blanco del documento.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

Fuentes principales

- 0% Fuentes de Internet
- 0% Publicaciones
- 14% Trabajos entregados (trabajos del estudiante)

Fuentes principales

Las fuentes con el mayor número de coincidencias dentro de la entrega. Las fuentes superpuestas no se mostrarán.

1	Trabajos entregados	imfice on 2024-08-21	<1%
2	Trabajos entregados	University of Oulu on 2023-05-29	<1%
3	Trabajos entregados	Associatie K.U.Leuven on 2022-08-25	<1%
4	Trabajos entregados	Monash University on 2024-05-26	<1%
5	Trabajos entregados	Monash University on 2024-10-18	<1%
6	Trabajos entregados	Erasmus University of Rotterdam on 2021-08-30	<1%
7	Trabajos entregados	University of Oxford on 2024-08-07	<1%
8	Trabajos entregados	Unviersidad de Granada on 2022-07-04	<1%
9	Trabajos entregados	urjc on 2023-02-28	<1%
10	Trabajos entregados	Universidad Politécnica de Madrid on 2024-09-10	<1%
11	Trabajos entregados	University of Hull on 2022-05-13	<1%

12	Trabajos entregados	Escuela Superior Politécnica del Litoral on 2022-07-02	<1%
13	Trabajos entregados	University of Huddersfield on 2023-04-19	<1%
14	Trabajos entregados	SRH FernHochschule Riedlingen on 2024-12-25	<1%
15	Trabajos entregados	UNIBA on 2024-04-14	<1%
16	Trabajos entregados	Bournemouth University on 2024-05-17	<1%
17	Trabajos entregados	Universidad Carlos III de Madrid - EUR on 2024-06-06	<1%
18	Trabajos entregados	Arab Academy for Science, Technology & Maritime Transport CAIRO on 2023-01-18	<1%
19	Trabajos entregados	Sheffield Hallam University on 2021-05-20	<1%
20	Trabajos entregados	Universidad Rey Juan Carlos on 2024-07-11	<1%
21	Trabajos entregados	Universidad TecMilenio on 2024-06-15	<1%
22	Trabajos entregados	Universidad de Deusto on 2021-01-12	<1%
23	Trabajos entregados	Vrije Universiteit Amsterdam on 2024-12-20	<1%
24	Trabajos entregados	Jyväskylä University on 2024-11-12	<1%
25	Trabajos entregados	Universidad Autónoma de Nuevo León on 2019-11-26	<1%

26	Trabajos entregados	Universidad Francisco de Vitoria on 2023-03-04	<1%
27	Trabajos entregados	Universidad Nacional de Colombia on 2017-08-29	<1%
28	Trabajos entregados	Florida State University on 2021-07-21	<1%
29	Trabajos entregados	Universidad de Valladolid on 2024-07-22	<1%
30	Trabajos entregados	University of Ulster on 2023-12-17	<1%
31	Trabajos entregados	Unviersidad de Granada on 2024-09-06	<1%
32	Trabajos entregados	Pentecost University College on 2020-09-05	<1%
33	Trabajos entregados	Universidad Internacional de la Rioja on 2023-07-18	<1%
34	Trabajos entregados	Universidad de Murcia on 2019-06-26	<1%
35	Trabajos entregados	Universiteit van Amsterdam on 2024-02-02	<1%
36	Trabajos entregados	University of Northumbria at Newcastle on 2020-05-28	<1%
37	Trabajos entregados	University of Strathclyde on 2024-04-25	<1%
38	Trabajos entregados	Texas State University- San Marcos on 2024-11-19	<1%
39	Trabajos entregados	Universiti Kebangsaan Malaysia on 2024-06-20	<1%

40	Trabajos entregados	Monash University on 2024-09-13	<1%
41	Trabajos entregados	University of Aberdeen on 2024-02-15	<1%
42	Trabajos entregados	University of Limerick on 2024-04-05	<1%
43	Trabajos entregados	University of Sunderland on 2022-05-03	<1%
44	Trabajos entregados	Adtalem Global Education on 2024-02-05	<1%
45	Trabajos entregados	Leiden University on 2021-02-05	<1%
46	Trabajos entregados	Monash University on 2021-04-30	<1%
47	Trabajos entregados	British University in Egypt on 2024-06-19	<1%
48	Trabajos entregados	Coventry University on 2022-07-11	<1%
49	Trabajos entregados	Monash University on 2024-10-19	<1%
50	Trabajos entregados	Corporación Universitaria Minuto de Dios, UNIMINUTO on 2023-10-14	<1%
51	Trabajos entregados	The Scientific & Technological Research Council of Turkey (TUBITAK) on 2024-08-06	<1%
52	Trabajos entregados	Sophia University on 2021-05-31	<1%
53	Trabajos entregados	SASTRA University on 2013-08-16	<1%

54	Trabajos entregados	Queen Mary and Westfield College on 2024-08-23	<1%
55	Trabajos entregados	Universidad de Valladolid on 2020-09-18	<1%
56	Trabajos entregados	Tilburg University on 2024-11-26	<1%
57	Trabajos entregados	University of East Anglia on 2022-09-01	<1%
58	Trabajos entregados	Universidad Internacional de la Rioja on 2023-06-03	<1%
59	Trabajos entregados	Tilburg University on 2024-03-11	<1%
60	Trabajos entregados	Universidad de Málaga - Tii on 2024-06-19	<1%
61	Trabajos entregados	City University on 2019-12-13	<1%
62	Trabajos entregados	Indian Institute of Technology, Madras on 2024-10-08	<1%
63	Trabajos entregados	Universidad Internacional de la Rioja on 2024-12-19	<1%
64	Trabajos entregados	Universidad de Guadalajara on 2024-02-12	<1%
65	Trabajos entregados	Universidad Católica San Pablo on 2017-06-21	<1%
66	Trabajos entregados	University of Teesside on 2022-05-04	<1%
67	Trabajos entregados	Cardiff University on 2024-04-17	<1%

68	Trabajos entregados	Consortio CIXUG on 2018-11-21	<1%
69	Trabajos entregados	Pontificia Universidad Javeriana Cali on 2024-09-07	<1%
70	Trabajos entregados	UTEC Universidad de Ingenieria & Tecnologia on 2024-05-23	<1%
71	Trabajos entregados	Universidad EAN on 2023-11-06	<1%
72	Trabajos entregados	Universidad Tecnologica del Peru on 2023-12-15	<1%
73	Trabajos entregados	Universidad de Costa Rica on 2022-03-07	<1%
74	Trabajos entregados	University of Hong Kong on 2024-05-02	<1%
75	Trabajos entregados	Colegio Universitario de Estudios Financiero on 2024-06-13	<1%
76	Trabajos entregados	Cranfield University on 2019-11-02	<1%
77	Trabajos entregados	Infile on 2024-12-02	<1%
78	Trabajos entregados	Manchester Metropolitan University on 2024-05-30	<1%
79	Trabajos entregados	Universidad de Málaga - Tii on 2024-09-20	<1%
80	Trabajos entregados	Queen Mary and Westfield College on 2024-03-11	<1%
81	Trabajos entregados	Universidad Católica San Pablo on 2024-02-28	<1%

82	Trabajos entregados	Universidad de Xalapa A. C. on 2024-09-03	<1%
83	Trabajos entregados	Vrije Universiteit Amsterdam on 2024-11-27	<1%
84	Trabajos entregados	Associatie K.U.Leuven on 2025-01-15	<1%
85	Trabajos entregados	Queensland University of Technology on 2014-12-12	<1%
86	Trabajos entregados	Universidad Internacional de la Rioja on 2021-07-22	<1%
87	Trabajos entregados	Universidad Internacional de la Rioja on 2022-03-01	<1%
88	Trabajos entregados	Universidad de Sevilla on 2019-10-15	<1%
89	Trabajos entregados	University of Bedfordshire on 2015-01-09	<1%
90	Trabajos entregados	University of Technology, Sydney on 2017-09-22	<1%
91	Trabajos entregados	Brunel University on 2016-08-19	<1%
92	Trabajos entregados	Universidad Anahuac México Sur on 2023-05-08	<1%
93	Trabajos entregados	Universidad Autónoma Metropolitana-Xochimilco on 2022-06-07	<1%
94	Trabajos entregados	consultoriadeserviciosformativos on 2024-07-22	<1%
95	Trabajos entregados	Corporacion Mexicana de Investigacion en Matriales S.A. de C.V. (COMMIMSA) on ...	<1%

96	Trabajos entregados	Intercollege on 2021-08-22	<1%
97	Trabajos entregados	Universidad Pontificia Bolivariana on 2019-06-26	<1%
98	Trabajos entregados	University of Canberra on 2024-11-03	<1%
99	Trabajos entregados	imfice on 2024-10-02	<1%
100	Trabajos entregados	Infile on 2023-07-05	<1%
101	Trabajos entregados	Universidad Internacional de la Rioja on 2024-12-22	<1%
102	Trabajos entregados	Universidad de Córdoba on 2021-05-10	<1%
103	Trabajos entregados	Universidad de San Martín de Porres on 2020-12-20	<1%
104	Trabajos entregados	Universitat Politècnica de València on 2021-09-27	<1%
105	Trabajos entregados	BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA BIBLIOTECA on 2024-01-11	<1%
106	Trabajos entregados	CONACYT on 2018-05-15	<1%
107	Trabajos entregados	Colegio Universitario de Estudios Financiero on 2023-06-12	<1%
108	Trabajos entregados	Corporación Universitaria Minuto de Dios, UNIMINUTO on 2024-09-11	<1%
109	Trabajos entregados	Facultad Latinoamericana de Ciencias Sociales (FLACSO) - Sede Ecuador on 2019-0...	<1%

110	Trabajos entregados	Northeastern Illinois University on 2019-07-08	<1%
111	Trabajos entregados	Universidad Internacional de la Rioja on 2020-07-23	<1%
112	Trabajos entregados	Universidad Internacional de la Rioja on 2023-07-19	<1%
113	Trabajos entregados	Universidad Tecnica De Ambato- Direccion de Investigacion y Desarrollo , DIDE o...	<1%
114	Trabajos entregados	Universidad de Alcalá on 2021-11-15	<1%
115	Trabajos entregados	Universidad de Almeria on 2017-08-09	<1%
116	Trabajos entregados	Universidad de León on 2021-09-08	<1%
117	Trabajos entregados	Vrije Universiteit Amsterdam on 2024-11-27	<1%
118	Trabajos entregados	Anna University on 2024-05-09	<1%
119	Trabajos entregados	Consorcio CIXUG on 2017-12-15	<1%
120	Trabajos entregados	Consorcio CIXUG on 2023-12-21	<1%
121	Trabajos entregados	Facultad Latinoamericana de Ciencias Sociales (FLACSO) - Sede Ecuador on 2017-1...	<1%
122	Trabajos entregados	Facultad Latinoamericana de Ciencias Sociales (FLACSO) - Sede Ecuador on 2019-0...	<1%
123	Trabajos entregados	Infile on 2019-10-08	<1%

124	Trabajos entregados	Malta College of Arts,Science and Technology on 2024-06-09	<1%
125	Trabajos entregados	UNIBA on 2024-09-02	<1%
126	Trabajos entregados	Universidad Abierta para Adultos on 2020-02-20	<1%
127	Trabajos entregados	Universidad Carlos III de Madrid on 2021-09-07	<1%
128	Trabajos entregados	Universidad Internacional de la Rioja on 2024-11-04	<1%
129	Trabajos entregados	Universidad Internacional de la Rioja on 2024-12-09	<1%
130	Trabajos entregados	Universidad Nacional de Educación a Distancia on 2021-09-20	<1%
131	Trabajos entregados	Universidad San Francisco de Quito on 2024-04-22	<1%
132	Trabajos entregados	Universidad TecMilenio on 2024-01-20	<1%
133	Trabajos entregados	Universidad de Alcalá on 2022-07-21	<1%
134	Trabajos entregados	Universidad de Alicante on 2022-07-08	<1%
135	Trabajos entregados	University of Derby on 2012-06-07	<1%
136	Trabajos entregados	University of St. Gallen on 2022-06-30	<1%
137	Trabajos entregados	unifranz on 2024-08-15	<1%

138	Trabajos entregados	Associatie K.U.Leuven on 2022-05-20	<1%
139	Trabajos entregados	Universidad de Valladolid on 2020-09-10	<1%
140	Trabajos entregados	Birla Institute of Technology and Science Pilani on 2018-05-03	<1%
141	Trabajos entregados	FFHS Fernfachhochschule Schweiz on 2024-08-07	<1%
142	Trabajos entregados	Institución Tecnológica Metropolitana de Medellín on 2020-10-24	<1%
143	Trabajos entregados	Radboud Universiteit Nijmegen on 2017-08-02	<1%
144	Trabajos entregados	Turun yliopisto on 2015-04-01	<1%
145	Trabajos entregados	University of Kassel on 2020-09-09	<1%



Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Mexicali
Subdirección académica

INSTITUTO TECNOLÓGICO DE MEXICALI
DIVISIÓN DE ESTUDIOS DE POSGRADO

DEPARTAMENTO DE SISTEMAS COMPUTACIONALES

TESIS

Aprendizaje máquina para la clasificación de señales de series de tiempo

QUE PARA OBTENER EL TÍTULO DE
MAESTRO(A)

MAESTRÍA EN SISTEMAS COMPUTACIONALES

PRESENTA
DANIEL RAMOS RIVERA

ASESOR
VERÓNICA QUINTERO ROSAS

CO-ASESOR
LEONARDO TRUJILLO REYES

MEXICALI, BAJA CALIFORNIA, MÉXICO

ENERO 2025



2025
Año de
La Mujer
Indígena

Av. Tecnológico S/N Col. Elías Calles C.P. 21376, Mexicali, B.C.
Tel. 686 580 49 80 al 84
e-mail direccion@itmexicali.edu.mx tecnm.mx | itmexicali.edu.mx





Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Mexicali
Subdirección académica

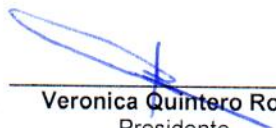
Mexicali, B.C., **10/enero/2025**
Oficio DEPI No. 02/2025

Ramos Rivera Daniel
PRESENTE

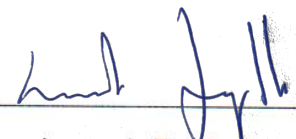
El que suscribe, Jefe de la División de Estudios de Posgrado e Investigación, comunica a usted que para la obtención del grado de Maestría en Sistemas Computacionales, se ha autorizado la reproducción de su trabajo de tesis, cuyo título es:

"Aprendizaje máquina para la clasificación de señales de series de tiempo"

La autorización se emite con base a la revisión y dictamen emitido favorablemente y avalado con su rúbrica por los integrantes del Comité Tutorial, integrado por:


Veronica Quintero Rosas
Presidente


Mario Alberto Camarillo Ramos
Secretario

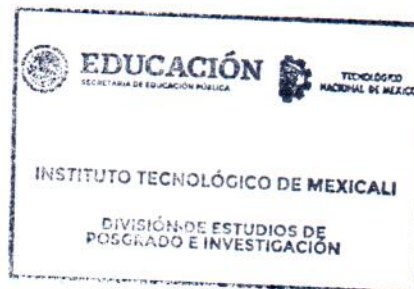

Leonardo Trujillo Reyes
Vocal

ATENTAMENTE

*Excelencia en Educación Tecnológica
La Tecnología para el Bien de la Humanidad*


Mario A. Camarillo Ramos
Jefe de la División de Estudios de Posgrado e Investigación

C.c.p. Departamento de Servicios Escolares
C.c.p. Subdirección Académica
C.c.p. Integrantes del Comité Tutorial
C.c.p. Archivo.-



Resumen

Las series de tiempo son fundamentales para analizar fenómenos dinámicos debido a su naturaleza interdependiente, lo que dificulta el uso de métodos tradicionales de clasificación. El aprendizaje automático resulta esencial para identificar patrones complejos en estos datos, mejorando la precisión y eficiencia de las tareas de análisis (Bagnall y cols., 2019). Este trabajo compara dos enfoques para la clasificación de series temporales en dos casos de estudio: la detección de patrones de deambulación asociados a la demencia y la identificación de daños en las palas de turbinas eólicas. El primer enfoque combina la extracción de características mediante la Transformada Discreta de Wavelet (DWT, por sus siglas en inglés) *Discrete Wavelet Transform* con modelos clásicos de aprendizaje automático (*Random Forest*, Máquinas de Soporte Vectorial, k-Vecinos más Cercanos). El segundo utiliza modelos especializados para series temporales en crudo, como ROCKET y HIVE-COTE2. Los resultados muestran que el enfoque basado en características extraídas mediante DWT supera a los modelos especializados como en el primer caso. Para la detección de patrones de deambulación, el modelo *Random Forest* combinado con características extraídas logró el mejor rendimiento. En el caso de los daños en turbinas eólicas, las Máquinas de Soporte Vectorial con las series temporales en crudo demostraron una mayor precisión. Estos hallazgos destacan la eficiencia y robustez de los

ii

modelos clásicos combinados con técnicas de extracción de características frente a métodos más complejos en problemas específicos de clasificación de series temporales.

Abstract

This work compares two approaches for time series classification in two case studies: detecting wandering patterns associated with dementia and identifying damage in wind turbine blades. The first approach combines feature extraction using the Discrete Wavelet Transform (DWT) with classic machine learning models (Random Forest, Support Vector Machines, k-Nearest Neighbors). The second approach employs specialized models for raw time series, such as ROCKET and HIVE-COTE2. The results show that the feature-based approach using DWT consistently outperforms the specialized models in one of the studie cases. For wandering pattern detection, the Random Forest model combined with extracted features achieved the best performance. In the wind turbine damage classification case, Support Vector Machines with raw time series demonstrated higher accuracy. These findings highlight the efficiency and robustness of classic models combined with feature extraction techniques compared to more complex methods in specific time series classification problems.

Dedicatoria

A mi familia, quienes me brindaron la base y el apoyo necesarios para estudiar una carrera universitaria; a mis amigos, que siempre me impulsaron a seguir adelante; a Astrid, quien fue testigo del inicio de esta aventura; al Doctor Arnoldo Díaz (q.e.p.d.), quien me abrió las puertas al mundo de la investigación; y a Canelito, que me acompañó en la última vuelta y hasta el último capítulo de este viaje.

Agradecimientos

Quiero expresar mi más profundo agradecimiento al Doctor Arnoldo Díaz, quien, mientras estuvo entre nosotros, nunca dejó de apoyar el desarrollo de la ciencia y cuya huella permanece en cada uno de nosotros. Mi gratitud infinita también para la profesora Verónica Quintero, quien, de principio a fin, veló por nuestros intereses y no descansó hasta vernos graduados.

Extiendo también mi agradecimiento al Doctor Leonardo Trujillo, de quien aprendí valiosas lecciones y con quien me siento honrado de haber trabajado y recibido su asesoramiento. Finalmente, a mi familia, a quienes debo mi gratitud desde mi nacimiento y hasta el último de mis días. Muchas gracias por su apoyo incondicional.

Índice general

1. Introducción	1
1.1. Importancia del análisis de series de tiempo	3
1.1.1. Detección de patrones de deambulaci3n	3
1.1.2. Detecci3n de la erosi3n en palas de aerogeneradores	5
1.2. Relevancia de los modelos de aprendizaje autom3tico en diversas aplicaciones	7
1.2.1. Interpolaci3n	8
1.2.2. Pron3stico	9
1.2.3. Clasificaci3n	10
1.3. Justificaci3n del Uso de Modelos de ML para Clasificaci3n de Series de Tiempo	11
1.3.1. Transformaci3n de series de tiempo en vectores de rasgos descriptivos	11
1.3.2. Modelos de aprendizaje de tiempo especializados para el uso de series de tiempo	12
1.4. Pregunta de investigaci3n	12
1.5. Objetivo General	13
1.6. Objetivos Específicos	13
1.7. Organizaci3n de la Tesis	14

2.1. ENFOQUE BASADO EN EXTRACCIÓN DE RASGOS	vii
1.7.1. Productos obtenidos	14
2. Trabajos Relacionados	16
2.1. Enfoque basado en extracción de rasgos para la clasificación de series de tiempo	17
2.1.1. Extracción de rasgos	17
2.1.2. Métodos de aprendizaje maquina	18
2.1.3. Optimización de hiperparámetros	19
2.2. Enfoques que analiza series de tiempo de manera directa	20
2.2.1. BOSS: Bag of SFA Symbols	20
2.2.2. KNN DTW: K-Nearest Neighbors Dynamic Time Warping	22
2.2.3. KNN DDTW: K-Nearest Neighbors Derivative Dynamic Time Warping	23
2.2.4. Shapelet	24
2.2.5. TSF: Time Series Forest	25
2.2.6. ROCKET	25
2.2.7. Mini-ROCKET: Mini RandOm Convolutional KErnel Transform	26
2.2.8. HIVECOTE2	27
2.3. Series de tiempo en la Detección de Demencia	28
2.4. Erosión del Borde de Ataque en Turbinas Eólica	31
3. Caso de Estudio 1: Detección de patrones de deambulaci3n	34
3.1. Metodología	35
3.1.1. Dataset	36
3.1.2. Imputaci3n de Datos	36
3.1.3. Extracci3n de Características	38
3.1.4. Configuraci3n Experimental	40

2.1. ENFOQUE BASADO EN EXTRACCIÓN DE RASGOS	viii
3.2. Resultados	42
3.3. Discusión	46
4. Caso de Estudio 2: Detección de Erosión y daños en palas de turbinas eólicas	49
4.1. Metodología	50
4.1.1. Dataset	51
4.1.2. Extracción de Características	54
4.2. Resultados	55
4.2.1. Problema A: Sana-Erosionada	56
4.2.2. Problema B: Sana-Todas las rugosidades	57
4.2.3. Problema C: Sana - Rugosidad $100\mu03$ y $100\mu09$	59
4.2.4. Problema D: Sana - Rugosidad $100\mu03$ y $140\mu03$	60
4.2.5. Problema E: Limpio-Erosionada-Rugosidades	61
4.2.6. Problema F: Problema multiclase completo	62
4.3. Discusión	64
5. Conclusiones Generales	67

Índice de figuras

1.1. Partes principales de un aerogenerador. Imagen de (Furlanetto y cols., 2020).	6
1.2. Ejemplos de erosión en el borde de ataque. Imagen de (Verma y cols., 2020).	7
1.3. La gráfica fue generada a partir de 10 puntos azules obtenidos de la función coseno. Para realizar la interpolación cúbica, se utilizó la función <code>interp1d</code> de <code>scipy</code> con el parámetro <code>cubic</code> , generando la curva suave representada por la línea roja.	8
1.4. La gráfica fue generada utilizando datos sintéticos de una serie temporal con tendencia lineal y ruido aleatorio, representados por la etiqueta Datos históricos. Se entrenó un modelo de regresión lineal de tipo <i>Ridge</i> para ajustar los datos históricos y predecir 10 valores futuros.	9

21

ÍNDICE DE FIGURAS

x

- 1.5. La gráfica fue generada utilizando datos sintéticos de clasificación binaria con dos características. Los datos se dividieron en conjuntos de entrenamiento y prueba, y se entrenó un modelo de Regresión Logística para clasificar los puntos. A partir del modelo, se generó una frontera de decisión al predecir en una malla de valores dentro del rango de las características, permitiendo visualizar las regiones de clasificación. 10
- 2.1. La figura reobsebt el procesamiento de el algorimot BOSS. Donde la señal sintética de series temporales (a) utilizando un enfoque de ventana deslizante (b), seguido de la aplicación de la DFT a cada ventana para extraer los componentes de frecuencia (c). La señal fue luego convertida en representaciones simbólicas basadas en la magnitud de sus componentes de frecuencia, creando un modelo de bolsa de palabras (d) donde cada valor de frecuencia es asignado a un símbolo ('A', 'B', 'C' o 'D'). 21
- 2.2. La figura ilustra la aplicación del algoritmo de DTW en dos series temporales. La serie azul representa el patrón original, mientras que la serie roja es una versión desplazada de esta. Las líneas verdes conectan los puntos correspondientes entre ambas series, visualizando el camino óptimo encontrado por DTW para alinearlas. Este camino minimiza la distancia total entre las series, demostrando la capacidad de DTW para sincronizar patrones temporales incluso con desplazamientos en el tiempo. 22

ÍNDICE DE FIGURAS

xi

- 2.3. Las series se construyeron utilizando una función seno: la primera como referencia y la segunda con un ligero desplazamiento temporal. Para calcular la alineación entre ambas, se derivaron las series y se utilizó la distancia DDTW para medir la similitud considerando las tendencias locales. La gráfica muestra ambas series, una en azul y otra en rojo, junto con las líneas verdes que representan las correspondencias o alineaciones establecidas por el algoritmo a lo largo del tiempo. 23
- 2.4. Se crearon dos series temporales sintéticas: una representando una onda sinusoidal (Clase 1) y otra una onda sinusoidal con un pico desplazado (Clase 2). Se extrajeron *shapelets*, de cada clase: un patrón de onda sinusoidal pequeña para la Clase 1 y una versión ligeramente desplazada para la Clase 2. El gráfico destaca cómo las *shapelets* capturan patrones distintivos entre clases, ayudando en la clasificación de series temporales. 24
- 2.5. La imagen visualiza una muestra de un conjunto de datos sintéticos, compuesto por 100 muestras con 50 puntos de tiempo cada una. Una serie temporal fue seleccionada como ejemplo para calcular y visualizar métricas clave como el promedio (línea roja discontinua), la desviación estándar (líneas verdes que indican los límites superior e inferior) y la pendiente derivada de una regresión lineal (línea naranja punteada). Estas métricas ofrecen una representación del comportamiento de la serie, que es fundamental para TSF. Este modelo aprovecha características estadísticas similares a nivel local y global para clasificar series temporales. . . 25

60

ÍNDICE DE FIGURAS

xii

- 2.6. Se creó una serie temporal sintética basada en una función seno, y luego se aplicaron 20 kernels de convolución aleatorios, cada uno de tamaño 5, generados mediante valores aleatorios. La gráfica muestra la serie original en azul y las series transformadas, resultantes de las convoluciones, en diferentes colores. Este proceso ejemplifica cómo ROCKET utiliza kernels aleatorios para extraer características relevantes de series temporales. 26
- 2.7. Se creó una serie temporal sintética basada en una función seno, representada en azul, y se le aplicaron 5 kernels aleatorios de tamaño 5, generados mediante valores aleatorios. Cada transformación resultante de la convolución se muestra como líneas adicionales en la gráfica. Este enfoque ilustra cómo Mini-ROCKET extrae características relevantes de series temporales al emplear un número reducido de kernels, manteniendo la esencia del método ROCKET completo pero con menos computación. 27
- 2.8. Figura del artículo (Middlehurst y cols., 2021b). Representa el modelo para predecir la clase de un caso de prueba desconocido. El modelo consta de cuatro nodos: STC, TDE, Arsenal y DrCIF, cada uno representando una fuente diferente de evidencia o característica. Cada nodo proporciona probabilidades para tres posibles clases (Clase 1, Clase 2 y Clase 3). Las probabilidades de cada nodo se combinan utilizando un promedio ponderado, con pesos determinados por el parámetro alfa (establecido en 4 en este caso). La predicción final se basa en la clase con la probabilidad más alta después de combinar la evidencia de los cuatro nodos. El modelo predice la Clase 2 para el caso de prueba desconocido. 28

ÍNDICE DE FIGURAS

xiii

3.1. Patrones de deambulaci3n: a) directo, b) pacing, c) lapping d) random.	35
3.2. Pasos por patr3n de deambulaci3n en el conjunto de datos. . . .	37
3.3. Ejemplos de series temporales para una muestra de cada clase: (a) Direct; (b) Lapping; (c) Pacing; and (d) Random.	39
3.4. Proyecciones bidimensionales de los datos de la serie temporal Slope, considerando: (a) la serie cruda y (b) la extracci3n de características con DWT. Los gráfcos se generan utilizando el método t-SNE.	43
3.5. Gráfcos de matrices de confusi3n de los puntajes de rendimien- to para las mejores configuraciones experimentales de los cuatro grupos de métodos resumidos en la Tabla 3.2. En cada caso se muestran la Precisi3n, F1, Precisi3n media, Recuperaci3n y AUC ROC calculados en los conjuntos de prueba.	44
4.1. Evoluci3n de los coeficientes de presi3n (cp) en funci3n del tiem- po para diferentes clases de superficie, mostrando las variaciones aerodinámicas asociadas con cada categoría.	53

Índice de cuadros

3.1. Valores de hiperparámetros utilizados para cada clasificador, mostrando los valores predeterminados y el rango de búsqueda utilizado por BO.	43
3.2. Promedios de cada medida de rendimiento para cada condición experimental, mostrando: accuracy, F1 score, precision, recall y ROC-AUC. Los mejores resultados para cada grupo de resultados están indicados en negrita.	47
4.1. El número de señales corresponde al número de muestras tomadas para 4 números diferentes de Reynolds y una media de 21 ángulos de ataque diferentes. Se obtuvieron 21 rasgos estadísticos para la clasificación.	52
4.2. Resultados obtenidos para el problema A, un problema binario y altamente equilibrado que utiliza las clases limpias y erosionadas.	56
4.3. Resultados obtenidos para el problema B.	58
4.4. Resultados obtenidos para el problema C.	60
4.5. Resultados obtenidos para el problema D.	61
4.6. Resultados obtenidos para el problema E.	63
4.7. Resultados obtenidos para el problema F.	64

Capítulo 1

Introducción

En un mundo donde la generación de datos crece de manera exponencial, el análisis de series temporales se ha convertido en una herramienta fundamental para extraer información valiosa en múltiples campos de aplicación. Desde la detección de patrones de comportamiento en la salud hasta el monitoreo de la integridad estructural en las energías renovables, estas técnicas ofrecen una perspectiva única para comprender fenómenos complejos y dinámicos.

Este trabajo aborda dos casos de estudio que demuestran la versatilidad del análisis de series temporales en problemas prácticos. El primer caso se enfoca en la detección de patrones de deambulación, una actividad que puede ser indicativa de enfermedades neurodegenerativas como la demencia (Martino-Saltzman y cols., 1991). Basándonos en datos obtenidos mediante sensores inerciales, se desarrollaron metodologías que combinan técnicas de extracción de características con modelos clásicos de aprendizaje automático. El segundo caso de estudio analiza la detección de erosión y daños en las palas de turbinas eólicas, un factor crítico para garantizar la eficiencia en su funcionamiento (Herring y cols., 2019). En este caso, se emplearon series temporales basadas en el coeficiente de pre-

sión para clasificar distintas condiciones de daño bajo diversas configuraciones experimentales.

Ambos estudios adoptan enfoques complementarios: el primero utiliza características derivadas mediante transformaciones como la Transformada Discreta de Wavelet, mientras que el segundo explora técnicas especializadas que procesan directamente las series temporales en crudo. Además, se evaluaron múltiples modelos de aprendizaje automático, optimizando hiperparámetros mediante estrategias avanzadas como la búsqueda Bayesiana.

Los resultados de este trabajo destacan la importancia de seleccionar metodologías adecuadas según las características del problema y el dominio de aplicación, al tiempo que subrayan las limitaciones y desafíos de trabajar con datos controlados en comparación con escenarios del mundo real. Este esfuerzo no solo contribuye al estado-del-arte en la clasificación de series temporales, sino que también sienta las bases para futuras investigaciones que busquen integrar estas soluciones en sistemas prácticos y escalables.

Los resultados de este trabajo destacan la importancia de seleccionar metodologías adecuadas según las características del problema y el dominio de aplicación, subrayando que, aunque actualmente existe un auge por adoptar técnicas avanzadas y sofisticadas como las de deep learning (Ismail Fawaz y cols., 2019), en la práctica, es común que técnicas relativamente sencillas y eficientes ofrezcan el mejor rendimiento en problemas de clasificación de patrones (Bagnall y cols., 2019). Este análisis no solo contribuye al estado-del-arte en la clasificación de series temporales, sino que también resalta las limitaciones y desafíos de trabajar con datos controlados en comparación con escenarios del mundo real. Además, sienta las bases para futuras investigaciones que busquen integrar estas soluciones en sistemas prácticos y escalables.

1.1. IMPORTANCIA DEL ANÁLISIS DE SERIES DE TIEMPO

3

1.1. Importancia del análisis de series de tiempo

En un mundo digital como el nuestro, la producción de grandes volúmenes de datos es una realidad innegable (Clissa y cols., 2023). Este rápido crecimiento se ha gestado desde la invención de las computadoras y se ve impulsado por los avances en dispositivos móviles, sensores digitales, comunicaciones, así como en computación y almacenamiento en la nube (Yaqoob y cols., 2016). En este contexto, el análisis de series de tiempo se convierte en una herramienta esencial para extraer información significativa de estos datos.

Una serie de tiempo representa un conjunto de valores obtenidos de mediciones secuenciales a lo largo del tiempo (Li y cols., 2017). Actualmente, el análisis de series de tiempo abarca una amplia gama de aplicaciones, incluyendo la economía, la predicción del clima, la evaluación de precios en la bolsa de valores y el desarrollo de negocios, entre otros (Alsharef y cols., 2022). Estos usos subrayan la relevancia del análisis temporal en la toma de decisiones informadas.

A pesar de que el análisis de series de tiempo ha sido objeto de estudio en el pasado, la convergencia de grandes volúmenes de datos y la complejidad de adaptar modelos existentes para fenómenos como la temperatura, la velocidad del aire, o el tráfico de red representan desafíos interesantes y cruciales. La capacidad de modelar y predecir estos comportamientos temporales no solo mejora nuestra comprensión de sistemas complejos, sino que también impulsa innovaciones en diversas industrias, desde la meteorología hasta el análisis financiero (Alsharef y cols., 2022).

1.1.1. Detección de patrones de deambulaci3n

La deambulaci3n es un sntoma com3n entre pacientes que sufren de demencia, con estudios que reportan una prevalencia del 67 % (Wan y cols., 2015). Los pacientes con demencia experimentan una p3rdida neuronal significativa y

1.1. IMPORTANCIA DEL ANÁLISIS DE SERIES DE TIEMPO

4

enfrentan dificultades que no son comunes en personas de su rango de edad, lo que provoca síntomas como pérdida de memoria, confusión, dificultad para articular pensamientos y desorientación (National Institute on Aging, 2022). Según el *National Institutes of Health* de los Estados Unidos, la demencia afecta a una parte considerable de la población, especialmente a los adultos mayores; cerca de un tercio de los individuos de 85 años o más experimentan sus efectos (National Institute on Aging, 2022). Es importante destacar que la demencia no es parte del envejecimiento normal, y aunque no existe cura, se pueden implementar medidas para mejorar la calidad de vida de los pacientes y sus seres queridos. Esto implica el monitoreo de las habilidades actuales del individuo y sus cambios a lo largo del tiempo, lo cual puede revelar señales de los primeros estadios de demencia.

En el estudio de Matizo-Saltzman et al. (Martino-Saltzman y cols., 1991), se encontró una correlación entre los patrones de deambulación y la demencia, corroborada por investigaciones posteriores (Nakaoka y cols., 2010). Estos patrones pueden clasificarse en cuatro categorías: directo, pacing, lapping y aleatorio; como se ilustra en la Figura 3.1. El patrón directo se refiere a un movimiento lineal entre dos puntos sin desviaciones. El patrón de pacing describe un movimiento de ida y vuelta en un área específica. Lapping implica un movimiento semicircular, a menudo repetitivo, mientras que el patrón aleatorio refleja un desplazamiento del paciente sin una dirección concreta hacia diferentes puntos dentro de un área.

Los autores encontraron que los niveles más altos de demencia estaban asociados con los patrones de lapping, seguidos de los patrones aleatorios. En contraste, aquellos que exhibían patrones de pacing mostraron niveles más bajos de demencia, y el patrón directo no mostró correlación con la demencia en ningún nivel. Esta información sugiere que el análisis de los patrones de deambulación

79

1.1. IMPORTANCIA DEL ANÁLISIS DE SERIES DE TIEMPO

5

puede ser un indicador valioso en la detección temprana de la demencia y en la evaluación de la progresión de la enfermedad.

1.1.2. Detección de la erosión en palas de aerogeneradores

La industria de las energías renovables, y en particular la generación de energía eólica, ha experimentado un crecimiento rápido en las últimas dos décadas debido a su papel como líder en la producción de energía limpia y sostenible (Herring y cols., 2019). En el último informe del *Global Wind Energy Council*, se reportó un total de instalaciones de 117 GW en 2023, lo que representa un incremento del 50 % en comparación con 2022. Actualmente, 54 países están involucrados en la creación de instalaciones para proporcionar energía eólica, con una proyección de alcanzar 2.75 TW para 2030 (Global Wind Energy Council, 2024).

Las palas (o *blades* por su traducción en inglés) de los aerogeneradores (véase figura 1.1) son componentes críticos de los sistemas de generación de energía eólica, ya que, operando en condiciones adversas, transforman la energía del viento en energía eléctrica (Du y cols., 2020). Estas condiciones pueden incluir vientos intensos, lluvias fuertes, rayos, formación de hielo, baja resistencia de los materiales, cargas de fatiga y errores humanos en su fabricación e instalación (Zárate y cols., 2022).

Entre los diversos tipos de daños que pueden ocurrir en las palas, la erosión ha cobrado una relevancia creciente, ya que el rendimiento de una turbina depende en gran medida de las propiedades aerodinámicas de las palas. La erosión del borde de ataque es causada por partículas abrasivas transportadas por el viento que impactan directamente en esta área (Zárate y cols., 2022). Este fenómeno puede provocar la remoción de material de la superficie de la pala como se observa en la imagen 1.2, dejando un perfil rugoso que degrada su ren-

1.1. IMPORTANCIA DEL ANÁLISIS DE SERIES DE TIEMPO

6

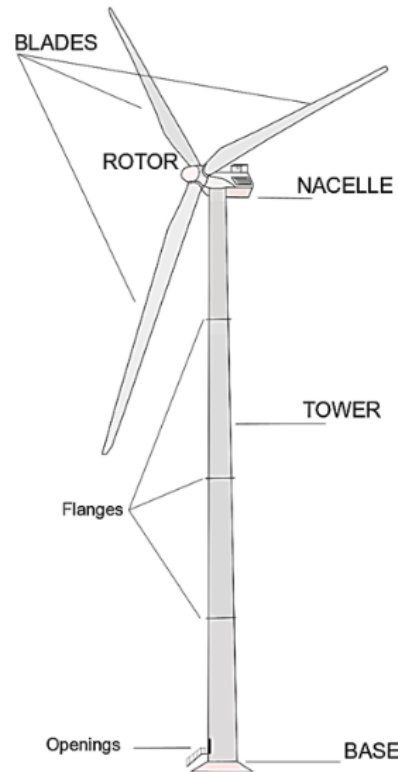


Figura 1.1: Partes principales de un aerogenerador. Imagen de (Furlanetto y cols., 2020).

dimiento aerodinámico e impacta la integridad estructural de la pala (Herring y cols., 2019).

El rendimiento aerodinámico de las palas está estrechamente relacionado con su perfil aerodinámico, que describe la forma de la sección transversal de la pala y determina su capacidad para generar sustentación y reducir la resistencia. Una pala bien diseñada mantiene un perfil aerodinámico óptimo que maximiza la eficiencia del flujo de aire. Este comportamiento está influenciado por el número de Reynolds, un parámetro adimensional que caracteriza el régimen de flujo alrededor de las palas, ya sea laminar o turbulento, y que depende de la velocidad del viento, la densidad del aire y las dimensiones de las palas (Batchelor, 2000).

127

1.2. RELEVANCIA DE LOS MODELOS DE ML

7

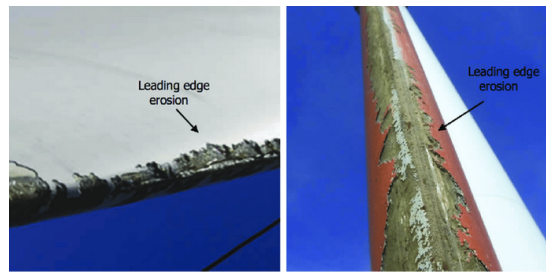


Figura 1.2: Ejemplos de erosión en el borde de ataque. Imagen de (Verma y cols., 2020).

Además, los coeficientes de presión, que cuantifican las variaciones de presión en la superficie de la pala, son fundamentales para evaluar los efectos de la erosión. La erosión altera estos coeficientes al modificar la distribución de presión en la superficie de la pala, lo que puede conducir a una disminución significativa en la eficiencia aerodinámica y a una mayor generación de turbulencias (Anderson, 2016).

1.2. Relevancia de los modelos de aprendizaje automático en diversas aplicaciones

El aprendizaje automático (ML, por sus siglas en inglés de *Machine Learning*) es la ciencia de programar computadoras para que puedan aprender a partir de datos de manera autónoma (Géron, 2019). Esta tecnología se ha vuelto fundamental en múltiples sectores, transformando radicalmente la forma en que se procesa la información y se toman decisiones. Su relevancia radica en la capacidad de analizar grandes volúmenes de datos, identificar patrones complejos y tomar decisiones con mínima intervención humana.

Existen distintos tipos de modelos de ML que se emplean en tareas específicas, tales como la interpolación, el pronóstico y la clasificación.

1.2. RELEVANCIA DE LOS MODELOS DE ML

8

1.2.1. Interpolación

En el contexto del ML, la interpolación se refiere a un método de estimación de valores desconocidos a partir de datos conocidos (Oh y cols., 2020). Los modelos comúnmente utilizados para este proceso incluyen los modelos polinomiales y los de procesos gaussianos (GP, por sus siglas en inglés de *Gaussian process*), que resultan especialmente útiles en aplicaciones donde es crucial estimar valores intermedios con precisión. Por ejemplo, en el análisis de series temporales, la interpolación se aplica para completar valores faltantes, lo cual preserva la continuidad temporal y garantiza la integridad de los datos (Shukla y Marlin, 2019). En la imagen 1.3, se muestra un ejemplo donde tenemos 10 puntos que representan una función la cual se muestran con los puntos azules y posteriormente se interpolaron 100 puntos adicionales en el mismo rango para producir una curva suave. La línea roja muestra el resultado de la interpolación.

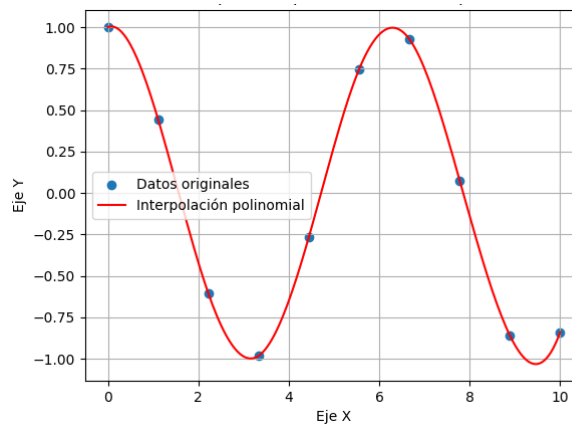


Figura 1.3: La gráfica fue generada a partir de 10 puntos azules obtenidos de la función coseno. Para realizar la interpolación cúbica, se utilizó la función `interp1d` de `scipy` con el parámetro `cubic`, generando la curva suave representada por la línea roja.

1.2. RELEVANCIA DE LOS MODELOS DE ML

9

1.2.2. Pronóstico

Los modelos de pronóstico están diseñados para predecir valores futuros basándose en datos históricos. Entre las técnicas más reconocidas se encuentran ARIMA (por sus siglas en inglés de *Auto-Regressive Integrated Moving Average*), redes de memoria a largo y corto plazo (o LSTM, por sus siglas en inglés de *Long short-term memory*) y arquitecturas de aprendizaje profundo más recientes, como los *transformers* y las redes neuronales de grafos, que han demostrado una mayor eficacia en el pronóstico de series temporales complejas (Casolaro y cols., 2023). Estos modelos son ampliamente utilizados en aplicaciones diversas, que van desde la predicción del mercado de valores hasta el pronóstico meteorológico.

La gráfica 1.4 es un ejemplo de pronóstico utilizando datos históricos sintéticos con una tendencia lineal y ruido aleatorio, representando el comportamiento de una serie temporal en el tiempo. Un modelo de regresión *Ridge*, una variante de regresión lineal que incluye regularización, fue entrenado con estos datos históricos para aprender la tendencia subyacente. A partir de este modelo, se predijeron valores futuros en un horizonte de 10 valores adicionales.

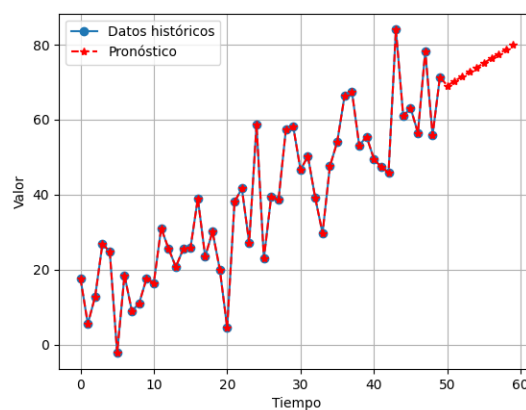


Figura 1.4: La gráfica fue generada utilizando datos sintéticos de una serie temporal con tendencia lineal y ruido aleatorio, representados por la etiqueta Datos históricos. Se entrenó un modelo de regresión lineal de tipo *Ridge* para ajustar los datos históricos y predecir 10 valores futuros.

1.2. RELEVANCIA DE LOS MODELOS DE ML

10

1.2.3. Clasificación

La clasificación es una de las aplicaciones más comunes de ML, en la cual los modelos asignan categorías predefinidas a **datos de entrada**. Ejemplos de algoritmos de clasificación incluyen Máquinas de Vectores de Soporte (SVM, por sus siglas en inglés de *Support vector machine*), árboles de decisión y redes neuronales convolucionales. Estos modelos **son ampliamente utilizados en tareas** como el **reconocimiento de imágenes** y el diagnóstico médico, donde la precisión y la capacidad para identificar patrones complejos son cruciales (Goodfellow y cols., 2016). En la figura 1.5 tenemos un ejemplo muestra cómo un modelo aprende a distinguir entre clases a partir de datos etiquetados y utiliza esta información para asignar nuevas observaciones a una clase específica. La frontera de decisión refleja **la capacidad del modelo para generalizar** los patrones aprendidos **y realizar predicciones en datos no vistos**.

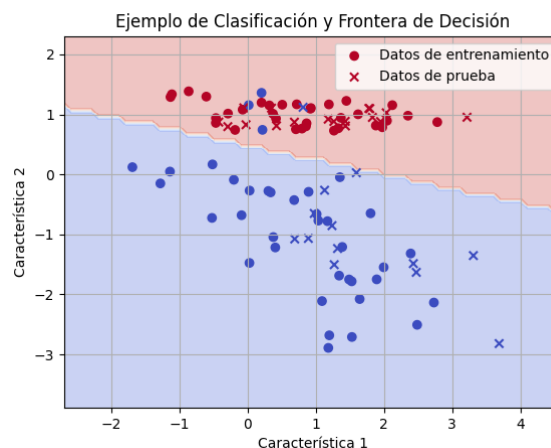


Figura 1.5: La gráfica fue generada utilizando datos sintéticos de clasificación binaria con dos características. Los datos se dividieron en conjuntos de entrenamiento y prueba, y se entrenó un modelo de Regresión Logística para clasificar los puntos. A partir del modelo, se generó una frontera de decisión al predecir en una malla de valores dentro del rango de las características, permitiendo visualizar las regiones de clasificación.

1.3. JUSTIFICACIÓN DEL USO DE MODELOS DE ML

11

1.3. Justificación del Uso de Modelos de ML para Clasificación de Series de Tiempo

La clasificación de series de tiempo es un subcampo de ML cuya relevancia se ha incrementado significativamente, como lo menciona Faouzi (Faouzi, 2022). Este crecimiento está directamente relacionado con el aumento en el volumen de datos recolectados a través de sensores y el tráfico generado por actividades en internet. Debido a la naturaleza no estructurada de las series de tiempo y su impacto en numerosos campos de aplicación, su estudio ha cobrado una importancia crucial en los últimos años.

Aunque la tarea de clasificación es ampliamente conocida en el ámbito del ML, muchos de los algoritmos clásicos están diseñados para trabajar con datos estructurados, lo cual no es el caso de las series de tiempo. Por ejemplo, ciertos modelos no consideran la correlación inherente en el orden temporal de los valores de las series (Faouzi, 2022). Para abordar este desafío, han surgido dos enfoques principales:

1.3.1. Transformación de series de tiempo en vectores de rasgos descriptivos

Este enfoque aprovecha modelos tradicionales de ML, ampliamente estudiados por su robustez y eficacia, combinándolos con un proceso de extracción de características previo. Este proceso consiste en convertir las series de tiempo en conjuntos de características descriptivas, lo que permite utilizar estos modelos en datos temporales. Algunas características comunes incluyen medidas estadísticas (media, varianza), transformaciones como Fourier o Wavelet, o incluso características específicas del dominio de aplicación.

Este enfoque destaca por su interpretabilidad y eficiencia computacional,

1.4. PREGUNTA DE INVESTIGACIÓN

12

especialmente cuando se trabaja con conjuntos de datos pequeños. Sin embargo, su éxito depende en gran medida de la calidad del proceso de extracción de características y del conocimiento experto sobre el dominio. Bagnall et al. (Bagnall y cols., 2019) demostraron que este enfoque puede ser competitivo en escenarios bien definidos y con recursos computacionales limitados.

1.3.2. Modelos de aprendizaje de tiempo especializados para el uso de series de tiempo

En contraste, los modelos especializados, como ROCKET (Dempster y cols., 2020a), Shapelet Transform (Hills y cols., 2014) y HIVE-COTE2 (Middlehurst y cols., 2021b), están diseñados para procesar directamente series de tiempo en crudo. Estos modelos explotan las propiedades únicas de las dependencias temporales y los patrones inherentes en los datos, eliminando la necesidad de procesos de extracción de características manuales.

Estudios como el *Time-Series Classification Bake-off* (Faouzi, 2022) han demostrado que estos modelos especializados superan consistentemente a los métodos clásicos en términos de precisión. No obstante, esta mejora suele venir acompañada de un alto costo computacional, particularmente durante el entrenamiento. Esto los hace más adecuados para conjuntos de datos grandes o cuando se dispone de recursos computacionales significativos.

1.4. Pregunta de investigación

¿Cómo se comparan los algoritmos de ML basados en extracción de características con los modelos especializados en la clasificación de series de tiempo en términos de precisión, eficiencia computacional e interpretabilidad al aplicarse a casos de estudio del mundo real?

1.5. OBJETIVO GENERAL

13

1.5. Objetivo General

Caracterizar el desempeño empírico de algoritmos de ML aplicados a problemas de clasificación donde los datos se representen como series de tiempo, considerando tanto algoritmos basados en extracción de características como aquellos que procesan directamente las series.

Adicionalmente, es importante destacar que, aunque la tendencia actual en el campo del aprendizaje automático favorece el uso de técnicas más sofisticadas y computacionalmente costosas, en muchos casos las técnicas tradicionales, más sencillas y eficientes, no solo pueden ser suficientes para resolver ciertos problemas, sino también pueden superar a sus contrapartes modernas en cuanto a rendimiento y eficiencia. Esta investigación busca explorar esta posibilidad, aportando evidencia empírica que contribuya al entendimiento y a la selección de algoritmos en función del problema y los recursos disponibles.

1.6. Objetivos Específicos

- Formular dos problemas de clasificación de series de tiempo como casos de estudio: detección de erosión en palas de aerogeneradores y detección de patrones de deambulación asociados a la demencia.
- Desarrollar una metodología basada en la extracción de características mediante la Transformada Discreta Wavelet, combinada con algoritmos de clasificación y optimización de hiperparámetros.
- Diseñar una metodología para la clasificación directa de series de tiempo utilizando métodos especializados, que incluyan técnicas basadas en shapelets, intervalos, filtros de convolución y ensambles.
- Evaluar ambas metodologías en los casos de estudio mediante análisis

1.7. ORGANIZACIÓN DE LA TESIS

14

estadístico y visual, comparando su desempeño en términos de precisión, eficiencia y aplicabilidad.

1.7. Organización de la Tesis

Esta tesis está organizada en los siguientes capítulos:

- **Capítulo 1: Introducción.** Presenta el contexto, relevancia, y objetivos de la investigación, incluyendo la justificación del uso de técnicas de aprendizaje automático en la clasificación de series de tiempo.
- **Capítulo 2: Trabajos Relacionados.** Describe los enfoques actuales en la clasificación de series de tiempo, tanto los basados en extracción de características como aquellos que procesan las series de tiempo directamente.
- **Capítulo 3: Caso de Estudio 1.** Analiza la detección de patrones de deambulaciones asociados a la demencia, destacando la metodología, resultados y discusión de este caso.
- **Capítulo 4: Caso de Estudio 2.** Se enfoca en la detección de erosión y daños en las palas de aerogeneradores, presentando los métodos y hallazgos experimentales.
- **Capítulo 5: Conclusiones Generales.** Resume los hallazgos principales, las contribuciones al estado-del-arte y propone líneas de investigación futura.

1.7.1. Productos obtenidos

- Artículo titulado "*Classification of Wandering Patterns in the Elderly Using Machine Learning and Time Series Analysis*", publicado en *IEEE*

1.7. ORGANIZACIÓN DE LA TESIS

15

Latin America Transactions. Disponible en:

<https://latamt.ieee9.org/index.php/transactions/article/view/8969>.

- Ponencia titulada "Clasificación de Patrones de Deambulación Utilizando Análisis de Series de Tiempo" presentada en la *Red Internacional de Control y Cómputo Aplicados (RICCA)*. Más información en:

<https://sites.google.com/tectijuana.edu.mx/ricca2024>.

- Video *abstract* del artículo "*Classification of Wandering Patterns in the Elderly Using Machine Learning and Time Series Analysis*", disponible en YouTube:

<https://www.youtube.com/watch?v=VA0fUB7BvMs>.

Capítulo 2

Trabajos Relacionados

Este capítulo establece el contexto general de la investigación, destacando la importancia del análisis de series de tiempo en la resolución de problemas complejos. También se exploran los enfoques actuales disponibles para la clasificación de series temporales, proporcionando una perspectiva integral sobre las metodologías más utilizadas en el área. Además, se introducen los dos casos de estudio en los que se compara el desempeño de dichos enfoques: la detección de patrones de deambulación asociados a la demencia y la clasificación de daños por erosión en palas de aerogeneradores. Estos casos permiten evaluar cómo las técnicas de aprendizaje automático pueden ofrecer soluciones robustas y efectivas en aplicaciones del mundo real. Finalmente, el capítulo presenta los objetivos general y específicos de la investigación, justificando la pertinencia y relevancia del enfoque adoptado.

2.1. ENFOQUE BASADO EN EXTRACCIÓN DE RASGOS

17

2.1. Enfoque basado en extracción de rasgos para la clasificación de series de tiempo

El enfoque basado en la extracción de rasgos para la clasificación de series de tiempo se centra en transformar datos en crudo en representaciones estructuradas que capturen características relevantes para los modelos de ML. Este enfoque permite simplificar la complejidad de las series temporales al extraer información crítica que describe su comportamiento, mejorando la precisión y eficiencia de los modelos. A continuación, se describen los principales componentes de este enfoque: la extracción de características, los métodos de ML y la optimización de hiperparámetros.

2.1.1. Extracción de rasgos

La extracción de características es un paso fundamental en la clasificación de series de tiempo, ya que permite transformar datos en crudo en un formato más estructurado, adecuado para los modelos de ML. Numerosos estudios han explorado diferentes enfoques para extraer características que capturen patrones temporales y dinámicas críticas. Por ejemplo, Fulcher y Jones (Fulcher y Jones, 2014) introdujeron un marco de trabajo comprensivo que calcula sistemáticamente más de 9,000 características de series de tiempo, proporcionando un amplio espectro de estadísticas descriptivas y atributos estructurales.

Además de estos métodos estadísticos y estructurales, técnicas como la Transformada Discreta de Wavelet (DWT, por sus siglas en inglés de *Discrete Wavelet Transform*) han sido ampliamente utilizadas para la extracción de características en sistemas de ML, particularmente para capturar información en los dominios temporal y de frecuencia. La DWT es una herramienta matemática que descompone una señal en componentes de diferentes escalas de resolución, permitiendo

2.1. ENFOQUE BASADO EN EXTRACCIÓN DE RASGOS

18

representar simultáneamente detalles locales y globales de la serie temporal. Este enfoque es especialmente útil en series no estacionarias, ya que permite analizar cambios dinámicos en el tiempo y en la frecuencia (Mallat, 1999).

Por ejemplo, Mert y Akan (Mert y Akan, 2015) emplearon DWT para analizar señales EEG (electroencefalográficas, o en inglés, *electroencephalographic*) en la detección de crisis epilépticas, logrando mejoras significativas en la precisión de clasificación. En este contexto, la DWT permitió identificar patrones específicos en las señales de EEG, separando información clave de ruido y otros artefactos. De manera similar, Subasi (Subasi, 2007) aplicó la extracción de características basada en wavelets a señales biomédicas, demostrando su efectividad en la distinción de diferentes estados fisiológicos. En ambos casos, la capacidad de DWT para representar la información en dominios duales fue crucial para mejorar el rendimiento de los modelos de clasificación.

2.1.2. Métodos de aprendizaje máquina

El uso de modelos de ML ha demostrado ser altamente efectivo en la clasificación de series de tiempo, permitiendo identificar patrones complejos y dinámicas inherentes a los datos temporales. Entre los modelos más utilizados destacan el *Random Forest* (RF), el SVM y el k-Vecinos más Cercanos (kNN), debido a su capacidad para abordar una amplia gama de problemas en este dominio. RF, propuesto originalmente por Breiman (Breiman, 2001), ha sido empleado en proyectos como el trabajo de Torgo (Torgo, 2010), donde se utiliza para predecir series temporales de calidad del aire, demostrando robustez frente a datos ruidosos y desbalanceados. Por otro lado, el SVM, introducido por Cortes y Vapnik (Cortes y Vapnik, 1995), ha sido aplicado en la clasificación de señales EEG en estudios como el de Murugappan (Murugappan y cols., 2010), que utiliza SVM para diferenciar emociones humanas a partir de señales fisiológicas.

2.1. ENFOQUE BASADO EN EXTRACCIÓN DE RASGOS

19

Finalmente, el kNN, conocido por su simplicidad y efectividad, se ha utilizado en proyectos como el de Ding (Ding y cols., 2008), quienes aplicaron kNN con distancias basadas en *Dynamic Time Warping* (DTW) para clasificar actividades humanas a partir de datos de acelerómetros. Estos modelos han sido aplicados exitosamente en diversas áreas, desde predicción financiera hasta monitoreo de salud, destacándose por su versatilidad y rendimiento en tareas de clasificación de series temporales.

2.1.3. Optimización de hiperparámetros

La optimización de hiperparámetros es el proceso de encontrar la mejor combinación de valores para los parámetros ajustables de un modelo de ML. A diferencia de los parámetros que el modelo aprende durante el entrenamiento, los hiperparámetros deben definirse antes del proceso de entrenamiento y tienen un impacto directo en el rendimiento del modelo. Ejemplos de hiperparámetros incluyen la tasa de aprendizaje en redes neuronales, la profundidad máxima en árboles de decisión, o el número de vecinos en kNN. Este proceso de optimización es fundamental porque una configuración inadecuada de hiperparámetros puede resultar en un rendimiento subóptimo, incluso si los datos y el modelo son de alta calidad (Bergstra y Bengio, 2011).

Entre los métodos de optimización, las técnicas tradicionales como la búsqueda aleatoria y la búsqueda en cuadrícula son fáciles de implementar pero a menudo ineficientes, especialmente en espacios de búsqueda grandes. Para abordar estas limitaciones, se han desarrollado enfoques más avanzados como la optimización Bayesiana (BO, por sus siglas en inglés de Bayesian Optimization), que utiliza modelos probabilísticos para guiar el proceso de búsqueda (Snoek y cols., 2012).

La BO es una técnica de optimización secuencial diseñada para encontrar

2.2. ENFOQUES QUE ANALIZA SERIES DE TIEMPO DE MANERA DIRECTA²⁰

el valor óptimo de una función objetivo que es cara de evaluar y no tiene una expresión matemática explícita. A diferencia de las técnicas tradicionales, la BO construye un modelo probabilístico del espacio de búsqueda (por ejemplo, usando procesos Gaussianos o árboles de estimación de Parzen) para predecir qué configuraciones de hiperparámetros son más prometedoras (Snoek y cols., 2012; Bergstra y cols., 2013)

La BO se ha utilizado con éxito en una variedad de aplicaciones, incluyendo la optimización de hiperparámetros en modelos de predicción financiera, análisis de señales biomédicas y clasificación de series temporales. Klein (Klein y cols., 2017) aplicaron esta técnica para ajustar hiperparámetros en RF y *Gradient Boosted Machines* en grandes conjuntos de datos, logrando mejoras significativas en precisión y eficiencia computacional.

2.2. Enfoques que analiza series de tiempo de manera directa

En contraste con el enfoque basado en la extracción de características, este enfoque emplea modelos diseñados específicamente para trabajar directamente con las series de tiempo en su formato original, evitando la necesidad de una etapa previa de transformación o extracción de rasgos. Estos modelos aprovechan propiedades intrínsecas de las series temporales, como la secuencialidad, las relaciones temporales y las variaciones dinámicas, para realizar la clasificación de manera eficiente y precisa.

2.2.1. BOSS: Bag of SFA Symbols

El modelo *Bag of SFA Symbols* (BOSS, por sus siglas en inglés) se basa en el uso de la transformación *Symbolic Fourier Approximation* (SFA, por sus siglas

2.2. ANÁLISIS DE SERIES DE TIEMPO DE MANERA DIRECTA 21

en inglés), la cual permite convertir series de tiempo en secuencias de palabras simbólicas. BOSS utiliza una bolsa de palabras para representar las características temporales de las series, logrando así la preservación de la información de fase. Este método es efectivo en la clasificación de series de tiempo debido a su enfoque en subsecuencias simbólicas y su capacidad para manejar variaciones en la frecuencia de patrones (Schäfer, 2015). Los gráficos de la imagen 2.1 muestran la serie temporal original, las ventanas deslizantes, sus transformadas de Fourier y la representación simbólica en forma de bolsa de palabras.

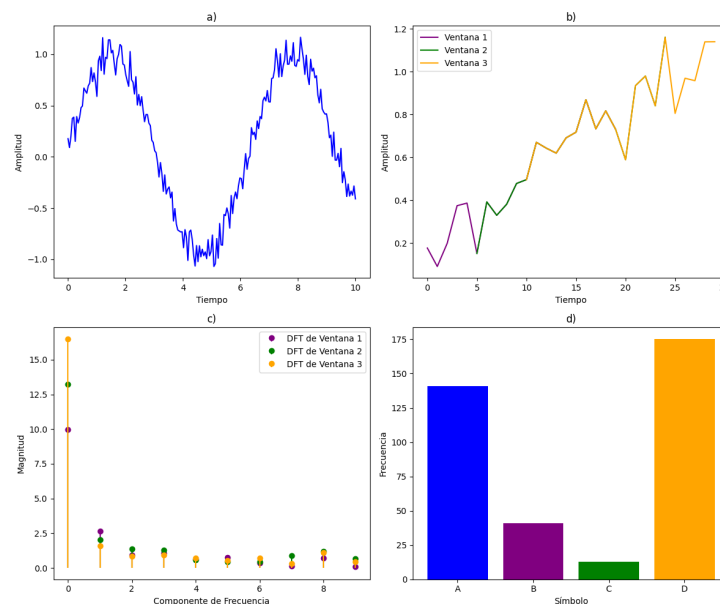


Figura 2.1: La figura reobsebt el procesamiento de el algorimot BOSS. Donde la señal sintética de series temporales (a) utilizando un enfoque de ventana deslizante (b), seguido de la aplicación de la DFT a cada ventana para extraer los componentes de frecuencia (c). La señal fue luego convertida en representaciones simbólicas basadas en la magnitud de sus componentes de frecuencia, creando un modelo de bolsa de palabras (d) donde cada valor de frecuencia es asignado a un símbolo ('A', 'B', 'C' o 'D').

2.2. ANÁLISIS DE SERIES DE TIEMPO DE MANERA DIRECTA 22

2.2.2. KNN DTW: K-Nearest Neighbors Dynamic Time Warping

El modelo kNN DTW (o *K-Nearest Neighbors Dynamic Time Warping* por sus siglas en inglés) utiliza la métrica Dynamic Time Warping (DTW, por sus siglas en inglés) para calcular la similitud entre series de tiempo. DTW permite comparar series que pueden estar desfasadas temporalmente, estirando o comprimiendo segmentos de la serie. kNN DTW es ampliamente utilizado debido a su sencillez y efectividad, especialmente en conjuntos de datos donde la alineación temporal es crítica (Berndt y Clifford, 1994). En la figura 2.2, se muestra un ejemplo del modelo.

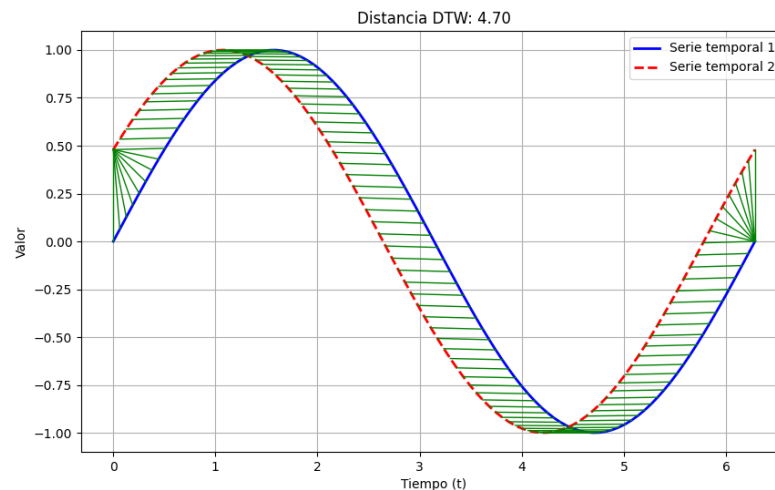


Figura 2.2: La figura ilustra la aplicación del algoritmo de DTW en dos series temporales. La serie azul representa el patrón original, mientras que la serie roja es una versión desplazada de esta. Las líneas verdes conectan los puntos correspondientes entre ambas series, visualizando el camino óptimo encontrado por DTW para alinearlas. Este camino minimiza la distancia total entre las series, demostrando la capacidad de DTW para sincronizar patrones temporales incluso con desplazamientos en el tiempo.

2.2. ANÁLISIS DE SERIES DE TIEMPO DE MANERA DIRECTA 23

2.2.3. KNN DDTW: K-Nearest Neighbors Derivative Dynamic Time Warping

El modelo kNN DDTW (o *K-Nearest Neighbors Derivative Dynamic Time Warping* por sus siglas en inglés) es una variación del DTW que usa la derivada de las series de tiempo en lugar de los valores originales. Esto es útil en casos donde la forma de la serie es importante, ya que *Dynamic Time Warping* (DDTW, por sus siglas en inglés) resalta cambios en la tendencia o pendiente en lugar de valores absolutos, proporcionando un mejor alineamiento para series con cambios de pendiente significativos (Keogh y Pazzani, 2001). Véase la figura 2.3 como ejemplo del modelo.

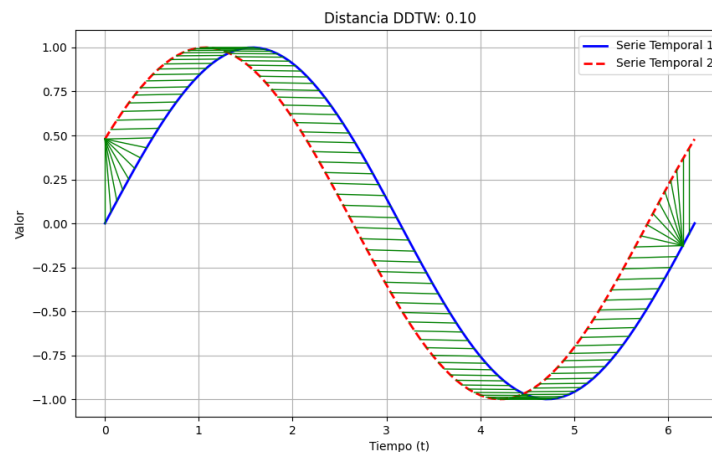


Figura 2.3: Las series se construyeron utilizando una función seno: la primera como referencia y la segunda con un ligero desplazamiento temporal. Para calcular la alineación entre ambas, se derivaron las series y se utilizó la distancia DDTW para medir la similitud considerando las tendencias locales. La gráfica muestra ambas series, una en azul y otra en rojo, junto con las líneas verdes que representan las correspondencias o alineaciones establecidas por el algoritmo a lo largo del tiempo.

2.2. ANÁLISIS DE SERIES DE TIEMPO DE MANERA DIRECTA 24

2.2.4. Shapelet

El modelo *Shapelet* se basa en la extracción de patrones cortos o *shapelets* que son subsecuencias características dentro de las series de tiempo y que permiten una clasificación precisa, basándose en métricas como la distancia mínima o la similitud entre patrones. Las *shapelets* son particularmente útiles para capturar patrones distintivos en series complejas (véase figura 2.4), reduciendo la dimensionalidad y el ruido (Ye y Keogh, 2011). Los shapelets son especialmente útiles en problemas de datos secuenciales, ya que resaltan patrones locales sin requerir información global de toda la serie.

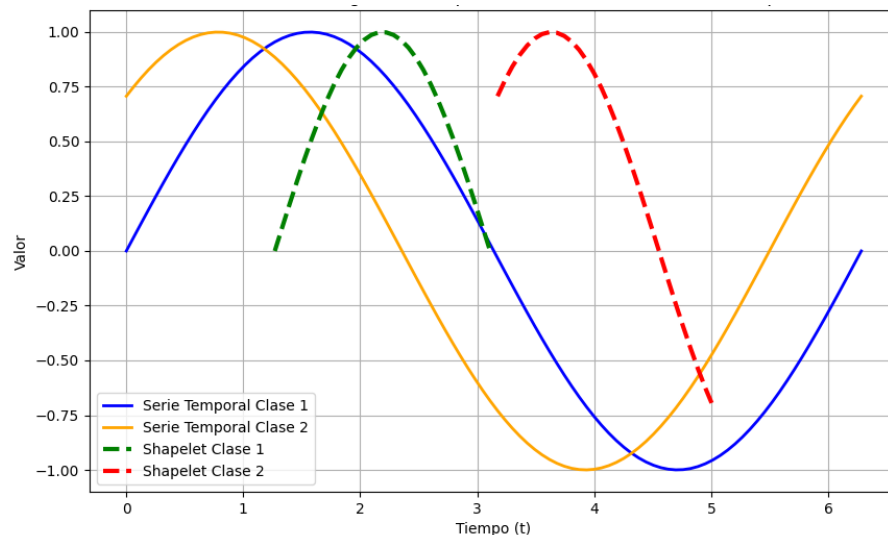


Figura 2.4: Se crearon dos series temporales sintéticas: una representando una onda sinusoidal (Clase 1) y otra una onda sinusoidal con un pico desplazado (Clase 2). Se extrajeron *shapelets*, de cada clase: un patrón de onda sinusoidal pequeña para la Clase 1 y una versión ligeramente desplazada para la Clase 2. El gráfico destaca cómo las *shapelets* capturan patrones dsitintivos entre clases, ayudando en la clasificación de series temporales.

60

2.2. ANÁLISIS DE SERIES DE TIEMPO DE MANERA DIRECTA 25

138

2.2.5. TSF: Time Series Forest

TSF (o Time Series Forest, por sus siglas en inglés) es un modelo basado en árboles de decisión, donde cada árbol utiliza intervalos de la serie para calcular estadísticos simples (media, desviación estándar, pendiente) que actúan como características (véase la figura 2.5). Esta técnica permite clasificar series de tiempo de manera eficiente, manteniendo la capacidad de interpretación y escalabilidad del modelo (Deng y cols., 2013).

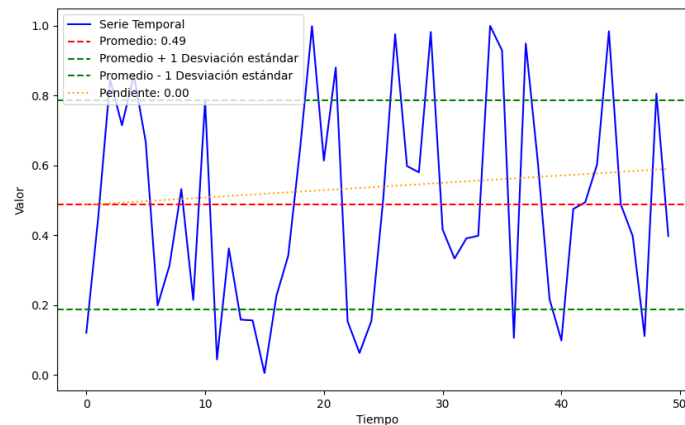


Figura 2.5: La imagen visualiza una muestra de un conjunto de datos sintéticos, compuesto por 100 muestras con 50 puntos de tiempo cada una. Una serie temporal fue seleccionada como ejemplo para calcular y visualizar métricas clave como el promedio (línea roja discontinua), la desviación estándar (líneas verdes que indican los límites superior e inferior) y la pendiente derivada de una regresión lineal (línea naranja punteada). Estas métricas ofrecen una representación del comportamiento de la serie, que es fundamental para TSF. Este modelo aprovecha características estadísticas similares a nivel local y global para clasificar series temporales.

2.2.6. ROCKET

ROCKET (o RandOm Convolutional KErnel Transform, por sus siglas en inglés) utiliza convoluciones aleatorias como una forma de transformar las series

2.2. ANÁLISIS DE SERIES DE TIEMPO DE MANERA DIRECTA 26

de tiempo en vectores de características. La clave de ROCKET es que no necesita ajustar parámetros de los kernels, lo cual lo convierte en un enfoque rápido y robusto. ROCKET ha demostrado ser eficiente para la clasificación de series de tiempo en diversas aplicaciones (Dempster y cols., 2020a). Una representación gráfica de este comportamiento del modelo se muestra en la figura 2.6.

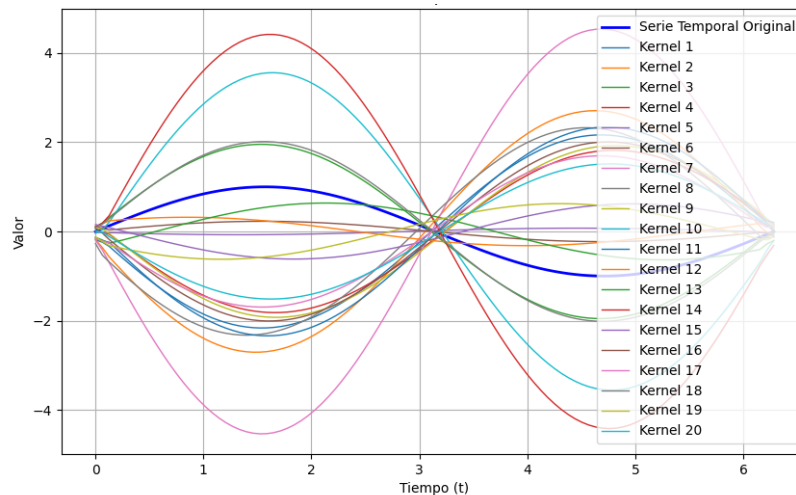


Figura 2.6: Se creó una serie temporal sintética basada en una función seno, y luego se aplicaron 20 kernels de convolución aleatorios, cada uno de tamaño 5, generados mediante valores aleatorios. La gráfica muestra la serie original en azul y las series transformadas, resultantes de las convoluciones, en diferentes colores. Este proceso ejemplifica cómo ROCKET utiliza kernels aleatorios para extraer características relevantes de series temporales.

2.2.7. Mini-ROCKET: Mini RandOm Convolutional Kernel Transform

Mini-ROCKET es una versión optimizada del modelo ROCKET que reduce el tiempo de cálculo, utilizando un subconjunto reducido de convoluciones aleatorias (véase figura 2.7). A través de esta selección, Mini-ROCKET permite una representación rápida de las series de tiempo con alto rendimiento y es adecuada para la clasificación de series a gran escala debido a su eficiencia computacional.

2.2. ANÁLISIS DE SERIES DE TIEMPO DE MANERA DIRECTA 27

(Dempster y cols., 2021).

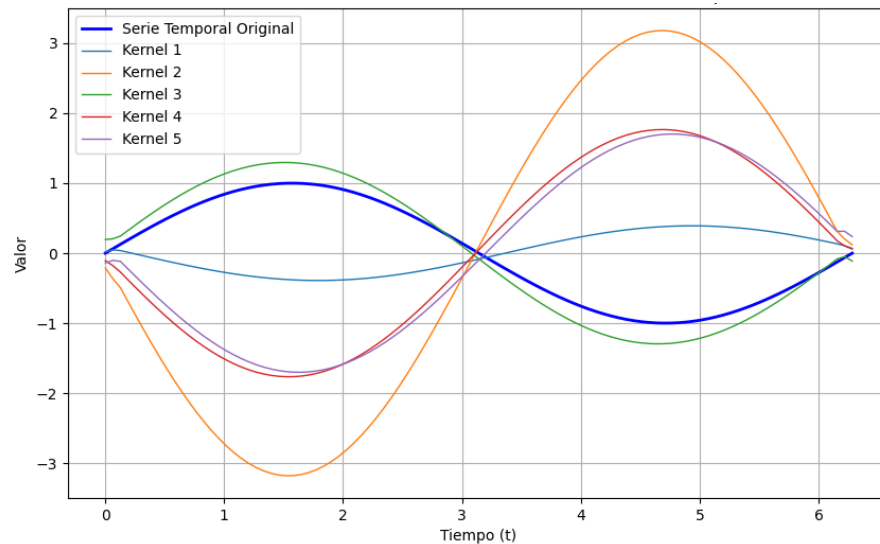


Figura 2.7: Se creó una serie temporal sintética basada en una función seno, representada en azul, y se le aplicaron 5 kernels aleatorios de tamaño 5, generados mediante valores aleatorios. Cada transformación resultante de la convolución se muestra como líneas adicionales en la gráfica. Este enfoque ilustra cómo Mini-ROCKET extrae características relevantes de series temporales al emplear un número reducido de kernels, manteniendo la esencia del método ROCKET completo pero con menos computación.

2.2.8. HIVECOTE2

HIVECOTE2 es una extensión del modelo HIVECOTE original, diseñado como un conjunto de clasificadores que combina varias transformaciones de series de tiempo y votación en múltiples niveles para mejorar la precisión de clasificación. HIVECOTE2 es un clasificador de meta-ensemble que incluye sub-ensembles especializados en diferentes representaciones y transformaciones de las series de tiempo, permitiendo una cobertura más amplia y precisa de patrones (Lines y cols., 2020). En la figura 2.8, se describe un ejemplo de cómo HIVECOTE2 funciona basado en artículo de (Middlehurst y cols., 2021b).

2.3. SERIES DE TIEMPO EN LA DETECCIÓN DE DEMENCIA

28

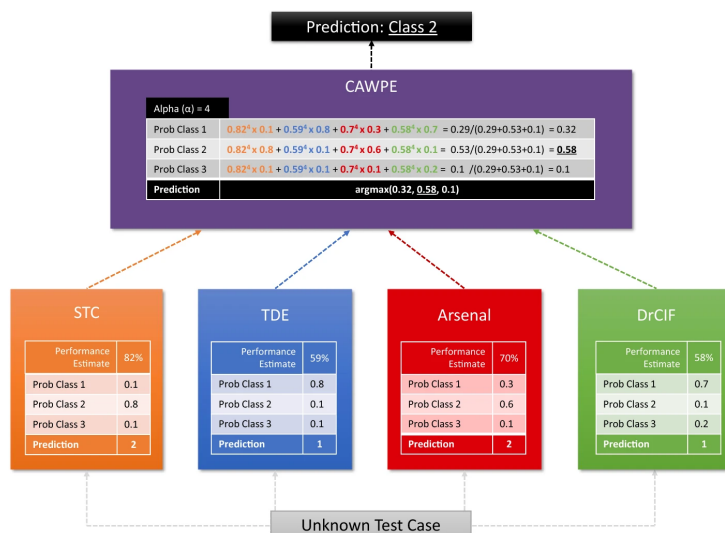


Figura 2.8: Figura del artículo (Middlehurst y cols., 2021b). Representa el modelo para predecir la clase de un caso de prueba desconocido. El modelo consta de cuatro nodos: STC, TDE, Arsenal y DrCIF, cada uno representando una fuente diferente de evidencia o característica. Cada nodo proporciona probabilidades para tres posibles clases (Clase 1, Clase 2 y Clase 3). Las probabilidades de cada nodo se combinan utilizando un promedio ponderado, con pesos determinados por el parámetro alfa (establecido en 4 en este caso). La predicción final se basa en la clase con la probabilidad más alta después de combinar la evidencia de los cuatro nodos. El modelo predice la Clase 2 para el caso de prueba desconocido.

2.3. Series de tiempo en la Detección de Demencia

En (Vuong y cols., 2014), los autores proponen un enfoque de ML y un algoritmo determinístico basado en un árbol predefinido. En el enfoque de ML, se emplean ocho algoritmos clásicos, incluidos Naive Bayes, Perceptrón Multicapa, Árboles de Decisión Poda, RF, Logitboost y *Bagging* con Árboles de Decisión Poda como clasificadores base, kNN con un vecino, y SVM. Todos los modelos de ML utilizaron las siguientes características extraídas para cada episodio de deambulación: entropía, ubicaciones repetidas, direcciones repetidas y direccio-

2.3. SERIES DE TIEMPO EN LA DETECCIÓN DE DEMENCIA 29

nes opuestas. En el enfoque determinista, los autores obtuvieron una precisión del 98.2 %.

La propuesta de Yang et al. (Yang y cols., 2018) evaluó los escenarios de deambulaci3n utilizando una t3cnica de sensado en banda S. Los patrones de deambulaci3n fueron caracterizados seg3n la informaci3n de amplitud y fase recibida, midiendo la perturbaci3n en la se1al de radio ideal. Luego, los autores utilizaron el algoritmo SVM con las variaciones de la amplitud de la informaci3n y la fase calibrada. El m3todo propuesto mostr3 una precisi3n del 90 %.

En la propuesta de (Barua y cols., 2020), los autores emplearon un sistema de localizaci3n basado en banda ultraancha (UWB, por sus siglas en ingl3s de *Ultra-wideband*) y los algoritmos SVM y kNN. Las caracter3sticas utilizadas fueron las coordenadas de los patrones de deambulaci3n en un mapa 2D amplio. Los resultados de precisi3n oscilan entre el 95 % y el 99 %. Sin embargo, los experimentos de deambulaci3n se realizaron en entornos muy controlados; por ejemplo, solo se definieron de manera expl3cita y est3tica dos trayectorias del patr3n aleatorio, lo cual no refleja un escenario real.

En (Chaudhary y cols., 2020), se usaron los flujos de eventos de sensores como caracter3sticas extra3das de los conjuntos de datos del sitio p3blico de CASAS (University, s.f.), con la ventaja de utilizar una red neuronal recurrente para manejar los datos de movimiento en bruto. En particular, se emplearon la *Long Short-Term Memory* (LSTM, por sus siglas en ingl3s) y la *Gated Recurrent Unit* (GRU, por sus siglas en ingl3s), siendo esta 3ltima la que obtuvo la mejor precisi3n con un 98.6 %. Es importante se1alar que las trayectorias de deambulaci3n fueron predefinidas mediante segmentos est3ticos. Adem3s, esta propuesta requiere el uso de infraestructura previamente instalada, es decir, sensores montados en el techo del 3rea de monitoreo.

Con el sistema basado en tel3fono m3vil y aceler3metro tri-axial y un mag-

2.3. SERIES DE TIEMPO EN LA DETECCIÓN DE DEMENCIA 30

netómetro, los autores de (Vuong y cols., 2014) capturaron datos de los cuales extrajeron características como la fuerza de aceleración a lo largo de los ejes x, y, y z, la aceleración lineal y la orientación del dispositivo. Para la evaluación, se usaron redes neuronales artificiales (ANN, por sus siglas en inglés de *artificial neural network*), árboles de decisión (DT, por sus siglas en inglés de *Decision tree*), SVM, RF, XGBoost y LSTM bidireccional (bi-LSTM, por sus siglas en inglés de *Bidirectional recurrent neural networks*). El bi-LSTM mostró la mejor precisión con un 82.9%.

De manera similar, en (Díaz-Ramírez y cols., 2022), los autores utilizaron un teléfono móvil para recolectar datos, pero emplearon una unidad de medición inercial como una alternativa más precisa y económica. Su principal contribución fue el uso de filtros de suavizado y técnicas de extracción de características en series temporales combinadas con algoritmos como Análisis Discriminante Lineal, Gaussiana Multivariante, Adaboost y kNN. El mejor rendimiento se obtuvo con kNN, con una precisión del 90 %.

En los trabajos precedentes, se demostró un alto nivel de precisión. Es fundamental destacar que estas variaciones se deben en gran medida a las diferencias en los conjuntos de datos utilizados. Por ejemplo, en la referencia (Vuong y cols., 2014), los autores emplearon un conjunto de datos relativamente pequeño que comprendía 220 experimentos de deambulación. Dentro de este conjunto, 97 experimentos correspondían al patrón directo, mientras que los patrones restantes de deambulación estaban representados por no más de 36 experimentos cada uno.

Los experimentos de deambulación se llevaron a cabo bajo condiciones altamente controladas en el trabajo presentado en (Barua y cols., 2020). Específicamente, solo se definieron de manera explícita y estática dos trayectorias para el patrón aleatorio. En la propuesta descrita en (Chaudhary y cols., 2020), las

2.4. EROSIÓN DEL BORDE DE ATAQUE EN TURBINAS EÓLICA 31

trayectorias de deambulaci3n tambi3n fueron predefinidas y caracterizadas por segmentos est3ticos. Sin embargo, su propuesta requiere el uso de infraestructura previamente instalada, es decir, sensores montados en el techo del 3rea de monitoreo. En contraste, nuestro enfoque propuesto fue evaluado usando un conjunto de datos integral (D3az-Ram3rez y cols., 2022) que abarca diversos patrones de deambulaci3n.

Adem3s, nuestro enfoque no est3 limitado a un sistema espec3fico de recopilaci3n de datos ni depende de patrones de deambulaci3n predefinidos. Dado que los patrones de deambulaci3n aleatoria y de vuelta comparten caracter3sticas de comportamiento similares, plantean desaf3os 3nicos para la extracci3n de caracter3sticas y la clasificaci3n.

2.4. Erosi3n del Borde de Ataque en Turbinas E3lica

La investigaci3n de (Best y cols., 2024) se centra en la clasificaci3n de la erosi3n del borde de ataque en turbinas e3licas *offshore* utilizando la base de datos LEES. Se emplearon varios modelos de ML, incluyendo SVM, RF y kNN. El modelo SVM con caracter3sticas extra3das mediante DWT logr3 una precisi3n del 90 %. La metodolog3a incluy3 un proceso exhaustivo de extracci3n de caracter3sticas y validaci3n cruzada para asegurar la robustez del modelo. Se utilizaron diferentes t3cnicas de extracci3n como DFT (o *Discrete Fourier Transform* por sus siglas en ingl3s) y HOG (o *Histogram of Oriented Gradients* por sus siglas en ingl3s), lo que permiti3 identificar patrones significativos en los datos.

La propuesta de (Chatterjee y Byun, 2023) se abord3 el diagn3stico de fallas en turbinas e3licas, utilizando una base de datos interna con registros hist3ricos

2.4. EROSIÓN DEL BORDE DE ATAQUE EN TURBINAS EÓLICA 32

de fallas. Se implementaron métodos híbridos que combinan técnicas de remuestreo con algoritmos como RF y SVM. La precisión del modelo alcanzó un 92 % en la clasificación correcta de fallas. La metodología se centró en el tratamiento del desequilibrio en los datos y utilizó técnicas avanzadas como el ajuste fino de hiperparámetros para mejorar el rendimiento del modelo.

En el trabajo de (Carmona Troyo, 2024) se centra en la predicción de fallas en palas de aerogeneradores utilizando la base de datos LEES (o *Leading-Edge Erosion Study* por sus siglas en inglés). En el Caso de Estudio III, se realizaron experimentos para analizar la reducción del rendimiento debido a la erosión y rugosidad de las palas. Se utilizaron coeficientes de presión (cp) obtenidos para diferentes números de Reynolds (Re) y ángulos de ataque (α). Se extrajeron características estadísticas para clasificar el daño en las palas, llevando a cabo seis experimentos con diferentes particiones de la base de datos. Los resultados mostraron una eficiencia de clasificación superior al 85 %, utilizando bibliotecas de AutoML en Python y el framework H2O-DAI.

En el trabajo de (Carmona Troyo, 2024) se aborda la predicción de fallas en palas de aerogeneradores a través de cuatro casos de estudio que utilizan diferentes bases de datos y técnicas. En el Caso de Estudio I, se comparan los métodos de clasificación propuestos por investigadores del área con los utilizados en el trabajo de Carmona, usando una base de datos externa para validar la efectividad de ambos enfoques. El Caso de Estudio II se centra en una base de datos construida mediante simulaciones, donde se analizaron características estadísticas de potencia y aceleración para clasificar el daño en distintas zonas del borde de ataque de las palas: superior, inferior y combinadas. El Caso de Estudio III, considerado el más relevante, utiliza la base de datos LEES (o *Leading-Edge Erosion Study* por sus siglas en inglés) y aborda la clasificación de daños en palas utilizando coeficientes de presión (cp) obtenidos para diferentes

2.4. EROSIÓN DEL BORDE DE ATAQUE EN TURBINAS EÓLICA 33

números de Reynolds (Re) y ángulos de ataque (α). En este caso, se realizaron seis experimentos empleando particiones de la base de datos, lo que permitió obtener resultados de clasificación con una eficiencia superior al 85 %, utilizando bibliotecas de AutoML en Python y el framework H2O-DAI. Finalmente, en el Caso de Estudio IV, se llevaron a cabo experimentos con una pala instrumentada con sensores de presión y probada en un túnel de viento, analizando su desempeño bajo diferentes velocidades, ángulos de ataque y niveles de erosión. Estos experimentos destacaron la capacidad de los algoritmos de ML para clasificar patrones complejos de daño.

Capítulo 3

Caso de Estudio 1:

Detección de patrones de deambulación

El primer caso de estudio se centra en la detección de patrones de deambulaciones, un problema crítico en el monitoreo de enfermedades neurodegenerativas. Este capítulo detalla la metodología utilizada, que incluye desde la selección del conjunto de datos y la extracción de características hasta la evaluación del rendimiento de diferentes modelos. Además, se presentan y analizan los resultados, destacando las implicaciones clínicas y técnicas del enfoque propuesto. Este capítulo demuestra cómo el análisis de series de tiempo puede aportar soluciones efectivas en el ámbito de la salud.

3.1. METODOLOGÍA

35



Figura 3.1: Patrones de deambulaci3n: a) directo, b) pacing, c) lapping d) random.

3.1. Metodologí

La metodologí empleada se puede describir en cuatro etapas. La primera etapa consistió en definir un conjunto de datos basado en (Díaz-Ramírez y cols., 2022). En la segunda etapa, se plantearon dos enfoques para caracterizar los patrones de deambulaci3n. El primer enfoque utiliza tres representaciones diferentes de series de tiempo. La representaci3n más básica emplea las coordenadas X e Y de cada patr3n. Adem3s, se derivan dos series de tiempo adicionales: la serie de tiempo de *slope* (o pendiente, por su traducci3n en espaól del inglés) y la serie de tiempo de eficiencia del camino. En este enfoque, los clasificadores deben detectar patrones directamente a partir de las series de tiempo sin procesar.

El segundo enfoque aplica la extracci3n de características a cada serie de tiempo para facilitar la clasificaci3n. Se emplea la DWT y se calculan diversos descriptores estadísticos, tales como medidas de tendencia central, medidas de dispersi3n, percentiles (5 %, 25 %, 75 % y 95 %), entropía y la cantidad de índices

3.1. METODOLOGÍA

36

de cruce no nulo y cruce no medio.

En la tercera etapa, se implementan modelos de ML para generar clasificadores, comparando tanto las series de tiempo sin procesar como las características extraídas.

Finalmente, la cuarta etapa emplea la BO para el ajuste de hiperparámetros. Además, el trabajo experimental evalúa el desempeño utilizando hiperparámetros predeterminados como referencia.

3.1.1. Dataset

El dataset contiene 1,600 patrones de deambulación, 400 de cada uno de los cuatro tipos (Martino-Saltzman y cols., 1991): directo, pacing, lapping y random (Díaz-Ramírez y cols., 2022) como se observa en la figura 3.1. Existe variación en la longitud y duración de cada patrón. En (Díaz-Ramírez y cols., 2022), se estima el número de pasos a partir de las lecturas del sensor, con una longitud promedio de paso de 0.4m. El número de pasos varía entre 10 y 200 (excluyendo valores atípicos), mientras que la duración varía entre 10 y 25 segundos. La Figura 3.2 resume el número de pasos para cada patrón mediante diagramas de caja.

3.1.2. Imputación de Datos

Para tener el mismo número de lecturas en cada experimento de deambulación, se estableció un límite de 80 lecturas (pasos) por experimento utilizando los siguientes criterios. Considere n como el número total de pasos en un episodio de deambulación, r como el resultado de $80 - n$ y r^2 como el resultado de $n - 80$, entonces:

- Caso 1: Si la lectura tiene menos de 80 pasos, se agregan h pasos al inicio y h' al final, donde

3.1. METODOLOGÍA

37

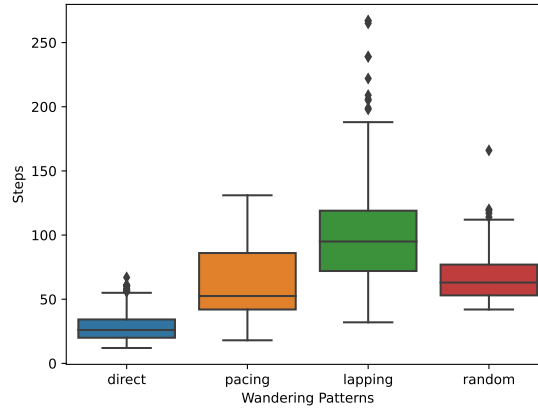


Figura 3.2: Pasos por patrón de deambulación en el conjunto de datos.

$$h = \lfloor \frac{r}{2} \rfloor, \quad (3.1)$$

$$h' = \begin{cases} h, & \text{si } r \text{ es par} \\ h + 1, & \text{de lo contrario} \end{cases} \quad (3.2)$$

El objetivo es dividir el resto en dos partes iguales. A partir de esta premisa, la cantidad de pasos a agregar se divide por dos y se redondea hacia abajo para trabajar exclusivamente con valores enteros. Para h' , si h es par, entonces h y h' son iguales. De lo contrario, debido al error de redondeo, h' es igual a h más uno.

- Caso 2: Si la lectura tiene más de 80 pasos, entonces se sustrae un número m de pasos al inicio y m' al final de la lectura, similar al Caso 1, de manera que

$$m = \lfloor \frac{r^2}{2} \rfloor, \quad (3.3)$$

3.1. METODOLOGÍA

38

$$m' = \begin{cases} m, & \text{si } r2 \text{ es par} \\ m + 1, & \text{de lo contrario} \end{cases} \quad (3.4)$$

- Caso 3: Si la lectura tiene 80 pasos, no se realiza ningún cambio.

3.1.3. Extracción de Características

En este estudio, el sistema propuesto se centra en las coordenadas cartesianas de cada paso durante la trayectoria de cada patrón de deambulaci3n, como se describe en (Díaz-Ramírez y cols., 2022), organizando los puntos de datos en orden cronológico para crear una serie temporal. Las siguientes series temporales se extrajeron de las lecturas del IMU:

- Coordenadas (Coords): los datos de coordenadas brutas se extraen de los sensores IMU, como se describe en (Díaz-Ramírez y cols., 2022). Son dos series temporales, una para las posiciones x y otra para las posiciones y .
- Pendiente: la pendiente de una línea es la medida de su inclinaci3n o empinamiento y representa la tasa de cambio de la variable dependiente respecto a la variable independiente. Matemáticamente, la pendiente (m) se calcula como la relaci3n entre el cambio vertical (Δy) y el cambio horizontal (Δx) entre dos puntos: $m = \frac{\Delta y}{\Delta x}$.
- Eficiencia del Camino (Path-Eff): relaci3n entre la longitud del camino 3ptimo y la longitud del camino real, basada en (Kumar y cols., 2016).

La Figura 3.3 muestra ejemplos de cada serie temporal. Los gráfcos muestran la serie temporal de Coords (tanto el componente x , denotado como $Coords\ x$, como el componente y , denotado como $Coords\ y$), así como la Pendiente y la Eficiencia del Camino para una muestra de cada clase (Direct, Lapping, Pacing

3.1. METODOLOGÍA

39

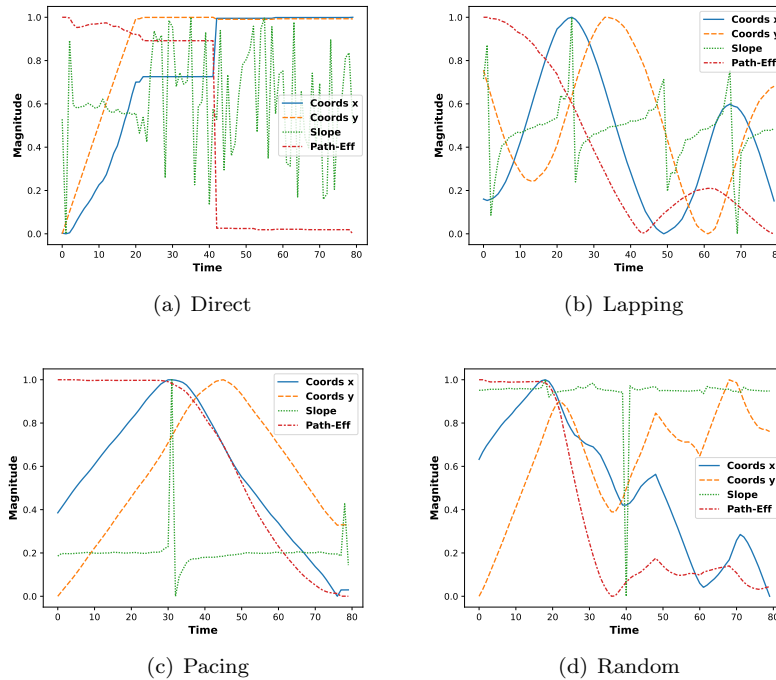


Figura 3.3: Ejemplos de series temporales para una muestra de cada clase: (a) Direct; (b) Lapping; (c) Pacing; and (d) Random.

y Random). Cada serie temporal está normalizada en el rango $[-1, 1]$ en estos gráficos.

La extracción de características es una parte significativa de cualquier sistema de ML (AlSharabi y cols., 2022). Este trabajo emplea la DWT, que puede ser utilizada para cuantificar tendencias, puntos de inflexión, discontinuidades y auto-similitudes. El método aprovecha un conjunto de funciones base localizadas en los dominios del tiempo y la frecuencia (Mao y Aggarwal, 2001), y puede ser implementado de manera eficiente (Ukil y Barlocher, 2007). Existen varias familias de wavelets, pero el presente trabajo utiliza la Wavelet Biortogonal rbio3.1, dado su sólido desempeño en dominios similares (Wang y cols.,

3.1. METODOLOGÍA

40

2016).

La DWT se emplea para extraer características descriptivas de los datos de series temporales, con las características representadas por los coeficientes de wavelet calculados. El siguiente procedimiento se lleva a cabo para extraer un total de 12 características. Primero, se selecciona una serie temporal (coordenadas crudas, pendiente o eficiencia del camino) como señal de entrada. Segundo, se aplica la DWT a la serie temporal y se extraen las siguientes características. *Entropía*, que se refiere a una medida del nivel de desorden, aleatoriedad o incertidumbre en un conjunto de datos. Además, se calculan los siguientes descriptores estadísticos: los percentiles del 5 %, 25 %, 75 % y 95 %; la *mediana*, la *media*, la *desviación estándar*, la *varianza* y la *media cuadrática*. También se calculan los siguientes descriptores conocidos como *cruces*. *Índices de Cruces No-Cero*, que se refiere a la cantidad de instancias donde una señal cruza el eje cero sin ser igual a cero. Esto puede proporcionar información sobre el comportamiento oscilatorio de la señal. *Índices de Cruces de la Media*, que se refiere a la cantidad de instancias donde una señal cruza su valor medio. Esto ofrece información sobre el comportamiento de la señal alrededor de su tendencia central. Finalmente, todos los datos se escalan utilizando la normalización min-max.

3.1.4. Configuración Experimental

Instrumentación y Especificaciones de los Sensores

Los datos fueron adquiridos utilizando un teléfono inteligente Xiaomi Mi A1 equipado con: un acelerómetro Bosch (modelo 2062600) con una resolución de 0.002 m/s^2 ; un magnetómetro AKM (modelo 1) con una resolución de $0.01 \text{ } \mu\text{T}$; un giroscopio Bosch (modelo 2062600) con una resolución de 0.06 deg/s ; y otros sensores (Díaz-Ramírez y cols., 2022). Los sensores fueron calibrados y la

3.1. METODOLOGÍA

41

frecuencia de muestreo fue de 500 Hz. Para calibrar el giroscopio, el teléfono se colocó sobre una superficie plana sin moverlo, y luego se aplicó un filtro pasabajos. Para calibrar el magnetómetro, el teléfono se movió lentamente siguiendo una trayectoria similar a un número 8, y se eliminó el sesgo. La tasa de muestreo fue de 500 Hz; se proporcionan más detalles en (Díaz-Ramírez y cols., 2022).

Hardware Utilizado para el Entrenamiento de ML

Los experimentos se llevaron a cabo en el entorno de nube de AWS utilizando el servicio Amazon SageMaker en una instancia ml.c5.2xlarge, que ofrece 8 CPU virtuales y 16 GiB de RAM. Las instancias C5 están equipadas con procesadores Intel Xeon Platinum 8000 de primera o segunda generación (AWS, 2024).

Software y Librerías

La implementación del modelo se realizó utilizando el lenguaje de programación Python. Para el desarrollo y entrenamiento de los modelos de ML se empleó Scikit-learn (Pedregosa y cols., 2011). Las representaciones gráficas se generaron utilizando Matplotlib (Hunter, 2007) y Seaborn (Waskom, 2021). Para la BO se utilizó la librería Hyperopt (Bergstra y cols., 2015), y para la DWT se empleó la librería PyWavelet¹.

Métricas de Evaluación

Se realizaron un total de 10 ejecuciones de validación cruzada de 5 pliegues; es decir, 50 ejecuciones para cada configuración. Para la optimización de hiperparámetros se utilizó un nivel adicional de validación cruzada (validación cruzada anidada) con tres pliegues. Los experimentos se evaluaron en términos de rendimiento promedio, basado en exactitud (accuracy) y puntaje F1, así como promedios macro de precisión, exhaustividad (recall) y el área bajo la curva

¹<https://pywavelets.readthedocs.io>

3.2. RESULTADOS

42

(AUC) de la característica operativa del receptor (ROC) calculada mediante un enfoque uno contra todos (one-vs-rest). La matriz de confusión se calcula para las mejores configuraciones experimentales con el fin de realizar análisis y obtener detalles adicionales.

3.2. Resultados

Todos los ensayos experimentales emplearon datos escalados con el escalador Min-Max (Pedregosa y cols., 2011), todas las características en el rango $[0, 1]$. La Tabla 3.2 resume todos los resultados en los datos de prueba para cada condición experimental. La tabla presenta el puntaje promedio logrado en cada métrica, calculado en 50 ejecuciones (10 repeticiones de validación cruzada de cinco pliegues). Se utilizaron tres clasificadores (SVC, RF, KNN), con (BO) y sin (usando los valores por defecto) optimización de hiperparámetros, utilizando extracción de características (DWT) o las series temporales sin procesar (crudas). Se consideraron cuatro series temporales: Coords, Slope, Path Eff y una combinación de Coords y Slope (Coords-Slope). Los resultados se agrupan en cuatro bloques principales en la Tabla 3.2, considerando el uso (o no) de optimización de hiperparámetros y extracción de características. Los ajustes de hiperparámetros para cada clasificador se resumen en la Tabla 3.1. El código para replicar los experimentos está disponible en: <https://github.com/DanielRR10/wandering-patterns-time-series>.

Los resultados de los modelos que usan los datos crudos con valores predefinidos de hiperparámetros constituyen el primer grupo en la Tabla 3.2. El modelo con el mejor desempeño fue el clasificador RF utilizando la serie temporal Coords, logrando una precisión promedio de 0.962 y un AUC de 0.999. El segundo grupo de resultados en la Tabla 3.2 también se basa en series temporales crudas, pero con optimización de hiperparámetros. La precisión de la mayoría de

3.2. RESULTADOS

43

Cuadro 3.1: Valores de hiperparámetros utilizados para cada clasificador, mostrando los valores predeterminados y el rango de búsqueda utilizado por BO.

Algoritmo	Hiperparámetro	Predeterminado	Rango BO
KNN	Vecinos (k)	5	[3, 20]
KNN	Métrica	minkowski	{manhattan, euclidean, chebyshev, minkowski}
KNN	Pesos	uniforme	{uniforme, distancia}
KNN	Algoritmo	auto	{ball-tree, kd-tree, brute, auto}
SVC	C	1	$[10^{-2}, 10]$
SVC	Gamma	scale	$[10^{-2}, 10]$
SVC	Grado	3	[2, 5]
SVC	Kernel	rbf	{rbf, sigmoid, poly, linear}
Random Forests	Criterio	gini	{entropy, gini, log-loss}
Random Forests	Profundidad máxima	Ninguna	[1, 21]
Random Forests	Máx. características	auto	{sqrt, log2, Ninguna}
Random Forests	N-estimadores	100	[50, 210]

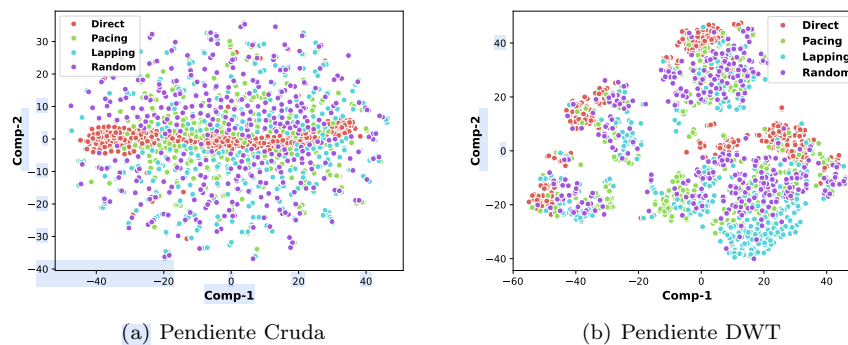


Figura 3.4: Proyecciones bidimensionales de los datos de la serie temporal Slope, considerando: (a) la serie cruda y (b) la extracción de características con DWT. Los gráficos se generan utilizando el método t-SNE.

los algoritmos mejoró. Por ejemplo, utilizando la serie temporal Coords-Slope con el clasificador kNN, la precisión mejoró en un 20 %. Sin embargo, en la mayoría de los casos, la mejora fue menor. Nuevamente, RF logró el AUC más alto de 0.99, mientras que el clasificador SVC logró el mejor desempeño en la mayoría de las métricas.

3.2. RESULTADOS

44

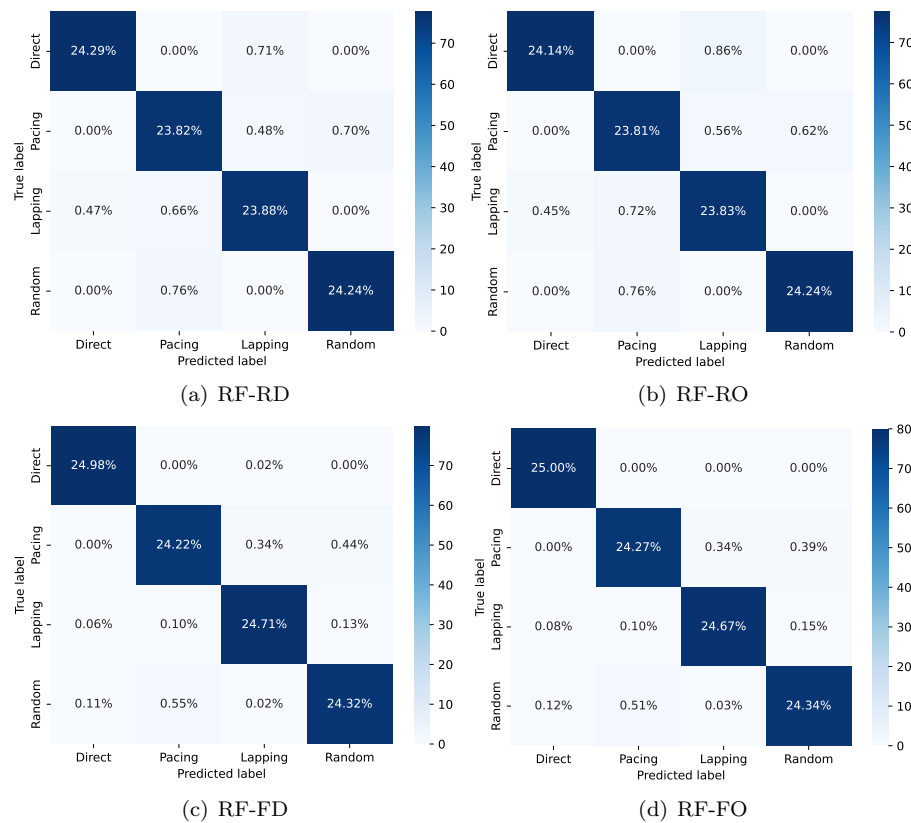


Figura 3.5: Gráficos de matrices de confusión de los puntajes de rendimiento para las mejores configuraciones experimentales de los cuatro grupos de métodos resumidos en la Tabla 3.2. En cada caso se muestran la Precisión, F1, Precisión media, Recuperación y AUC ROC calculados en los conjuntos de prueba.

Los siguientes dos grupos de resultados utilizan extracción de características, lo que llevó a mejoras de precisión con la mayoría de las configuraciones superando el 90 % de precisión. En estos grupos, la optimización de hiperparámetros fue aún menos significativa que antes. El mejor clasificador, RF, fue insensible a la optimización de hiperparámetros, logrando el mismo desempeño con valores predeterminados. RF utilizando la serie temporal Coords-Slope alcanzó la precisión más alta del 98.2 %. Para ilustrar el impacto de la extracción de características con DWT, se realizó un análisis adicional de dos conjuntos de datos: la

3.2. RESULTADOS

45

señal cruda y la extracción de características DWT de la serie temporal Slope. Considere los resultados de SVC en la Tabla 3.2 para ambos casos. Usando la serie temporal cruda, la precisión fue del 22.6 % con hiperparámetros predeterminados y del 24.1 % con BO. Por otro lado, con la extracción de características DWT, las puntuaciones de precisión fueron del 70.9 % y 70.7 %, respectivamente, un gran incremento en el rendimiento. Para entender las razones, se generaron visualizaciones 2D de ambos conjuntos de datos usando t-SNE (van der Maaten y Hinton, 2008), mostradas en la Figura 3.4. Para la serie temporal cruda, existe una gran superposición entre las clases, lo que indica una tarea de clasificación difícil. Por el contrario, usando características DWT, cada clase se proyecta en clústeres multimodales más separables, explicando el aumento del rendimiento.

En general, los mejores resultados se obtuvieron con el clasificador RF. La Figura 3.5 presenta una representación detallada de las mejores configuraciones, a saber: 1) Clasificador RF utilizando la serie temporal cruda Coords con hiperparámetros predeterminados (KNN-RD); 2) Clasificador RF con la serie temporal cruda Coords-Slope y optimización de hiperparámetros (RF-RO); 3) Clasificador RF con características DWT y hiperparámetros predeterminados (RF-FD); y 4) Clasificador RF con características DWT y optimización de hiperparámetros (RF-FO). La Figura 3.5 presenta la matriz de confusión promedio para cada clasificador. En general, el rendimiento es muy sólido, mostrando que: la optimización de hiperparámetros no es crucial y que la extracción de características basada en DWT logra un mejor rendimiento, pero las series temporales crudas también se pueden usar para obtener resultados competitivos. El enfoque propuesto no solo logró resultados cercanos a los de (Barua y cols., 2020), sino que también operó en un conjunto de datos notable y único en este dominio (Díaz-Ramírez y cols., 2022).

Las últimas tres filas de la Tabla 3.2 muestran el rendimiento de métodos

3.3. DISCUSIÓN

46

de ML diseñados específicamente para la clasificación de series temporales desde SKTIME ². Los métodos son BOSS (Schäfer y Högvist, 2012), ROCKET (Dempster y cols., 2020b) y HIVE-COTE 2.0 (Middlehurst y cols., 2021a). Estos métodos se ejecutaron sin optimización de hiperparámetros, mostrando solo la configuración que logró los mejores resultados. La configuración RF-FD es extremadamente competitiva, superando al algoritmo considerado como el estado-del-arte, HIVE-COTE 2.0.

3.3. Discusión

Este trabajo propone una metodología para clasificar cuatro tipos diferentes de patrones de deambulación, altamente correlacionados con la demencia, utilizando el reciente conjunto de datos de (Díaz-Ramírez y cols., 2022). El trabajo experimental propone tres representaciones diferentes de cada patrón de deambulación, expresadas como series temporales. La metodología también aplica la DWT para extraer características descriptivas y emplea tres clasificadores de ML, utilizando configuraciones de hiperparámetros predeterminados y optimizados.

Los resultados experimentales muestran que se logra un mejor rendimiento utilizando características extraídas con DWT en lugar de las series temporales crudas, aunque la diferencia es menor de lo que se anticipaba inicialmente. El clasificador RF supera a otros métodos de ML, y el mejor rendimiento se alcanza utilizando la serie temporal combinada Coords-Slope. Para el clasificador RF, la optimización de los hiperparámetros no produjo mejoras notables.

Algunas de las principales limitaciones del trabajo actual son la simplificación relativa del problema general al utilizar un conjunto de datos derivado de personas sanas, y el no abordar cuestiones de privacidad y seguridad que se en-

²<https://www.sktime.net>

3.3. DISCUSIÓN

47

Classifier	Hyperparameters	Features	Time Series	Accuracy	F1	Precision	Recall	AUC
SVC	Default	Raw	Coords	0.943	0.943	0.943	0.943	0.997
SVC	Default	Raw	Coords-Slope	0.268	0.161	0.281	0.282	0.701
SVC	Default	Raw	Path-Eff	0.848	0.848	0.865	0.848	0.964
SVC	Default	Raw	Slope	0.226	0.092	0.057	0.250	0.486
RF	Default	Raw	Coords	0.962	0.961	0.961	0.962	0.999
RF	Default	Raw	Coords-Slope	0.962	0.961	0.961	0.962	0.999
RF	Default	Raw	Path-Eff	0.891	0.890	0.893	0.891	0.981
RF	Default	Raw	Slope	0.797	0.799	0.804	0.798	0.944
KNN	Default	Raw	Coords	0.956	0.956	0.956	0.956	0.998
KNN	Default	Raw	Coords-Slope	0.685	0.680	0.683	0.686	0.887
KNN	Default	Raw	Path-Eff	0.872	0.871	0.883	0.872	0.964
KNN	Default	Raw	Slope	0.528	0.500	0.509	0.529	0.794
SVC	BO	Raw	Coords	0.962	0.962	0.962	0.962	0.999
SVC	BO	Raw	Coords-Slope	0.348	0.237	0.210	0.353	0.548
SVC	BO	Raw	Path-Eff	0.893	0.892	0.895	0.893	0.980
SVC	BO	Raw	Slope	0.241	0.097	0.060	0.250	0.491
RF	BO	Raw	Coords	0.961	0.961	0.961	0.962	0.999
RF	BO	Raw	Coords-Slope	0.961	0.961	0.961	0.962	0.999
RF	BO	Raw	Path-Eff	0.892	0.891	0.894	0.892	0.983
RF	BO	Raw	Slope	0.810	0.811	0.815	0.810	0.950
KNN	BO	Raw	Coords	0.961	0.960	0.961	0.961	0.998
KNN	BO	Raw	Coords-Slope	0.899	0.899	0.900	0.899	0.980
KNN	BO	Raw	Path-Eff	0.876	0.875	0.884	0.876	0.958
KNN	BO	Raw	Slope	0.557	0.537	0.540	0.558	0.781
SVC	Default	DWT	Coords	0.961	0.961	0.961	0.961	0.997
SVC	Default	DWT	Coords-Slope	0.963	0.963	0.964	0.963	0.998
SVC	Default	DWT	Path-Eff	0.923	0.922	0.924	0.923	0.991
SVC	Default	DWT	Slope	0.709	0.705	0.708	0.710	0.899
RF	Default	DWT	Coords	0.961	0.961	0.961	0.961	0.997
RF	Default	DWT	Coords-Slope	0.982	0.982	0.982	0.982	0.999
RF	Default	DWT	Path-Eff	0.924	0.923	0.925	0.924	0.989
RF	Default	DWT	Slope	0.842	0.841	0.843	0.842	0.965
KNN	Default	DWT	Coords	0.952	0.952	0.952	0.952	0.991
KNN	Default	DWT	Coords-Slope	0.950	0.950	0.951	0.950	0.992
KNN	Default	DWT	Path-Eff	0.901	0.899	0.905	0.900	0.971
KNN	Default	DWT	Slope	0.624	0.611	0.611	0.625	0.829
SVC	BO	DWT	Coords	0.960	0.960	0.960	0.960	0.995
SVC	BO	DWT	Coords-Slope	0.963	0.963	0.964	0.963	0.996
SVC	BO	DWT	Path-Eff	0.924	0.924	0.924	0.924	0.989
SVC	BO	DWT	Slope	0.707	0.703	0.703	0.708	0.855
RF	BO	DWT	Coords	0.962	0.962	0.962	0.962	0.997
RF	BO	DWT	Coords-Slope	0.982	0.982	0.982	0.982	0.999
RF	BO	DWT	Path-Eff	0.924	0.923	0.925	0.924	0.990
RF	BO	DWT	Slope	0.842	0.842	0.845	0.842	0.966
KNN	BO	DWT	Coords	0.958	0.957	0.959	0.958	0.992
KNN	BO	DWT	Coords-Slope	0.961	0.960	0.962	0.960	0.993
KNN	BO	DWT	Path-Eff	0.911	0.909	0.914	0.911	0.971
KNN	BO	DWT	Slope	0.693	0.687	0.693	0.692	0.889
BOSS	Default	Raw	Coords-Slope	0.939	0.937	0.941	0.939	0.996
ROCKET	Default	Raw	Coords	0.984	0.984	0.984	0.984	0.989
HIVE-COTE 2.0	Default	Raw	Coords	0.966	0.966	0.966	0.966	0.997

Cuadro 3.2: Promedios de cada medida de rendimiento para cada condición experimental, mostrando: accuracy, F1 score, precision, recall y ROC-AUC. Los mejores resultados para cada grupo de resultados están indicados en negrita.

frentarían durante su implementación en el mundo real. Por lo tanto, el trabajo futuro se centrará en la integración de la metodología propuesta dentro de un sistema que pueda capturar y modelar datos recolectados en un entorno clínico real, considerando plenamente los problemas mencionados anteriormente.

3.3. DISCUSIÓN

48

No obstante, los resultados son altamente competitivos con el estado-del-arte en la clasificación de series temporales y demuestran que se puede lograr un buen rendimiento clasificando patrones de deambulaci3n basados en IMUs, sin requerir m3todos que necesiten grandes cantidades de datos o recursos computacionales.

68

Capítulo 4

Caso de Estudio 2:

Detección de Erosión y

daños en palas de turbinas eólicas

El segundo caso de estudio aborda la detección de erosión en palas de aerogeneradores, un problema clave para la sostenibilidad de las energías renovables. Este capítulo describe los métodos aplicados para caracterizar y clasificar los daños a partir de series temporales de coeficientes de presión. A través de una discusión crítica, se evalúan los resultados obtenidos y se comparan con enfoques alternativos. Este capítulo ilustra cómo las técnicas analíticas pueden optimizar el rendimiento en aplicaciones industriales complejas.

4.1. METODOLOGÍA

50

4.1. Metodología

La metodología empleada en este estudio se divide en cuatro etapas principales:

1. Definición del Conjunto de Datos: La primera etapa consistió en definir un conjunto de datos basado en la base de datos LEES (Maniaci y cols., 2022).
2. Enfoques de Caracterización: Se plantearon dos enfoques para caracterizar los patrones de erosión:
 - Representación Directa de Series de Tiempo: Este enfoque utiliza tres representaciones distintas de series de tiempo. La representación más básica emplea los coeficientes de presión en el perfil aerodinámico NACA 633-418, bajo diferentes configuraciones de números de Reynolds y ángulos de ataque, como se describe en (Maniaci y cols., 2022). En este enfoque, los clasificadores deben detectar patrones directamente a partir de las series de tiempo sin procesar.
 - Extracción de Características: En el segundo enfoque, el procedimiento es similar al descrito en la sección 3.1, correspondiente a la metodología del caso 1. Sin embargo, se emplearon series de tiempo diferentes, utilizando los coeficientes de presión como base para el análisis.
3. Implementación de Modelos de ML: Se implementaron modelos de ML para generar clasificadores, comparando el desempeño entre series de tiempo sin procesar y las características extraídas.
4. BO: La última etapa aplicó BO para el ajuste de hiperparámetros. Además, se evaluó el desempeño utilizando hiperparámetros predeterminados como

4.1. METODOLOGÍA

51

referencia.

4.1.1. Dataset

El conjunto de datos utilizado se derivó del proyecto LEES (Maniaci y cols., 2022). Este conjunto incluye:

- Números de Reynolds:
 - $Re_1 = 16 \times 10^6$
 - $Re_2 = 24 \times 10^6$
 - $Re_3 = 32 \times 10^6$
 - $Re_4 = 40 \times 10^6$
- Ángulos de Ataque: Rango de -4° a 16° (21 ángulos en total).
- Coeficientes de Presión (C_p) para Cada Ángulo de Ataque: El coeficiente de presión permite analizar la reducción de la sustentación máxima y el aumento de la resistencia debido a la rugosidad. Los coeficientes fueron calculados para 58 puntos de presión en la superficie de la pala. La Figura ?? muestra los coeficientes de presión por cada punto, donde el eje x (x/c) representa la posición de medición en la pala que en nuestro caso sera nuestra medida de tiempo, y el eje y representa el coeficiente calculado (C_p).

Para los experimentos, se utilizaron datos del perfil aerodinámico NACA63-418 en tres condiciones:

- Palas con superficie sana.
- Palas erosionadas.

4.1. METODOLOGÍA

52

- Palas con diferentes niveles de rugosidad aditiva ($XXX\mu mYY$), donde XXX es la altura de la rugosidad en micrómetros y YY la densidad como porcentaje de cobertura.

El resumen de las señales y los sensores de presión en la base de datos LEES se encuentra en la tabla 4.1.

Clases	Señales	Sensores de presión
Clean	76	53
Eroded	74	58
100 μ 3	72	53
100 μ 09	74	53
100 μ 15	75	53
140 μ 03	70	53
140 μ 03D	74	53
140 μ 03E	76	53
200 μ 03	73	53

Cuadro 4.1: El número de señales corresponde al número de muestras tomadas para 4 números diferentes de Reynolds y una media de 21 ángulos de ataque diferentes. Se obtuvieron 21 rasgos estadísticos para la clasificación.

Clases de Condición de la Pala

En el proyecto LEES, la rugosidad se considera un indicador característico del daño. Así, se definieron nueve clases para clasificar los daños:

- Clase 0: Pala sana (identificada como *Clean* por su traducción inglés).
- Clase 1: Pala erosionada (identificada como *Eroded* por su traducción inglés).
- Clase 2: Rugosidad aditiva 100 μm 03 (simulación de características de insectos).
- Clase 3: Rugosidad aditiva 100 μm 9.
- Clase 4: Rugosidad aditiva 100 μm 15.

4.1. METODOLOGÍA

53

- Clase 5: Rugosidad aditiva $140\mu\text{m}03$.
- Clase 6: Rugosidad tipo diamante $140\mu\text{m}03$.
- Clase 7: Rugosidad extendida $140\mu\text{m}03$.
- Clase 8: Rugosidad aditiva $200\mu\text{m}03$.

La Figura 4.1 muestra la gráfica de los coeficientes de presión calculados para cada punto de presión en la pala. El eje x , representado por x/c en el dataset del proyecto LEES, indica la posición donde se tomó la presión en la superficie de la pala. Sin embargo, en este caso, se considera como una muestra en serie temporal, razón por la cual está etiquetado como tiempo. Por su parte, el eje y , etiquetado como C_p , corresponde al coeficiente de presión calculado.

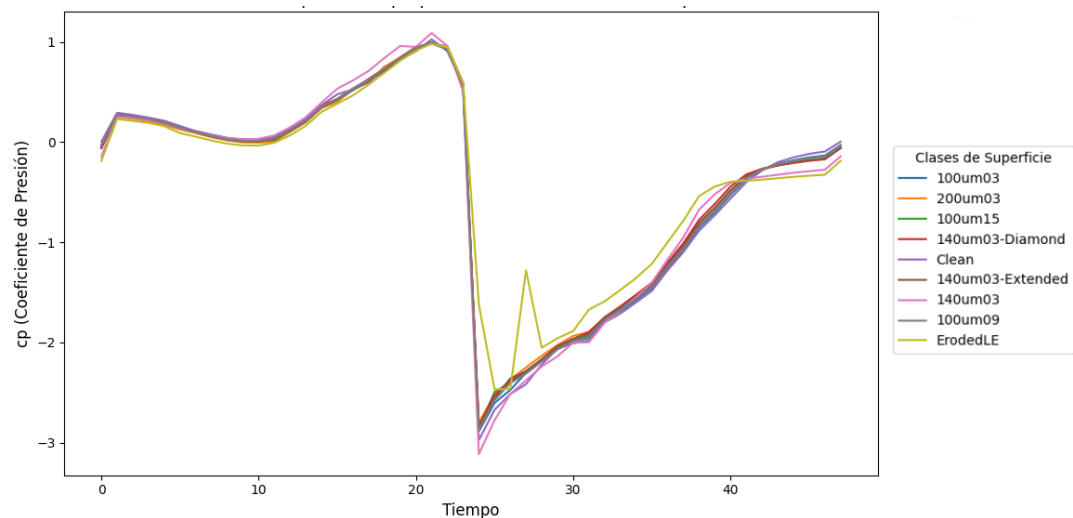


Figura 4.1: Evolución de los coeficientes de presión (cp) en función del tiempo para diferentes clases de superficie, mostrando las variaciones aerodinámicas asociadas con cada categoría.

Preparación de la Base de Datos

La preparación del conjunto de datos consistió en:

4.1. METODOLOGÍA

54

- Extraer información de cada número de Reynolds, representando la dinámica del flujo sobre la pala.
- Definir clases representativas del estado de las palas.
- Configurar experimentos a diferentes ángulos de ataque (-4° a $+16^\circ$) para cada configuración de pala.

4.1.2. Extracción de Características

De manera similar al primer caso de estudio, se utilizó la DWT para extraer características descriptivas de las series temporales. En total, se derivaron 12 características clave, que incluyen: entropía; percentiles del 5 %, 25 %, 75 % y 95 %; *mediana*, *media*, *desviación estándar*, *varianza* y *media cuadrática*.

Adicionalmente, se calcularon descriptores relacionados con patrones de cruce en las señales, conocidos como: *índices de cruces no-cero* e *índices de cruces de la media*. Por último, todos los datos se normalizaron utilizando el método de escalamiento min-max para garantizar uniformidad en el rango de valores.

Hardware Utilizado para el Entrenamiento de ML

Los experimentos se llevaron a cabo en una partición de un servidor ubicado en el Instituto Tecnológico de Tijuana. El servidor cuenta con un procesador de 16 núcleos, 32 GiB de memoria RAM y 600 GiB de almacenamiento en disco HDD.

Software y Librerías

La implementación del modelo se llevó a cabo utilizando Python como lenguaje de programación principal. Para el desarrollo y entrenamiento de los modelos de ML, se empleó Scikit-learn (Pedregosa y cols., 2011). Las visualizaciones

70

4.2. RESULTADOS

55

gráficas fueron generadas con las librerías Matplotlib (Hunter, 2007) y Seaborn (Waskom, 2021).

La BO se realizó mediante la librería Hyperopt (Bergstra y cols., 2015), mientras que la DWT fue implementada utilizando la librería PyWavelet¹.

Métricas de Evaluación

Se llevaron a cabo 10 ejecuciones de validación cruzada con 5 pliegues, lo que resultó en un total de 50 ejecuciones por configuración experimental. Para la optimización de hiperparámetros, se empleó un esquema de validación cruzada anidada con tres pliegues.

El desempeño de los experimentos se evaluó utilizando métricas estándar como la exactitud (*accuracy*), el puntaje F1 y los promedios macro de precisión, *recall* y el área bajo la curva (AUC) de la característica operativa del receptor (ROC), calculados mediante un enfoque *one-vs-rest*.

Es importante mencionar que se consideraron 6 problemas de clasificación tomando en cuenta el tipo de daño y las clases generadas.

4.2. Resultados

Aquí tienes una versión reescrita para mantener la originalidad del texto:

El propósito de este trabajo es clasificar el tipo de daño que se encuentra en el borde de ataque de una pala de aerogenerador, utilizando el coeficiente de presión calculado a lo largo del perfil aerodinámico como señal de entrada y extrayendo un conjunto de características en el dominio del tiempo.

Es relevante señalar que algunos términos que se presentan en las tablas de resultados, como en la Tabla 4.2, deben ser aclarados. En las columnas correspondientes a los clasificadores se mencionan los modelos empleados para la

¹<https://pywavelets.readthedocs.io>

4.2. RESULTADOS

56

clasificación. En las columnas de hiperparámetros, se especifica si se utilizaron los valores por defecto (Default) o si se optó por versiones optimizadas mediante BO. A continuación, en las columnas de características, se señala si se usaron los datos originales (raw) o si se aplicó la DWT. Por último, en las columnas de métricas se muestran los valores de accuracy, F1, precision, recall y AUC.

4.2.1. Problema A: Sana-Erosionada

Este es el más sencillo de los problemas planteados, se trata de un problema de clasificación binario y muy equilibrado, en el que el objetivo es diferenciar entre una hoja limpia y otra erosionada. Los resultados se presentan en el Cuadro 4.2.

Classifier	Hyperparameters	Features	Accuracy	F1	Precision	Recall	AUC
RF	Default	DWT	1.000	1.000	1.000	1.000	1.000
HIVECOTE2	Default	Raw	1.000	1.000	1.000	1.000	1.000
SVC	Default	DWT	1.000	1.000	1.000	1.000	1.000
KNN	BO	Raw	1.000	1.000	1.000	1.000	1.000
SVC	BO	Raw	1.000	1.000	1.000	1.000	1.000
SVC	BO	DWT	1.000	1.000	1.000	1.000	1.000
RF	BO	DWT	1.000	1.000	1.000	1.000	1.000
ROCKET	Default	Raw	1.000	1.000	1.000	1.000	1.000
KNN DDTW	Default	Raw	1.000	1.000	1.000	1.000	1.000
KNN	BO	DWT	1.000	1.000	1.000	1.000	1.000
Shapelet	Default	Raw	0.999	0.999	0.999	0.999	1.000
BOSS	Default	Raw	0.997	0.997	0.997	0.997	1.000
TimeSeriesForest	Default	Raw	0.997	0.997	0.997	0.997	1.000
Mini ROCKET	Default	Raw	0.996	0.996	0.996	0.996	0.996
KNN DTW	Default	Raw	0.991	0.991	0.992	0.991	0.991
KNN	Default	DWT	0.982	0.981	0.983	0.983	1.000
RF	Default	Raw	0.966	0.965	0.967	0.967	0.996
RF	BO	Raw	0.962	0.962	0.965	0.962	0.989
KNN	Default	Raw	0.897	0.894	0.904	0.898	0.973
SVC	Default	Raw	0.859	0.853	0.889	0.863	0.989

Cuadro 4.2: Resultados obtenidos para el problema A, un problema binario y altamente equilibrado que utiliza las clases limpias y erosionadas.

Los resultados presentados en la Tabla 4.2 demuestran el rendimiento sobresaliente de los clasificadores bajo diversas configuraciones de hiperparámetros y elecciones de características. Es particularmente destacable que los clasificadores como RF, HIVECOTE2, SVC y KNN, utilizando tanto características

4.2. RESULTADOS

57

crudas como DWT, muestran una clasificación perfecta, alcanzando una exactitud, puntuación F1, precisión, recall y AUC de 1.000. Estos modelos resultan ser especialmente eficaces para la tarea de clasificación de hojas limpias y erosionadas.

Varios clasificadores, como Shapelet, BOSS y TimeSeriesForest, presentan un rendimiento ligeramente inferior, pero siguen manteniendo altas puntuaciones de precisión, con pequeñas reducciones en precisión y recall. Por ejemplo, Shapelet alcanza una precisión de 0.999, lo que indica que es casi tan efectivo como los modelos de mejor rendimiento, aunque es ligeramente menos robusto en comparación con los anteriores.

De manera interesante, ciertos clasificadores como RF con características crudas, y KNN con características crudas, muestran una disminución notable en el rendimiento cuando los hiperparámetros se optimizan mediante BO. Por ejemplo, la precisión de los modelos RF y KNN con características crudas disminuye de 1.000 (cuando se usan parámetros predeterminados) a 0.962 y 0.897, respectivamente. Esto sugiere que la optimización de los parámetros no siempre conduce a mejores resultados en este problema específico.

4.2.2. Problema B: Sana-Todas las rugosidades

Este es un problema binario muy desbalanceado, que considera la pala limpia (clase minoritaria) en una clase y todas las clases con rugosidad aditiva combinadas en un único grupo (clase mayoritaria). El objetivo aquí es determinar si los modelos son capaces de diferenciar entre las cuchillas limpias y las que tienen diferentes cantidades y tipos de rugosidad aditiva. El elevado desbalanceo de clases añade un nivel adicional de complejidad al problema.

Los resultados se presentan en el Cuadro 4.3, donde se observa que los modelos de clasificación muestran un rendimiento variable dependiendo de la com-

4.2. RESULTADOS

58

binación de características y técnicas de optimización utilizadas. Es destacable que, en este caso de desbalanceo de clases, los clasificadores que emplean BO, como SVC, tienen un rendimiento superior, alcanzando una precisión de 0.998 y un AUC de 1.000, lo que indica que son muy efectivos en la detección de la clase minoritaria.

Classifier	Hyperparameters	Features	Accuracy	F1	Precision	Recall	AUC
SVC	BO	Raw	0.998	0.996	0.994	0.999	1.000
ROCKET	Default	Raw	0.983	0.957	0.990	0.932	0.932
Shapelet	Default	Raw	0.990	0.978	0.990	0.967	0.999
Mini ROCKET	Default	Raw	0.979	0.949	0.986	0.921	0.921
HIVECOTE2	Default	Raw	0.975	0.938	0.986	0.906	1.000
RF	BO	DWT	0.976	0.943	0.970	0.924	0.987
SVC	BO	DWT	0.980	0.952	0.967	0.943	0.993
RF	Default	DWT	0.973	0.935	0.965	0.913	0.988
SVC	Default	DWT	0.923	0.765	0.957	0.706	0.978
TimeSeriesForest	Default	Raw	0.921	0.758	0.940	0.701	0.961
KNN	BO	Raw	0.942	0.855	0.916	0.816	0.824
RF	BO	Raw	0.925	0.803	0.896	0.756	0.917
BOSS	Default	Raw	0.890	0.595	0.894	0.577	0.914
KNN DDTW	Default	Raw	0.937	0.844	0.889	0.817	0.817
KNN	BO	DWT	0.936	0.840	0.884	0.812	0.812
KNN	Default	DWT	0.901	0.681	0.871	0.638	0.878
KNN DTW	Default	Raw	0.928	0.814	0.870	0.786	0.786
RF	Default	Raw	0.916	0.767	0.854	0.729	0.927
KNN	Default	Raw	0.873	0.483	0.577	0.509	0.875
SVC	Default	Raw	0.871	0.465	0.436	0.500	0.998

Cuadro 4.3: Resultados obtenidos para el problema B.

La [Tabla 4.3](#) muestra que el rendimiento de los clasificadores varía significativamente con respecto a la precisión y el recall, especialmente debido al desbalanceo de clases. En particular, los clasificadores como ROCKET, Shapelet y Mini ROCKET, aunque muestran un rendimiento general alto (precisión superior al 97 %), tienen una caída notable en recall y AUC, lo que sugiere que su capacidad para identificar la clase minoritaria es algo limitada en comparación con modelos como SVC con BO, que muestran un rendimiento sobresaliente en estas métricas.

Otro hallazgo importante es la efectividad de la técnica BO en algunos modelos. Por ejemplo, el clasificador SVC con BO alcanza la puntuación más alta en precisión (0.998) y recall (0.999), lo que indica que el ajuste de hiperparámetros

4.2. RESULTADOS

59

tros mediante BO mejora significativamente el rendimiento en escenarios de desbalanceo de clases. Por otro lado, los modelos como KNN y BOSSS, incluso con BO, muestran una caída en rendimiento en comparación con sus versiones predeterminadas.

4.2.3. Problema C: Sana - Rugosidad $100\mu 03$ y $100\mu 09$

Este es un problema balanceado de tres clases. La primera clase corresponde a las palas limpias, mientras que las otras dos clases presentan rugosidad aditiva con la misma altura (100μ), pero con diferentes porcentajes de área de cobertura. Específicamente, se consideran dos clases con un 3 % y un 9 % de cobertura, lo que introduce una mayor dificultad al problema, ya que las diferencias en la cantidad de área dañada son mínimas. El objetivo es evaluar el rendimiento de los clasificadores bajo estas condiciones y observar cómo se comportan frente a pequeñas variaciones en los patrones de daño.

En la Tabla 4.4 se muestran los resultados obtenidos para cada clasificador, con diferentes configuraciones de hiperparámetros y características. En general, el clasificador SVC con optimización de hiperparámetros mediante BO y características crudas (Raw) obtuvo los mejores resultados, alcanzando una precisión de 0.917, un F1 de 0.916 y un AUC de 0.959. Esto sugiere que el modelo SVC es particularmente eficaz para capturar las diferencias entre las clases, incluso cuando la rugosidad es apenas perceptible.

Por otro lado, los clasificadores como ROCKET y Shapelet también mostraron un desempeño bastante competitivo, con precisiones de 0.812 y 0.801, respectivamente, cuando se utilizaron características crudas. Sin embargo, el rendimiento de los clasificadores basados en DWT, como SVC y RF, fue inferior al de los modelos con características crudas, con precisiones que varían entre 0.629 y 0.800. Esto indica que, aunque la transformada wavelet puede

4.2. RESULTADOS

60

ser útil en algunos casos, en este contexto específico no mejoró el rendimiento significativamente.

Classifier	Hyperparameters	Features	Accuracy	F1	Precision	Recall	AUC
SVC	BO	Raw	0.917	0.916	0.918	0.916	0.959
SVC	BO	DWT	0.835	0.833	0.839	0.834	0.935
HIVECOTE2	Default	Raw	0.823	0.818	0.822	0.823	0.946
ROCKET	Default	Raw	0.812	0.809	0.816	0.811	0.859
RF	BO	DWT	0.800	0.795	0.804	0.799	0.908
Shapelet	Default	Raw	0.801	0.794	0.800	0.801	0.925
Mini ROCKET	Default	Raw	0.751	0.746	0.756	0.747	0.811
RF	Default	DWT	0.737	0.732	0.740	0.738	0.891
SVC	Default	DWT	0.629	0.632	0.682	0.646	0.836
KNN	BO	DWT	0.581	0.592	0.628	0.580	0.749
KNN	Default	DWT	0.527	0.535	0.616	0.533	0.744
RF	BO	Raw	0.568	0.535	0.583	0.563	0.720
KNN DDTW	Default	Raw	0.547	0.548	0.566	0.545	0.660
TimeSeriesForest	Default	Raw	0.501	0.502	0.525	0.499	0.694
KNN	BO	Raw	0.446	0.465	0.513	0.442	0.584
RF	Default	Raw	0.486	0.488	0.510	0.484	0.684
BOSS	Default	Raw	0.458	0.465	0.506	0.458	0.629
KNN DTW	Default	Raw	0.425	0.444	0.495	0.424	0.570
KNN	Default	Raw	0.193	0.156	0.253	0.205	0.482
SVC	Default	Raw	0.242	0.168	0.140	0.282	0.959

Cuadro 4.4: Resultados obtenidos para el problema C.

4.2.4. Problema D: Sana - Rugosidad 100μ03 y 140μ03

Este problema consiste en un conjunto equilibrado de tres clases, donde la primera clase corresponde a las palas limpias, mientras que las otras dos clases presentan rugosidad aditiva con una cobertura del 3 %. La diferencia clave entre estas dos últimas clases radica en la altura de la rugosidad: una tiene una altura de 100 μ03 y la otra de 140 μ03. La tarea consiste en diferenciar entre estas pequeñas diferencias de altura en la rugosidad, lo que plantea un desafío significativo debido a la sutilidad de las variaciones en los patrones. Los resultados obtenidos para este problema se presentan en el Cuadro 4.5.

Al observar los resultados en la tabla, es evidente que el clasificador SVC con optimización de hiperparámetros mediante BO y características crudas (Raw) se destaca notablemente, logrando una precisión de 0.993, un F1 de 0.993 y un AUC de 1.000. Estos resultados indican una capacidad excepcional para distinguir

4.2. RESULTADOS

61

entre las clases, incluso cuando la diferencia en las alturas de la rugosidad es mínima. Este comportamiento subraya la alta capacidad discriminativa de este clasificador en tareas complejas de clasificación.

Otros clasificadores como HIVECOTE2 y Mini ROCKET también presentan resultados destacados, con precisiones de 0.970 y 0.968 respectivamente, lo que demuestra su efectividad para abordar problemas con diferencias sutiles en los datos. Aunque el rendimiento de ROCKET (0.955) y Shapelet (0.932) es ligeramente inferior al de los primeros, siguen siendo modelos competitivos que logran un buen balance entre precisión y AUC, lo que sugiere que son adecuados para este tipo de tareas.

Classifier	Hyperparameters	Features	Accuracy	F1	Precision	Recall	AUC
SVC	BO	Raw	0.993	0.993	0.994	0.993	1.000
HIVECOTE2	Default	Raw	0.970	0.970	0.971	0.972	0.998
Mini ROCKET	Default	Raw	0.968	0.967	0.969	0.968	0.976
ROCKET	Default	Raw	0.955	0.954	0.957	0.957	0.967
Shapelet	Default	Raw	0.932	0.928	0.934	0.929	0.992
SVC	BO	DWT	0.899	0.898	0.907	0.899	0.972
RF	BO	DWT	0.879	0.877	0.890	0.877	0.962
RF	Default	DWT	0.880	0.876	0.887	0.878	0.974
RF	BO	Raw	0.817	0.817	0.838	0.815	0.911
SVC	Default	DWT	0.770	0.764	0.801	0.773	0.931
RF	Default	Raw	0.767	0.767	0.786	0.770	0.904
TimeSeriesForest	Default	Raw	0.746	0.745	0.757	0.752	0.903
KNN DDTW	Default	Raw	0.749	0.742	0.752	0.750	0.813
KNN	BO	Raw	0.705	0.701	0.730	0.705	0.790
KNN	BO	DWT	0.701	0.697	0.719	0.699	0.826
KNN DTW	Default	Raw	0.690	0.683	0.715	0.694	0.770
BOSS	Default	Raw	0.670	0.664	0.685	0.675	0.825
KNN	Default	DWT	0.600	0.586	0.667	0.602	0.791
KNN	Default	Raw	0.374	0.347	0.494	0.382	0.575
SVC	Default	Raw	0.271	0.212	0.212	0.290	0.469

Cuadro 4.5: Resultados obtenidos para el problema D.

4.2.5. Problema E: Limpio-Erosionada-Rugosidades

El Problema E consiste en un desafío de clasificación de tres clases con un fuerte desequilibrio entre ellas. Es una combinación de los Problemas A y C, con la clase mayoritaria representada por la agrupación de rugosidad aditiva. Este problema pone a prueba la capacidad de los clasificadores para manejar el

4.2. RESULTADOS

62

desbalance entre clases mientras abordan las variaciones en las características de las palas limpias, erosionadas y rugosidades. Los resultados obtenidos para este problema se presentan en el Cuadro 4.6.

Al revisar los resultados en la tabla, es notable que el clasificador SVC con optimización de hiperparámetros mediante búsqueda BO y características crudas (Raw) obtiene el mejor rendimiento, alcanzando una precisión de 0.999, un F1 de 0.999 y un AUC de 1.000. Estos resultados destacan la capacidad de SVC para manejar el desequilibrio de clases con una excelente discriminación, lo que sugiere que su capacidad para ajustar los hiperparámetros es crucial para obtener un rendimiento excepcional en problemas desbalanceados.

En cuanto al clasificador RF, se observa un buen desempeño con optimización de hiperparámetros, logrando una precisión de 0.980 con características DWT. Este resultado resalta la capacidad de RF para adaptarse bien a diferentes tipos de características, superando a otros clasificadores como ROCKET y Mini ROCKET que, aunque efectivos, no alcanzan el mismo nivel de precisión.

Por otro lado, el clasificador HIVECOTE2 también presenta un desempeño sobresaliente, con una precisión de 0.975, un F1 de 0.975 y un AUC de 1.000. Aunque su rendimiento es ligeramente inferior al de SVC y RF, sigue siendo muy eficaz, lo que demuestra su capacidad para abordar problemas complejos y desbalanceados de manera efectiva.

4.2.6. Problema F: Problema multiclase completo

El Problema F representa la versión más completa y compleja de la tarea de detección de daños, ya que utiliza una configuración equilibrada con nueve clases diferentes. En este caso, el objetivo es diferenciar entre todos los tipos de daños considerados, lo que presenta un desafío considerable debido a la mayor diversidad de clases en comparación con los problemas anteriores. Los resultados

4.2. RESULTADOS

63

Classifier	Hyperparameters	Features	Accuracy	F1	Precision	Recall	AUC
SVC	BO	Raw	0.999	0.999	0.999	0.999	1.000
HIVECOTE2	Default	Raw	0.975	0.975	0.984	0.970	1.000
RF	BO	DWT	0.980	0.980	0.982	0.978	0.994
ROCKET	Default	Raw	0.972	0.972	0.981	0.967	0.974
Mini ROCKET	Default	Raw	0.976	0.975	0.981	0.973	0.979
Shapelet	Default	Raw	0.974	0.974	0.975	0.975	0.998
RF	Default	DWT	0.968	0.968	0.975	0.964	0.994
SVC	BO	DWT	0.952	0.953	0.953	0.953	0.988
SVC	Default	DWT	0.873	0.856	0.919	0.845	0.972
KNN DDTW	Default	Raw	0.883	0.881	0.893	0.881	0.908
TimeSeriesForest	Default	Raw	0.876	0.868	0.891	0.863	0.951
KNN	BO	DWT	0.870	0.863	0.889	0.853	0.939
KNN DTW	Default	Raw	0.872	0.864	0.884	0.863	0.894
RF	Default	Raw	0.876	0.869	0.883	0.867	0.946
KNN	BO	Raw	0.866	0.859	0.880	0.851	0.886
RF	BO	Raw	0.856	0.850	0.862	0.844	0.952
BOSS	Default	Raw	0.820	0.803	0.840	0.800	0.917
KNN	Default	DWT	0.804	0.785	0.829	0.781	0.916
KNN	Default	Raw	0.676	0.611	0.642	0.631	0.806
SVC	Default	Raw	0.667	0.523	0.531	0.571	0.962

Cuadro 4.6: Resultados obtenidos para el problema E.

obtenidos en este contexto se resumen en el Cuadro 4.7.

Al analizar los resultados, es evidente que el clasificador SVC con optimización de hiperparámetros mediante BO y características crudas (Raw) sobresale entre los demás. Con una precisión de 0.880 y un F1 de 0.880, SVC no solo mantiene un buen balance entre precisión y recall, sino que también obtiene un AUC de 0.986, lo que indica una capacidad destacada para discriminar entre las nueve clases.

Por otro lado, el clasificador Shapelet, aunque no alcanza el rendimiento de SVC, presenta un desempeño decente con una precisión de 0.724 y un AUC de 0.940. Estos resultados sugieren que, aunque Shapelet puede no ser tan potente como otros clasificadores, sigue siendo capaz de capturar patrones útiles en los datos, especialmente para tareas multiclase con características complejas.

El clasificador ROCKET, aunque es generalmente conocido por su eficiencia, muestra un desempeño limitado en este escenario con una precisión de 0.668 y un AUC de 0.815. Este resultado resalta la dificultad de ROCKET para manejar problemas multiclase completos en comparación con otros clasificadores

4.3. DISCUSIÓN

64

como SVC, que ha mostrado ser más efectivo en esta tarea. Además, resulta sorprendente el bajo rendimiento de HIVECOTE2, con una precisión del 67 %, a pesar de ser uno de los métodos más competitivos según el estado-del-arte. Este desempeño fue superado por métodos más simples, como RF, Shapelet y SVC.

Classifier	Hyperparameters	Features	Accuracy	F1	Precision	Recall	AUC
SVC	BO	Raw	0.880	0.880	0.890	0.879	0.986
Shapelet	Default	Raw	0.724	0.724	0.733	0.728	0.940
SVC	BO	DWT	0.688	0.688	0.705	0.690	0.936
ROCKET	Default	Raw	0.668	0.672	0.686	0.672	0.815
RF	BO	DWT	0.677	0.673	0.683	0.678	0.927
HIVECOTE2	Default	Raw	0.653	0.656	0.671	0.657	0.922
Mini ROCKET	Default	Raw	0.640	0.641	0.654	0.645	0.800
RF	Default	DWT	0.630	0.629	0.639	0.635	0.914
KNN DDTW	Default	Raw	0.468	0.491	0.539	0.471	0.702
TimeSeriesForest	Default	Raw	0.475	0.483	0.508	0.480	0.836
RF	BO	Raw	0.467	0.472	0.499	0.469	0.822
KNN	BO	Raw	0.394	0.413	0.475	0.393	0.661
KNN DTW	Default	Raw	0.397	0.418	0.473	0.399	0.662
RF	Default	Raw	0.440	0.448	0.471	0.444	0.816
SVC	Default	DWT	0.404	0.408	0.461	0.420	0.854
KNN	BO	DWT	0.402	0.413	0.447	0.402	0.670
BOSS	Default	Raw	0.396	0.406	0.436	0.399	0.697
KNN	Default	Raw	0.245	0.259	0.403	0.249	0.699
KNN	Default	DWT	0.328	0.340	0.395	0.332	0.720
SVC	Default	Raw	0.144	0.119	0.125	0.166	0.625

Cuadro 4.7: Resultados obtenidos para el problema F.

4.3. Discusión

Este trabajo presenta una metodología para la clasificación multiclase de daños en palas de turbinas eólicas, utilizando un conjunto de datos derivado del proyecto LEES (Maniaci y cols., 2022). El conjunto de datos incluye mediciones detalladas de los coeficientes de presión C_p en 58 puntos distribuidos a lo largo de la superficie de la pala, con el objetivo de diferenciar entre diversas condiciones de las palas: palas sanas, palas erosionadas y palas con distintos niveles de rugosidad aditiva. Se emplearon diversas representaciones de las características extraídas de las series temporales, que incluyen tanto características crudas

4.3. DISCUSIÓN

65

como aquellas obtenidas mediante la DWT. Además, se evaluaron diferentes clasificadores de ML, tanto con configuraciones de hiperparámetros predeterminados como optimizados.

Los resultados experimentales evidencian que el uso de características extraídas mediante la DWT mejora el desempeño del modelo en comparación con el uso de las series temporales crudas, especialmente en el caso del modelo RF en el problema E (véase Cuadro 4.6). No obstante, la diferencia en el rendimiento no fue tan pronunciada en todos los problemas como se anticipaba inicialmente, lo que sugiere que las características crudas, al menos en este caso, contienen suficiente información para que los modelos de clasificación generen resultados competitivos. Este hallazgo resalta la importancia de una adecuada selección de características, pero también subraya que en ciertos contextos, como el análisis de la rugosidad en las palas, los métodos de extracción de características pueden no ser estrictamente necesarios si los datos son suficientemente representativos.

El clasificador de SVC mostró el mejor rendimiento en términos de precisión, recall, F1 y AUC, superando a otros clasificadores como HIVECOTE2 y ROCKET, modelos que son considerados estado-del-arte en el análisis de series temporales. Este resultado resalta la capacidad del SVC para abordar problemas complejos de clasificación multiclase, donde las clases son altamente similares y deben diferenciarse con precisión. La optimización de hiperparámetros mediante búsqueda Bayesiana resultó ser particularmente eficaz en la mejora del rendimiento del clasificador SVC, lo que sugiere que la afinación de los parámetros es crucial para lograr una clasificación precisa en problemas de esta naturaleza.

Por otro lado, el rendimiento del clasificador KNN evidenció dificultades para capturar las complejas interrelaciones entre las variables que afectan la rugosidad y el daño en las palas. Esta limitación sugiere que los métodos basados en distancias no son los más adecuados para problemas en los que las clases son

4.3. DISCUSIÓN

66

sutilmente diferentes, y donde la estructura temporal de los datos juega un papel crucial en la discriminación de las clases.

Además, los resultados obtenidos subrayan la importancia de los ángulos de ataque y los números de Reynolds en la dinámica del flujo sobre la pala, ya que estas variables influyen directamente en la distribución de los coeficientes de presión y, por ende, en la capacidad de los modelos para clasificar correctamente los estados de las palas. Aunque el clasificador SVC mostró un rendimiento superior, los resultados sugieren que aún existe margen para mejorar la clasificación mediante la incorporación de más factores.

Una posible mejora del estudio radica en que los datos utilizados provienen de un conjunto controlado, en el cual los daños y las condiciones de las palas fueron simulados, lo que reduce la complejidad del problema en comparación con situaciones del mundo real, donde los daños pueden ser menos uniformes y más difíciles de modelar. Investigaciones futuras podrían explorar conjuntos de datos más diversos, con condiciones más representativas de la realidad, y centrarse en la integración de la metodología propuesta en sistemas de monitoreo de condiciones de palas en tiempo real.

En conclusión, los resultados obtenidos validan la efectividad de los clasificadores clásicos, como el SVC, para tareas de clasificación multiclase de series temporales en problemas complejos de detección de daños en palas de turbinas eólicas. Estos modelos pueden ofrecer un rendimiento igual o superior al de otros métodos más complejos y costosos en cuanto a tiempo de entrenamiento, como los clasificadores de series temporales avanzados, tales como HIVECOTE2.

Capítulo 5

Conclusiones Generales

Este trabajo presenta metodologías novedosas para la clasificación de problemas complejos utilizando datos en formato de series temporales. En dos estudios de caso distintos, se exploraron enfoques innovadores para resolver problemas de gran relevancia: la identificación de patrones de deambulación asociados con la demencia y la detección de daños en palas de turbinas eólicas. Ambos casos emplearon datos derivados de experimentos controlados y aplicaron tanto características crudas como extraídas mediante DWT, además de evaluar múltiples clasificadores de ML con configuraciones de hiperparámetros predeterminados y optimizados.

Los resultados experimentales demuestran que el uso de características extraídas con DWT tiende a mejorar el rendimiento de los clasificadores, aunque la magnitud de esta mejora varía según el contexto del problema. En el primer caso, el clasificador RF mostró el mejor desempeño para clasificar patrones de deambulación, destacando la importancia de combinaciones específicas de características, como la serie temporal combinada *Coords-Slope*. Por otro lado, en el segundo caso, el clasificador de SVC sobresalió al abordar un problema de

clasificación multiclase de alta complejidad, logrando superar incluso a modelos considerados estado-del-arte en el análisis de series temporales.

Entre las principales limitaciones identificadas, se destaca el uso de datos obtenidos en entornos controlados, lo que reduce la complejidad del problema en comparación con escenarios del mundo real. Esto plantea desafíos para la implementación práctica de las metodologías propuestas, especialmente en términos de privacidad, seguridad y representatividad de los datos.

A pesar de estas limitaciones, los resultados obtenidos validan la efectividad de clasificadores clásicos y de métodos de extracción de características basados en DWT, mostrando que es posible lograr resultados competitivos sin la necesidad de modelos computacionalmente costosos o grandes volúmenes de datos. Este hallazgo es particularmente relevante en contextos donde los recursos son limitados o el tiempo de entrenamiento debe ser minimizado.

Es importante destacar, como señalan los autores en (Togelius y Yannakakis, 2024), que la tendencia actual en la academia es recurrir directamente a técnicas de deep learning. Estas técnicas, aunque producen resultados sobresalientes, a menudo lo hacen sin un análisis profundo de su eficiencia, convirtiéndose en el estado del arte para muchos problemas. Sin embargo, los investigadores no solo compiten entre ellos, sino también con empresas de tecnología que invierten millones de dólares en hardware con capacidades masivas, lo que hace casi imposible igualar su nivel de escalabilidad y resultados. En el contexto de México, donde el presupuesto para CONAHCYT en 2024 fue de 1,289 millones de dólares (Consejo Nacional de Humanidades, 2023), entrenar un modelo como Gemini Ultra, cuyo costo alcanza los 191 millones de dólares (Lynch, 2024), resulta inimaginable para un grupo de investigación local.

Frente a estas limitaciones, la alternativa propuesta en (Togelius y Yannakakis, 2024) y adoptada en este trabajo es el uso de modelos simples y eficientes

dado el problema. Estos modelos, aunque sacrifican algo de precisión, requieren significativamente menos recursos, pueden ejecutarse en entornos limitados y son ideales para implementaciones en *on edge* o sistemas con restricciones de tiempo real. Además, tienen la ventaja de necesitar menos datos para su entrenamiento (Togelius y Yannakakis, 2024), manteniéndose competitivos frente a técnicas de punta.

Futuras investigaciones deberán centrarse en la evaluación de estas metodologías en condiciones más representativas del mundo real, explorando la integración de sistemas de monitoreo en tiempo real y abordando problemas éticos, como la privacidad de los datos. Este trabajo, por tanto, contribuye al avance en la aplicación de técnicas de ML en series temporales, ofreciendo soluciones prácticas y escalables para problemas en los campos de la salud y la energía renovable.

Bibliografía

13

AlSharabi, K., Bin Salamah, Y., Abdurraqueeb, A. M., Aljalal, M., y Alturki, F. A. (2022). EEG signal processing for alzheimer's disorders using discrete wavelet transform and machine learning approaches. *IEEE Access*, 10, 89781-89797. Descargado de <https://doi.org/10.1109/ACCESS.2022.3198988> doi: 10.1109/ACCESS.2022.3198988

14

Alsharef, A., Aggarwal, K., Sonia, Kumar, M., y Mishra, A. (2022, 01 de Nov). Review of ml and automl solutions to forecast time-series data. *Archives of Computational Methods in Engineering*, 29(7), 5297-5311. Descargado de <https://doi.org/10.1007/s11831-022-09765-0> doi: 10.1007/s11831-022-09765-0

Anderson, J. D. (2016). *Fundamentals of aerodynamics* (6th ed.). McGraw-Hill Education. Descargado de <https://www.mheducation.com/cover-images/Jpeg400-high/0073398101jpeg> (Comprehensive resource on aerodynamics, including pressure coefficients and their applications.)

80

AWS. (2024). *Amazon EC2 C5 Instances — Amazon Web Services (AWS)*. Descargado de <https://aws.amazon.com/ec2/instance-types/c5/>

BIBLIOGRAFÍA

71

Bagnall, A., y cols. (2019). Benchmarking time series classification: Functional data vs machine learning approaches. *arXiv preprint*. Descargado de <https://arxiv.org/html/1911.07511> (Accessed: 2024-11-19)

Barua, A., Dong, C., Al-Turjman, F., y Yang, X. (2020). Edge computing-based localization technique to detecting behavior of dementia. *IEEE Access*, 8, 82108-82119. Descargado de <https://doi.org/10.1109/ACCESS.2020.2988935> doi: 10.1109/ACCESS.2020.2988935

Batchelor, G. K. (2000). *An introduction to fluid dynamics* (1st ed.). Cambridge University Press. Descargado de <https://doi.org/10.1017/CB09780511800955> (Covers fundamentals of fluid mechanics, including Reynolds number and its applications.)

Bergstra, J., y Bengio, Y. (2011). Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems*, 24, 2546–2554.

Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., y Cox, D. D. (2015). Hyperopt: a Python library for model selection and hyperparameter optimization. *Computational Science and Discovery*, 8(1), 014008. Descargado de <https://doi.org/10.1088/1749-4699/8/1/014008> doi: 10.1088/1749-4699/8/1/014008

Bergstra, J., Yamins, D., y Cox, D. D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. *Proceedings of the 30th International Conference on Machine Learning*, 115–123.

Berndt, D. J., y Clifford, J. (1994). Using dynamic time warping to find patterns in time series. En *Proceedings of the 3rd international conference on knowledge discovery and data mining* (p. 359-370).

BIBLIOGRAFÍA

72

Best, O., Khan, A., Sharma, S., Collins, K., y Gianni, M. (2024). Leading edge erosion classification in offshore wind turbines using feature extraction and classical machine learning. *Energies*, 17(21). Descargado de <https://www.mdpi.com/1996-1073/17/21/5475> doi: 10.3390/en17215475

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.

Carmona Troyo, J. A. (2024). *Predicción de fallas en palas de aerogeneradores utilizando aprendizaje máquina* (Tesis Doctoral no publicada). Instituto Tecnológico de Tijuana.

Casolaro, A., Capone, V., Iannuzzo, G., y Camastra, F. (2023). Deep learning for time series forecasting: Advances and open problems. *Information*, 14(11). Descargado de <https://www.mdpi.com/2078-2489/14/11/598> doi: 10.3390/info14110598

Chatterjee, S., y Byun, Y.-C. (2023). Highly imbalanced fault classification of wind turbines using data resampling and hybrid ensemble method approach. *Engineering Applications of Artificial Intelligence*, 126, 107104. Descargado de <https://www.sciencedirect.com/science/article/pii/S0952197623012885> doi: <https://doi.org/10.1016/j.engappai.2023.107104>

Chaudhary, A., Gupta, H. P., Shukla, K. K., y Dutta, T. (2020). Sensor signals-based early dementia detection system using travel pattern classification. *IEEE Sensors Journal*, 20(23), 14474–14481. Descargado de <https://doi.org/10.1109/JSEN.2020.3008063> doi: 10.1109/JSEN.2020.3008063

Clissa, L., Lassnig, M., y Rinaldi, L. (2023, octubre). How big is big data? a comprehensive survey of data production, storage, and streaming in science and industry. *Front Big Data*, 6, 1271639.

BIBLIOGRAFÍA

73

Consejo Nacional de Humanidades, C. y. T. C. (2023).

Programa anual de trabajo 2024 consejo nacional de humanidades, ciencias y tecnologías. Descargado de

<https://www.siicyt.gob.mx/index.php/transparencia/informes-conacyt/programa-de-trabajo-1/5>

Cortes, C., y Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.

Dempster, A., Petitjean, F., y Webb, G. I. (2020a, julio). Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5), 1454–1495. Descargado de <http://dx.doi.org/10.1007/s10618-020-00701-z> doi: 10.1007/s10618-020-00701-z

Dempster, A., Petitjean, F., y Webb, G. I. (2020b, julio). Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5), 1454–1495. Descargado de <https://doi.org/10.1007/s10618-020-00701-z>

Dempster, A., Schmidt, D. F., y Webb, G. I. (2021). Mini-rocket: A very fast (almost) deterministic transform for time series classification. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 248–257. doi: 10.1145/3447548.3467231

Deng, H., Underwood, G., Anthony, J., y Myers, C. L. (2013). A time series forest for classification and feature extraction. *Information Sciences*, 239, 142–153. doi: 10.1016/j.ins.2013.02.030

Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., y Keogh, E. (2008). Querying and mining of time series data: experimental comparison of representa-

BIBLIOGRAFÍA

74

tions and distance measures. En *Proceedings of the vldb endowment* (Vol. 1, pp. 1542–1552).

Du, Y., Zhou, S., Jing, X., Peng, Y., Wu, H., y Kwok, N. (2020). Damage detection techniques for wind turbine blades: A review. *Mechanical Systems and Signal Processing*, 141, 106445. Descargado de <https://www.sciencedirect.com/science/article/pii/S0888327019306661> doi: <https://doi.org/10.1016/j.ymssp.2019.106445>

Díaz-Ramírez, A., Miranda-Vega, J. E., Ramos-Rivera, D., Rodríguez, D. A., Flores-Fuentes, W., y Sergiyenko, O. (2022). Time series data processing for classifying wandering patterns in people with dementia. *IEEE Sensors Journal*, 22(11), 10196-10206. Descargado de <https://doi.org/10.1109/JSEN.2021.3123543> doi: 10.1109/JSEN.2021.3123543

Faouzi, J. (2022). Time Series Classification: A review of Algorithms and Implementations. En K. Kotecha (Ed.), *Machine Learning (Emerging Trends and Applications)*. Proud Pen. Descargado de <https://inria.hal.science/hal-03558165>

Fulcher, B. D., y Jones, N. S. (2014). Highly comparative feature-based time-series classification. *IEEE Transactions on Knowledge and Data Engineering*, 26(12), 3026–3037.

Furlanetto, A., Gomes, H. M., y de Almeida, F. S. (2020, 01 de Oct). Design optimization of tapered steel wind turbine towers by qpso algorithm. *International Journal of Steel Structures*, 20(5), 1552-1563. Descargado de <https://doi.org/10.1007/s13296-020-00389-3> doi: 10.1007/s13296-020-00389-3

BIBLIOGRAFÍA

75

Géron, A. (2019). *Hands-on machine learning with scikit-learn, keras, and tensorflow: Concepts, tools, and techniques to build intelligent systems* (2nd ed.). O'Reilly Media.

Global Wind Energy Council. (2024). *Global wind report 2024*. Descargado de <https://gwec.net/global-wind-report-2024/> (Accessed: 2024-10-30)

Goodfellow, I., Bengio, Y., y Courville, A. (2016). *Deep learning*. MIT Press. (<http://www.deeplearningbook.org>)

Herring, R., Dyer, K., Martin, F., y Ward, C. (2019). The increasing importance of leading edge erosion and a review of existing protection solutions. *Renewable and Sustainable Energy Reviews*, 115, 109382. Descargado de <https://www.sciencedirect.com/science/article/pii/S1364032119305908> doi: <https://doi.org/10.1016/j.rser.2019.109382>

Hills, J., Lines, J., Baranauskas, E., Mapp, J., y Bagnall, A. (2014, 01 de Jul). Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery*, 28(4), 851-881. Descargado de <https://doi.org/10.1007/s10618-013-0322-1> doi: 10.1007/s10618-013-0322-1

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90-95. Descargado de <https://doi.org/10.1109/MCSE.2007.55> doi: 10.1109/MCSE.2007.55

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., y Muller, P.-A. (2019, marzo). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4), 917-963. Descargado de <http://dx.doi.org/10.1007/s10618-019-00619-1> doi: 10.1007/s10618-019-00619-1

BIBLIOGRAFÍA

76

Keogh, E., y Pazzani, M. J. (2001). Derivative dynamic time warping. *Proceedings of the 1st SIAM International Conference on Data Mining*, 1-11.

Klein, A., Falkner, S., Bartels, S., Hennig, P., y Hutter, F. (2017). Fast bayesian optimization of machine learning hyperparameters on large datasets. En *Proceedings of the 20th international conference on artificial intelligence and statistics* (pp. 528-536).

Kumar, A., Lau, C. T., Chan, S., Ma, M., y Kearns, W. D. (2016). A unified grid-based wandering pattern detection algorithm. En *2016 38th annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (p. 5401-5404). Descargado de <https://doi.org/10.1109/EMBC.2016.7591948> doi: 10.1109/EMBC.2016.7591948

Li, J., Wang, Z. L., Zhao, H., Gravina, R., Fortino, G., Jiang, Y., y Tang, K. (2017). Networked human motion capture system based on quaternion navigation. *BodyNets International Conference on Body Area Networks*, 45(1). doi: 10.1145/0000000.0000000

Lines, J., Middlehurst, M., y Bagnall, A. (2020). Hive-cote 2.0: a new meta ensemble for time series classification. *Machine Learning*, 109, 2147-2184. doi: 10.1007/s10994-020-05921-9

Lynch, S. (2024). Ai index: State of ai in 13 charts. *Stanford University*.

Mallat, S. (1999). *A wavelet tour of signal processing* (2nd ed.). San Diego, CA: Academic Press. Descargado de <https://doi.org/10.1016/B978-012466606-1/50009-1>

BIBLIOGRAFÍA

77

Maniaci, D. C., MacDonald, H., Paquette, J., y Clarke, R. J. (2022, 12). Leading edge erosion classification system. Descargado de <https://www.osti.gov/biblio/2432094> doi: 10.2172/2432094

46 Mao, P., y Aggarwal, R. (2001). A novel approach to the classification of the transient phenomena in power transformers using combined wavelet transform and neural network. *IEEE Transactions on Power Delivery*, 16(4), 654-660. Descargado de <https://doi.org/10.1109/61.956753> doi: 10.1109/61.956753

23 Martino-Saltzman, D., Blasch, B. B., Morris, R. D., y McNeal, L. W. (1991, 10). Travel Behavior of Nursing Home Residents Perceived as Wanderers and Nonwanderers1. *The Gerontologist*, 31(5), 666-672. Descargado de <https://doi.org/10.1093/geront/31.5.666> doi: 10.1093/geront/31.5.666

91 Mert, A., y Akan, A. (2015). Epileptic seizure detection with wavelet feature extraction and a multiclass support vector machine classifier. *IEEE Transactions on Biomedical Engineering*, 62(2), 371-379.

39 Middlehurst, M., Large, J., Flynn, M., Lines, J., Bostrom, A., y Bagnall, A. (2021a, septiembre). Hive-cote 2.0: a new meta ensemble for time series classification. *Machine Learning*, 110(11-12), 3211-3243. Descargado de <http://dx.doi.org/10.1007/s10994-021-06057-9> doi: 10.1007/s10994-021-06057-9

4 Middlehurst, M., Large, J., Flynn, M., Lines, J., Bostrom, A., y Bagnall, A. (2021b, 01 de Dec). Hive-cote 2.0: a new meta ensemble for time series classification. *Machine Learning*, 110(11), 3211-3243. Descargado de <https://doi.org/10.1007/s10994-021-06057-9> doi: 10.1007/s10994-021-06057-9

BIBLIOGRAFÍA

78

Murugappan, M., Ramachandran, N., y Sazali, Y. (2010). Emotion recognition from eeg using wavelet transform and support vector machine. *International Journal of Cognitive Informatics and Natural Intelligence*, 4(1), 63–75.

Nakaoka, A., Suto, S., Makimoto, K., Yamakawa, M., Shigenobu, K., y Tabushi, K. (2010). Pacing and lapping movements among institutionalized patients with dementia. *American Journal of Alzheimer's Disease & Other Dementias*, 25(2), 167–172. Descargado de <https://doi.org/10.1177/1533317509356688> (PMID: 20107237) doi: 10.1177/1533317509356688

National Institute on Aging. (2022, 12). *What is dementia? symptoms, types, and diagnosis*. Descargado de <https://www.nia.nih.gov/health/what-is-dementia>

Oh, C., Han, S., y Jeong, J. (2020). Time-series data augmentation based on interpolation. *Procedia Computer Science*, 175, 64–71. Descargado de <https://www.sciencedirect.com/science/article/pii/S1877050920316914> (The 17th International Conference on Mobile Systems and Pervasive Computing (MobiSPC),The 15th International Conference on Future Networks and Communications (FNC),The 10th International Conference on Sustainable Energy Information Technology) doi: <https://doi.org/10.1016/j.procs.2020.07.012>

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. Descargado de <http://jmlr.org/papers/v12/pedregosa11a.html>

Schäfer, P., y Höggqvist, M. (2012). SFA: a symbolic fourier approximation and index for similarity search in high dimensional datasets. En *Proceedings of the*

BIBLIOGRAFÍA

79

15th international conference on extending database technology (p. 516–527). New York, NY, USA: Association for Computing Machinery. Descargado de <https://doi.org/10.1145/2247596.2247656>

Schäfer, P. (2015). The boss is concerned with time series classification in the presence of noise. *Data Mining and Knowledge Discovery*, 29(6), 1505-1530. doi: 10.1007/s10618-014-0377-7

Shukla, S. N., y Marlin, B. M. (2019). Interpolation-prediction networks for irregularly sampled time series. *CoRR*, abs/1909.07782. Descargado de <http://arxiv.org/abs/1909.07782>

Noek, J., Larochelle, H., y Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 25, 2951–2959.

Subasi, A. (2007). Eeg signal classification using wavelet feature extraction and a mixture of expert model. *Expert Systems with Applications*, 32(4), 1084–1093.

Togelius, J., y Yannakakis, G. N. (2024). *Choose your weapon: Survival strategies for depressed ai academics*. Descargado de <https://arxiv.org/abs/2304.06035>

Torgo, L. (2010). *Data mining with r: learning with case studies*. Chapman and Hall/CRC.

Ukil, A., y Barlocher, A. (2007). Implementation of discrete wavelet transform for embedded applications using tms320vc5510. En *2007 international symposium on industrial embedded systems* (p. 357-360). Descargado de <https://doi.org/10.1109/SIES.2007.4297361>

BIBLIOGRAFÍA

80

University, W. S. (s.f.). *Casas datasets*. Descargado de <https://casas.wsu.edu/datasets/>

van der Maaten, L., y Hinton, G. E. (2008). Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9, 2579-2605.

Descargado de <http://jmlr.org/papers/v9/vandermaaten08a.html>

Verma, A., Castro, S., Jiang, Z., y Teuwen, J. (2020, 02). Numerical investigation of rain droplet impact on offshore wind turbine blades under different rainfall conditions: A parametric study. *Composite Structures*, 241, 112096. doi: 10.1016/j.compstruct.2020.112096

Vuong, N. K., Chan, S., y Lau, C. T. (2014). Automated detection of wandering patterns in people with dementia. *Gerontechnology*, 12, 127-147. Descargado de <https://api.semanticscholar.org/CorpusID:62610392>

Wan, J., Byrne, C. A., O'Grady, M. J., y O'Hare, G. M. P. (2015). Managing wandering risk in people with dementia. *IEEE Transactions on Human-Machine Systems*, 45(6), 819-823. doi: 10.1109/THMS.2015.2453421

Wang, S.-H., Zhan, T.-M., Chen, Y., Zhang, Y., Yang, M., Lu, H.-M., ... Phillips, P. (2016). Multiple sclerosis detection based on biorthogonal wavelet transform, rbf kernel principal component analysis, and logistic regression. *IEEE Access*, 4, 7567-7576. Descargado de <https://doi.org/10.1109/ACCESS.2016.2620996> doi: 10.1109/ACCESS.2016.2620996

Waskom, M. L. (2021). Seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60), 3021. Descargado de <https://doi.org/10.21105/joss.03021> doi: 10.21105/joss.03021

BIBLIOGRAFÍA

81

Yang, X., Shah, S. A., Ren, A., Zhao, N., Fan, D., Hu, F., ... Tian, J. (2018). Wandering pattern sensing at s-band. *IEEE Journal of Biomedical and Health Informatics*, 22(6), 1863-1870. Descargado de <https://doi.org/10.1109/JBHI.2017.2787595> doi: 10.1109/JBHI.2017.2787595

Yaqoob, I., Hashem, I. A. T., Gani, A., Mokhtar, S., Ahmed, E., Anuar, N. B., y Vasilakos, A. V. (2016). Big data: From beginning to future. *International Journal of Information Management*, 36(6, Part B), 1231-1247. Descargado de <https://www.sciencedirect.com/science/article/pii/S0268401216304753> doi: <https://doi.org/10.1016/j.ijinfomgt.2016.07.009>

Ye, L., y Keogh, E. (2011). Time series shapelets: a new primitive for data mining. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 947-956. doi: 10.1145/1557019.1557122

Zárate, J. E., de los Ángeles Gómez López, M., Troyo, J. A. C., y Trujillo, L. (2022). Analysis and detection of erosion in wind turbine blades. *Mathematical and Computational Applications*. Descargado de <https://api.semanticscholar.org/CorpusID:245970830>