



ESCUELA DE INGENIERÍA DE FUENLABRADA

Grado en Ingeniería Biomédica

Trabajo Fin de Grado

**Segmentación de las fases de la marcha humana y
predicción de sus eventos futuros mediante sensores
inerciales y técnicas de aprendizaje máquina**

por

Sofía Buisson García

Tutora: Sara García de Villa
Co-tutor: David Casillas Pérez

Año académico 2022/2023

Dedicado a Paola

Agradecimientos

En primer lugar, me gustaría agradecer a mis tutores Sara García de Villa y David Casillas Pérez, por su orientación y disponibilidad a lo largo de estos meses. Su dedicación y ayuda han sido fundamentales para el desarrollo de este proyecto.

También quiero agradecer a mis amigas de la universidad: Blanca Andreo, Mounat El Jarmouni, Blanca Liceras, Laura Martínez y Laura Sánchez; por haber compartido lo mejor y lo peor de estos años y haber sido un ejemplo para mí. En particular, agradecer de manera especial a Paola por habernos enseñado tanto.

Además, me gustaría agradecer a mis padres por su constante apoyo y confianza en mí; por acompañarme en esta y en todas las etapas de mi vida.

Por último, quiero agradecer al resto de mi familia y amigos que han estado a mi lado a lo largo de estos años, por ofrecerme la motivación necesaria en los momentos de dificultad.

Abstract

Human gait analysis is a research topic of great importance due to its impact on individuals' autonomy and freedom. In this regard, an alternative to facilitate gait analysis based on the extraction of features from the different phases that compose the gait cycle is proposed.

This Bachelor's Thesis focuses on the application of machine learning models to segment the main gait phases: *a)* heel strike, *b)* foot flat, *c)* push off, and *d)* swing. Furthermore, the remaining time until each gait phase is predicted. These predictions are made from signals recorded by inertial measurement units (IMUs), taking advantage of their low cost and ability to be used in everyday situations.

To carry out this study, signals from the accelerometer and gyroscope of 5 healthy volunteers collected in uncontrolled environments are used. Moreover, the gait phases are labeled using signals obtained from pressure sensors located on the soles of the feet. Based on this labeling, five experiments are conducted: an initial hyperparameter optimization experiment, an individual subject evaluation experiment, a synthetic data generation experiment, a leave-one-subject-out evaluation experiment, and a final experiment for estimating the remaining time until each phase. From this last experiment, gait phase segmentation is also performed. Nine machine learning models are employed for all these experiments, including linear regression, logistic regression, support vector machine (SVM), k-nearest neighbors (KNN), decision tree, random forest (RF), AdaBoost, XGBoost, multilayer perceptron (MLP), and a deep learning model, the long short-term memory (LSTM) neural network.

The results obtained in this research demonstrate the effectiveness of machine learning techniques for gait phase segmentation and prediction of the remaining time until each major event in the gait cycle. Regarding gait phase segmentation, the highest classification metrics are achieved by employing data standardization and balancing techniques with raw data from a single volunteer. Specifically, an accuracy rate exceeding 94% is achieved using the RF model. On the other hand, for predicting the remaining time until the start of gait phases, the most favorable results are obtained in predicting the remaining

time until the swing phase, achieving a mean absolute error (MAE) lower than 60 ms. Specifically, this MAE is achieved by employing the SVM model with raw data from a single volunteer. Based on the predictions of the remaining times until each phase, gait phase segmentation is performed, achieving an accuracy rate of up to 88.3%. Although it did not surpass other accuracy rates obtained, studying the remaining time until major gait events provides valuable additional information to the research developed in this work.

These results suggest that machine learning techniques have great potential for application in various areas, from clinical evaluation to the design of gait-assistive devices. This advancement contributes to the field of human gait study by presenting a comparative analysis of segmentation methods and proposals. It is noteworthy to mention that such a comparative analysis of machine learning classifiers is not found in the state-of-the-art methods, which commonly evaluate only one single segmentation method. Unlike existing approaches in the state of the art, which require the use of multiple sensors, this work demonstrates that precise gait phase segmentation can be achieved using only one IMU per foot. These results open up new possibilities for research in uncontrolled environments, where traditional analysis systems such as cameras have limitations. Furthermore, thanks to the low cost of inertial sensors, the cost of gait-related analyses would be reduced, allowing greater accessibility for a larger number of individuals. In summary, these results lay the groundwork for future investigations and offer opportunities for gait analysis in uncontrolled environments, overcoming the limitations of traditional systems.

Resumen

El estudio de la marcha humana es un campo de investigación de gran importancia debido a su impacto en la autonomía y libertad de las personas. Centrado en este ámbito, se propone una alternativa para facilitar su análisis basado en la extracción de características de las diferentes fases que componen el ciclo de la marcha.

El presente Trabajo de Fin de Grado (TFG) se centra en la aplicación de modelos de aprendizaje automático para segmentar las principales fases de la marcha: *a)* contacto inicial (*Heel Strike*), *b)* apoyo medio (*Foot Flat*), *c)* preoscilación (*Push Off*) y *d)* oscilación (*Swing*). Además, se realiza una predicción del tiempo restante hasta cada una de ellas, lo cual proporciona un valor añadido a la investigación. Estas predicciones se realizan a partir de señales grabadas mediante unidades de medición inercial (IMUs), aprovechando su bajo costo y capacidad de ser utilizadas en situaciones cotidianas.

Para llevar a cabo este TFG, se utilizan las señales del acelerómetro y del giróscopo de 5 voluntarios sanos recogidas en entornos no controlados. Además, se realiza un etiquetado de las fases de la marcha utilizando señales recogidas por sensores de presión situados en las plantas de los pies. A partir del etiquetado se llevan a cabo cinco experimentos: un experimento inicial de optimización de hiperparámetros, un experimento de evaluación en sujetos individuales, un experimento de generación de datos sintéticos, un experimento de evaluación con *leave-one-subject-out* y un último experimento para la estimación de los tiempos restantes hasta cada una de las fases. A partir de este último experimento, también se realiza una segmentación de las fases de la marcha. Para llevar a cabo todos estos experimentos, se emplean nueve modelos de *machine learning*, en concreto regresión lineal, regresión logística, *support vector machine* (SVM), *k-nearest neighbors* (KNN), árbol de decisión, *random forest* (RF), *AdaBoost*, *XGBoost*, perceptrón multicapa (MLP); y un modelo de *deep learning*, la red *long short-term memory* (LSTM).

Los resultados obtenidos en esta investigación demuestran la efectividad de las técnicas de aprendizaje automático para la segmentación de las fases de la marcha y la predicción del tiempo restante hasta cada uno de los principales eventos que componen el ciclo de la marcha. En la tarea de segmentación de las fases de la marcha las mayores métricas de

clasificación se obtienen al emplear las técnicas de estandarización y balanceo de datos con los datos en crudo de un único voluntario. En concreto, se alcanza una tasa de acierto superior al 94 % al emplear el modelo RF. Por otra parte, en la predicción del tiempo restante hasta el comienzo de las fases de la marcha, los resultados más favorables se obtienen en la predicción del tiempo restante hasta la fase de oscilación logrando un error medio absoluto (MAE) inferior a 60 ms. En concreto, este se obtiene al emplear el modelo SVM con los datos en crudo de un único voluntario. A partir de las predicciones de los tiempos restantes hasta cada una de las fases, se realiza una segmentación de las fases que componen el ciclo de la marcha, alcanzando hasta un 88.3 % de tasa de acierto. Aunque no logra superar otras tasas de acierto obtenidas, el estudio del tiempo restante hasta los principales eventos de la marcha proporciona información que añade valor adicional a la investigación realizada. Estos resultados sugieren que estas técnicas tienen un gran potencial para ser aplicadas en diversos ámbitos desde la evaluación clínica hasta el diseño de dispositivos de ayuda a la marcha.

Este avance presenta una contribución en el campo de estudio de la marcha humana al presentar una comparativa de métodos y propuestas de segmentación. Es importante destacar que en el estado del arte previo, no se encuentra esta comparativa, ya que generalmente se evalúa un único método de segmentación sin evaluar más alternativas. A diferencia de los enfoques existentes en el estado del arte, que requieren el uso de múltiples sensores, este trabajo demuestra que una segmentación precisa de las fases de la marcha puede lograrse utilizando únicamente una IMU en cada pie. Esto abre nuevas posibilidades de investigación en entornos no controlados, donde los sistemas de análisis clásicos, como las cámaras presentan limitaciones. Además, gracias al bajo coste que presentan los sensores inerciales, se reduciría el costo de los análisis relacionados con la marcha, permitiendo una mayor accesibilidad para un mayor número de personas. En resumen, estos resultados asientan las bases para futuras investigaciones y ofrecen oportunidades de análisis de la marcha en entornos no controlados, superando las limitaciones de los sistemas tradicionales.

Índice general

Agradecimientos	I
Abstract	II
Resumen	V
Lista de Figuras	XII
Lista de Tablas	XIV
Abreviaciones	XVII
1. Introducción	1
1.1. Objetivos	3
1.2. Estructura de la Memoria	4
2. Estado del arte	6
2.1. Sensores utilizados en el análisis de la marcha	7
2.2. Trabajos previos sobre segmentación y análisis de la marcha con IMUs	8
2.3. Bases de datos relacionadas con la marcha	14
3. Marco Teórico y Fundamentos Matemáticos	16
3.1. Biomecánica de la marcha humana	16
3.2. Unidades de medición inercial (IMUs)	19
3.3. Sensores de fuerza resistivos (FSRs)	20
3.4. Inteligencia artificial	21
3.4.1. Tipos de aprendizaje automático	22

3.5.	Algoritmos de ML	23
3.5.1.	Regresión Lineal	23
3.5.2.	Regresión Logística	24
3.5.3.	Support Vector Machine	26
3.5.4.	K-Nearest Neighbors	27
3.5.5.	Árbol de decisión	28
3.5.6.	Métodos Ensemble	29
3.5.6.1.	Random Forest	31
3.5.6.2.	AdaBoost	31
3.5.6.3.	XGBoost	33
3.5.7.	Perceptrón multicapa	33
3.5.8.	Long Short-Term Memory	35
3.6.	Estandarización y balanceo de datos	36
3.7.	Validación de modelos de aprendizaje automático	38
4.	Metodología y Desarrollo	40
4.1.	Base de Datos	41
4.2.	Descripción de las señales utilizadas	43
4.2.1.	Descripción y etiquetado de las señales FSR	43
4.2.1.1.	Etiquetado de las fases de la marcha	45
4.2.1.2.	Etiquetado de los tiempos	46
4.2.2.	Descripción de las señales de las IMUs	47
4.3.	Preparación de los datos para el entrenamiento	48
4.3.1.	Datos en crudo	50
4.3.2.	Análisis y Extracción de Características	50
4.3.2.1.	Extracción de Características Estadísticas	50
4.3.2.2.	Extracción de Características Polinomiales:	54
4.4.	Descripción de los experimentos	55
4.4.1.	Selección de hiperparámetros y ancho de ventana óptimo	57
4.4.2.	Evaluación individual en cinco voluntarios independientes	61

4.4.3.	Generación de datos sintéticos	62
4.4.4.	Desempeño en nuevos individuos	63
4.4.5.	Predicción de eventos de cambio de fase	63
4.5.	Métricas de evaluación	64
4.5.1.	Métricas de Clasificación	64
4.5.2.	Métricas de Regresión	67
4.6.	Herramientas utilizadas	68
4.6.1.	Python	68
4.6.2.	LaTeX	69
5.	Resultados y Discusión	70
5.1.	Selección de hiperparámetros y ancho de ventana óptimo	70
5.2.	Evaluación individual en 5 voluntarios independientes	75
5.3.	Generación de datos sintéticos	78
5.4.	Desempeño en nuevos individuos	79
5.5.	Predicción de eventos de cambio de fase	81
5.5.1.	Predicción del tiempo restante hasta HS	81
5.5.2.	Predicción del tiempo restante hasta FF	84
5.5.3.	Predicción del tiempo restante hasta PO	86
5.5.4.	Predicción del tiempo restante hasta SW	88
5.5.5.	Clasificación a partir de los resultados obtenidos en regresión	91
5.6.	Comparativa entre los experimentos realizados	92
6.	Conclusiones y Líneas Futuras	94
6.1.	Conclusiones	94
6.2.	Limitaciones y líneas futuras	96
6.3.	Impacto	97
6.3.1.	Impacto Social	97
6.3.2.	Impacto Medioambiental	97
6.3.3.	Impacto Económico	98
6.4.	Lecciones aprendidas	99
	Bibliografía	106

Índice de Figuras

1.1. Ciclo de la marcha.	2
3.1. Fases de la marcha.	17
3.2. Unidad de Medición Inercial o IMU.	20
3.3. Resistencia vs fuerza de un FSR.	21
3.4. Estructura de un FSR.	21
3.5. Ejemplo de clasificador KNN.	27
3.6. Comparativa entre un clasificador KNN para $k = 1$ y $k = 15$	28
3.7. Varianza y sesgo.	30
3.8. Estructura de una neurona.	34
3.9. Principales funciones de activación.	35
3.10. Estructura del perceptrón multicapa.	35
3.11. Arquitectura de la red LSTM.	36
3.12. Funcionamiento del algoritmo SMOTE.	38
3.13. Funcionamiento de la validación cruzada.	38
4.1. Flujo de trabajo.	40
4.2. Colocación de los sensores.	41
4.3. Recorridos realizados por los voluntarios.	42
4.4. Señal de los sensores de presión sin filtrar.	43
4.5. Periodograma de Welch de la señal recogida por los sensores de presión.	44
4.6. Respuesta al impulso del filtro Butterworth.	44
4.7. Etiquetado de la señal filtrada de los sensores de presión.	46
4.8. Tiempos de cambio de fase.	46

4.9. Etiquetado del tiempo restante hasta HS.	47
4.10. Señal del acelerómetro y giróscopo recogida por el dispositivo IMU.	47
4.11. Señal del giróscopo recogida por el dispositivo IMU.	48
4.12. Etiquetado de la señal muestreada de los sensores de presión.	49
4.13. Etiquetado de los tiempos para la tarea de regresión.	49
4.14. <i>Boxplot</i> de la media de la norma de las señales de tres ejes de aceleración lineal y velocidad angular.	51
4.15. <i>Boxplot</i> de la desviación típica de la norma de las señales de tres ejes de aceleración lineal y velocidad angular.	52
4.16. <i>Boxplot</i> del máximo de la norma de las señales de tres ejes de aceleración lineal y velocidad angular.	52
4.17. <i>Boxplot</i> del mínimo de la norma de las señales de tres ejes de aceleración lineal y velocidad angular.	53
4.18. <i>Boxplot</i> de la mediana de la norma de las señales de tres ejes de aceleración lineal y velocidad angular.	53
4.19. <i>Boxplot</i> de los coeficientes obtenidos con <i>Polyfit</i>	55
4.20. Flujo de trabajo del experimento HV.	57
4.21. Determinación del valor de k mediante validación cruzada.	58
4.22. Determinación del valor de la máxima profundidad del árbol mediante validación cruzada.	59
4.23. Determinación del número de árboles mediante validación cruzada.	59
4.24. Determinación del número de neuronas ocultas mediante validación cruzada.	60
4.25. Estructura de la red LSTM para clasificación	60
4.26. Ejemplo de entrenamiento de la red LSTM para clasificación.	61
4.27. Número de muestras de cada fase de la marcha antes y después de aplicar SMOTE.	62
5.1. <i>Precision</i> , <i>Recall</i> y <i>F1-score</i> de los modelos evaluados para identificar cada una de las fases de la marcha.	72
5.2. Métricas de clasificación en función del ancho de ventana.	75
5.3. Matrices de confusión obtenidas con RF, <i>XGBoost</i> y SVC en el Experimento NI.	80
5.4. Resultados de la predicción del tiempo restante hasta HS.	83

5.5.	Resultados de la predicción del tiempo restante hasta FF.	85
5.6.	Resultados de la predicción del tiempo restante hasta PO.	87
5.7.	Resultados de la predicción del tiempo restante hasta SW.	89
5.8.	Representación de las métricas de regresión para cada uno de los modelos en cada una de las fases.	90
5.9.	Matriz de confusión del experimento R.	91

Índice de Tablas

2.1. Características, ventajas y desventajas de los datos obtenidos mediante señales EMG, FSR e IMU.	7
2.2. Resumen del estado del arte.	8
2.3. Resumen del estado del arte 2.	9
3.1. Resumen de los músculos involucrados en la marcha humana.	19
3.2. Tabla comparativa de los modelos <i>bagging</i> y <i>boosting</i>	31
4.1. Información antropométrica de los participantes.	41
4.2. Resumen de los experimentos realizados.	56
4.3. Matriz de confusión binaria.	65
4.4. Matriz de confusión multiclase.	66
5.1. Resultados del Experimento HV con los datos en crudo.	71
5.2. Resultados del Experimento HV con extracción de características estadísticas.	73
5.3. Resultados del Experimento HV con extracción de características polinomiales.	74
5.4. Resultados del Experimento VI con los datos del voluntario 1.	76
5.5. Resultados del Experimento VI con los datos del voluntario 3.	76
5.6. Resultados del Experimento VI con los datos del voluntario 4.	76
5.7. Resultados del Experimento VI con los datos del voluntario 5.	77
5.8. Resultados del Experimento VI con los datos del voluntario 8.	77
5.9. Media de los resultados obtenidos en el Experimento VI.	78
5.10. Resultados del Experimento DS.	78
5.11. Resultados del Experimento NI.	79
5.12. Resultados del Experimento R para predecir el tiempo restante hasta HS.	81

5.13. Resultados del Experimento R para predecir el tiempo restante hasta FF. .	84
5.14. Resultados del Experimento R para predecir el tiempo restante hasta PO. .	86
5.15. Resultados del Experimento R para predecir el tiempo restante hasta SW. .	88
5.16. Resultados del Experimento PE para clasificar las fases de la marcha. . . .	91
6.1. Metas específicas de los ODS que se adecuan a este TFG.	99

Abreviaciones

ANN Artificial Neural Network (red neuronal artificial). 9–12

AUC Area Under the ROC Curve. 58, 59

BD Base de Datos. 14, 15, 41–43

CNN Convolutional Neural Network (red neuronal convolucional). 8, 11

DCNN Deep Convolutional Neural Network. 12

DL Deep Learning. 2, 8, 22, 99

DM-CNN Deep Memory Convolutional Neural Network. 9, 13

dps degrees per second (grados por segundo). 47

ECU Electronic Control Unit (unidad de control electrónico). 42

EMG señales electromiográficas. 7, 8, 10

FF Foot Flat. 17, 18, 45, 46, 48, 49, 51, 53, 62, 64, 66, 80, 81, 84–87, 91, 92, V

FIR Finite Impulse Response (respuesta finita al impulso). 10

FN False Negative (falso negativo). 65

FP False Positive (falso positivo). 65

FSR Force Sensitive Resistors (sensor de fuerza resistivo). 2, 3, 7–9, 11, 15, 16, 20, 21, 41–43, 45, 97

GNN Graph Neural Network (red neuronal gráfica). 9, 12

HO Heel Off. 18, 46

HS Heel Strike. 17, 18, 45–49, 51, 53, 55, 62, 64, 66, 72, 80, 81, 83–87, 90–92, V

IA Inteligencia Artificial. 6, 16, 21, 22, 37, 98

IMU Inertial Measurement Unit (unidad de medición inercial). 2, 3, 7–12, 14, 16, 19, 20, 41–43, 61, 97, 99, II, V, VI

KNN K Nearest Neighbors. 2–4, 27, 28, 56, 58, 71, 72, 78, 79, 82–88, 90–92, 95, II, V

LMT Logistic Model Tree (árbol de decisión logístico). 9

LR Logistic Regression (regresión logística). 9

LSTM Long Short-Term Memory. 3, 4, 12, 35, 56, 60, 61, 71–73, 78, 79, 95, 97, II, V

MAE Mean Absolute Error (error absoluto medio). 67, 68, 82, 84–89, VI

ML Machine Learning. 2–4, 6, 8, 9, 12, 15, 21–23, 48, 50, 56, 57, 61, 62, 68, 72, 99

MLP MultiLayer Perceptron (perceptrón multicapa). 2–4, 10, 12, 33, 35, 56, 59, 79, II, V

ms milisegundos. 8, 11, 81

msDTW multi-dimensional subsequence Dynamic Time Warping. 14

MSE Mean Square Error (error cuadrático medio). 67

ODS Objetivos de Desarrollo Sostenible. 98

ONU Organización de Naciones Unidas. 98

PCA Principal Component Analysis (análisis de componentes principales). 13

PO Push Off. 17, 18, 45, 46, 48, 49, 51, 62, 64, 66, 72, 81, 86, 87, 91, 92, V

ReLU Rectified Linear Unit. 34, 35

RF Random Forest. 3, 4, 9, 11–13, 30, 31, 56, 59, 71–74, 78–80, 82–86, 88, 90, 92, 93, 95, II, V, VI

RMSE Root Mean Square Error (raíz del error cuadrático medio). 67, 68, 82, 84–89, 95

RNN Recurrent Neural Network (red neuronal recurrente). 8, 11, 35

SMOTE Synthetic Minority Over-sampling Technique. 37, 56, 62, 63, 78

SSE Sum Square Error. 24, 29

SVC Support Vector Classifier. 26, 56, 71, 78, 80, 81, 92, 95

SVM Support Vector Machine (máquina de vector soporte). 2–4, 9, 10, 12, 26, 58, 72, 93, II, V, VI

SVR Support Vector Regressor. 26, 56, 82–86, 88, 90–92, 95

SW Swing. 18, 45–49, 51, 53, 64, 66, 72, 80, 81, 88, 90–92, 95, V

TN True Negative (verdadero negativo). 65

TO Toe Off. 18, 46

TP True Positive (verdadero positivo). 65

TS Toe Strike. 18, 46

TSVC Transition Sequence Verification and Correction. 9, 12

VWI-DNN Voted-Weighted Integrated Neural Network. 9, 13

Capítulo 1

Introducción

La marcha humana es el movimiento realizado por los seres humanos con el fin de desplazarse [1]. Se trata de una habilidad fundamental ya que permite al ser humano interactuar con el entorno. Además, la marcha proporciona al ser humano autonomía, posibilidad de huir de los peligros, movilidad e incluso bienestar emocional. Sin la marcha, el ser humano pierde su autonomía y se convierte en dependiente de los otros, perdiendo así parte de su libertad.

En ocasiones, esta libertad puede verse comprometida por diversas patologías o graves traumatismos [2–5]. Algunas de las principales patologías que afectan a la marcha son la enfermedad de Parkinson [6], el accidente cerebrovascular [7], la osteoartritis [8] o las lesiones de médula espinal [9]. Todas estas tienen en común la limitación de libertad de movimientos en los miembros inferiores, impidiendo así un patrón de marcha normal en los pacientes.

Tanto para personas sanas, como para personas que presentan alguna patología, el estudio de la marcha es de gran utilidad. Uno de los principales objetivos del análisis de la marcha es la obtención de información sobre la actividad neuromuscular y la actividad biomecánica de los miembros inferiores [10]. Son múltiples los métodos utilizados para el estudio de la marcha. Algunos se basan simplemente en la observación de la marcha por parte del profesional de la salud [10]; otros utilizan algoritmos de seguimiento del movimiento mediante el uso de cámaras [11]; y otros emplean sensores portables para realizar un análisis cinemático del movimiento de las extremidades inferiores durante la marcha [12,13]. La evaluación del ciclo de la marcha (vea la Figura 1.1), y por tanto de las fases que lo componen es un aspecto importante en este tipo de estudios. Gracias a esto se logra la identificación precoz de ciertas patologías y la prevención de posibles anomalías futuras [14,15].

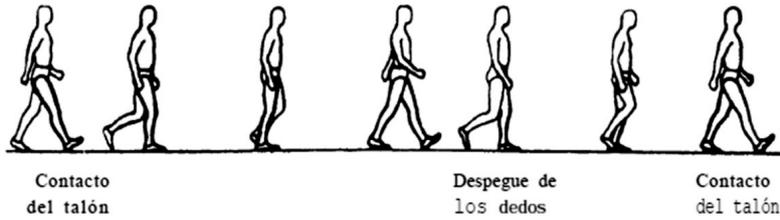


Figura 1.1: Representación del ciclo de la marcha [16].

Otra de las aplicaciones del estudio de la marcha es el diseño de dispositivos de ayuda a la rehabilitación o asistencia a la marcha [17]. Para el diseño de dichos dispositivos es importante el conocimiento del ciclo de la marcha del paciente, así como los eventos asociados a este. Un ejemplo de esto son las prótesis que hacen uso de estimulación eléctrica con el fin de mejorar el patrón de marcha [18]. Para este tipo de dispositivos, es fundamental el entendimiento de los diferentes eventos que componen la marcha, así como la estimación del instante temporal en el que se darán dichos eventos.

La evaluación de la marcha mediante métodos no dependientes de un entorno de laboratorio es especialmente interesante. De esta forma, las conclusiones extraídas pueden ser aplicables a entornos no controlados. Estos métodos suelen utilizar sensores para recoger medidas que permitan realizar un seguimiento y un estudio del ciclo de la marcha normal y patológica [19]. Algunos de los sensores más comunes para el estudio de la marcha en entornos no controlados son los sensores de fuerza resistiva (FSRs) [20] y las unidades de medición inercial (IMUs) [21]. Estos últimos pueden estar colocados en diferentes posiciones, siendo lo más habitual las articulaciones de los miembros inferiores, en concreto cadera, rodilla o tobillo [22]. Los sensores inerciales permiten realizar un análisis cinemático de la marcha del cual se pueden obtener parámetros espaciotemporales relevantes. Sin embargo, la segmentación precisa de las fases de la marcha empleando únicamente estos sensores no es trivial. Por otro lado, los FSRs son útiles para estimar las fases de la marcha, pero no permiten obtener otros parámetros relevantes como los ángulos del pie. Por lo tanto, un análisis basado únicamente en IMUs reduce la necesidad de utilizar múltiples sensores.

Los datos del movimiento medidos con IMUs se procesan mediante diversos algoritmos de aprendizaje automático, con el objetivo de estudiar las principales fases de la marcha. Estos se pueden dividir según si utilizan *machine learning* (ML) o *deep learning* (DL) para el procesamiento de las señales recogidas. Por ejemplo, el perceptrón multicapa (MLP) [23], *k-nearest neighbors* (KNN) [15], *support vector machine* (SVM) [24] o el árbol de decisión [25] emplean ML, mientras que las redes neuronales [26] utilizan DL. El uso de algoritmos de ML y DL resulta muy relevante en el análisis de la marcha humana puesto que pueden ayudar a identificar patrones y características que permiten segmentar las fases de la marcha. Gracias a estos algoritmos que pueden procesar grandes cantidades

de datos es posible realizar una evaluación más precisa y objetiva de la marcha.

En la literatura científica existen numerosos estudios que utilizan algoritmos de ML y DL para la segmentación de la marcha humana [27]. Sin embargo, no hay un consenso claro sobre cuál es el método óptimo cuando se emplean sensores inerciales. Además, todavía no se ha llevado a cabo una comparativa amplia acerca de la aplicación de diferentes métodos de ML sobre una base de datos común, lo que permitiría determinar la mejor alternativa para la segmentación de la marcha y predicción de eventos asociados a esta.

En este contexto, este Trabajo de Fin de Grado (TFG) tiene como propósito el empleo de diversos métodos de aprendizaje automático para llevar a cabo una segmentación de las fases de la marcha y predicción de eventos asociados a esta. Para esto se emplea una base de datos que contiene señales de voluntarios sanos que caminan en entornos no controlados [28]. Estas señales se recogen a partir de sensores inerciales y FSRs. Gracias a los datos obtenidos a partir de los FSRs es posible extraer un *ground truth* de las fases de la marcha que permita validar los modelos. Por otro lado, las señales grabadas con IMUs son procesadas para a continuación analizarlas empleando diferentes modelos de ML. Con la información extraída se define una tarea de clasificación que implica la segmentación de las fases de la marcha, y una tarea de regresión cuyo objetivo es la predicción del tiempo restante hasta cada una de estas fases. Finalmente, se llevan a cabo cinco experimentos: un experimento inicial de optimización de hiperparámetros, un experimento de evaluación en sujetos individuales, un experimento de generación de datos sintéticos, un experimento de evaluación con *leave-one-subject-out* y un último experimento para la estimación de los tiempos restantes hasta las fases. A partir de este último experimento, también se realiza una segmentación de las fases de la marcha. Los resultados obtenidos en estos experimentos se analizan utilizando métricas de clasificación y de diferencia entre la predicción del tiempo restante a cada una de las fases y el tiempo real.

1.1. Objetivos

Este TFG se basa en el análisis de diferentes alternativas de segmentación de la marcha a partir de sensores inerciales. Por lo tanto los principales objetivos que persigue este TFG son los siguientes:

- Segmentar mediante modelos de aprendizaje máquina de clasificación las principales fases de la marcha a partir de datos provenientes de señales grabadas con IMUs. Los métodos empleados para alcanzar este objetivo son: regresión logística, KNN, árbol de decisión, *random forest* (RF), *XGBoost*, *AdaBoost*, MLP, *long short-term memory* (LSTM) y SVM.

- Predecir mediante modelos de ML de regresión el tiempo restante desde un momento puntual hasta cada una de las fases principales de la marcha. Para esto se utilizan los siguientes modelos: regresión lineal, KNN, árbol de decisión, RF, *XGBoost*, *AdaBoost*, MLP, LSTM y SVM.
- Segmentar las fases de la marcha a partir de los tiempos predichos por los algoritmos de regresión.
- Realizar un estudio comparativo de los modelos de clasificación y regresión empleados.

1.2. Estructura de la Memoria

La memoria de este TFG se ha estructurado en seis capítulos divididos en dos grandes bloques. El primer bloque consta de los Capítulos 1, 2 y 3 y proporciona el contexto y las herramientas utilizadas para el desarrollo de este trabajo. El segundo bloque está compuesto por los Capítulos 4, 5 y 6 y recoge la metodología seguida, resultados obtenidos y conclusiones de este TFG, respectivamente. A continuación se presenta un resumen de cada uno de los capítulos que componen esta memoria:

- Capítulo 1: *Introducción*. Define la motivación y los objetivos de este TFG, enfocado en la segmentación de las fases de la marcha, así como un breve resumen de los puntos tratados en la memoria.
- Capítulo 2: *Estado del arte*. En este capítulo se realiza una revisión de la literatura científica y tecnológica relacionada con el análisis y la segmentación de la marcha.
- Capítulo 3: *Marco Teórico y Fundamentos Matemáticos*. Este capítulo tiene como objetivo establecer los fundamentos de la marcha humana, explicar los conceptos relacionados con los sensores inerciales y las medidas que estos proporcionan, y presentar las bases matemáticas de los algoritmos empleados en este trabajo.
- Capítulo 4: *Metodología y Desarrollo*. En este capítulo se presenta tanto la base de datos utilizada como el preprocesamiento de la misma. Además, se detalla la configuración de los modelos utilizados así como la determinación de sus hiperparámetros. Por último, se definen cada uno de los experimentos realizados y las métricas empleadas para su evaluación.
- Capítulo 5: *Resultados*. Recoge los resultados obtenidos de cada uno de los experimentos descritos en el Capítulo 4, así como un análisis y discusión de estos.

- Capítulo 6: *Conclusiones y Lineas Futuras*. En este capítulo se recogen las conclusiones obtenidas, el impacto del estudio y las principales limitaciones así como las futuras líneas de investigación.

Capítulo 2

Estado del arte

Las técnicas de aprendizaje automático se han empleado desde hace varias décadas para establecer relaciones entre diferentes magnitudes físicas utilizando grandes bases de datos [29]. Son múltiples las aplicaciones de estos nuevos algoritmos, algunas de ellas relacionadas con el estudio de la salud en las personas, como por ejemplo el análisis de la marcha [13, 23].

Los algoritmos de ML pueden procesar grandes cantidades de datos recogidos de sensores, y extraer de ellos características relevantes como la longitud del paso, cadencia, velocidad o simetría del movimiento [30]. Además, gracias a la inteligencia artificial (IA) se pueden identificar patrones que no son fácilmente reconocibles a simple vista o mediante métodos convencionales.

La identificación de estos patrones mediante algoritmos de aprendizaje automático permite realizar un análisis de la marcha que es de gran utilidad para los profesionales de la salud. Esto ayuda al diagnóstico de patologías relacionadas con los miembros inferiores o enfermedades neurológicas [15]. Por otra parte, los exoesqueletos de miembros inferiores proveen asistencia a los movimientos de las personas con dificultades en la marcha, moviéndose de manera sincrónica con el paso del paciente. El estudio de la marcha también cobra un papel fundamental en el diseño de estos dispositivos [25].

En este capítulo se proporciona una revisión de los conocimientos existentes en el análisis y segmentación de la marcha, incluyendo los principales enfoques y metodologías. La Sección 2.1 define los principales sensores utilizados en el análisis de la marcha; la Sección 2.2 resume la literatura centrada en este tema en los últimos años; y la Sección 2.3 describe algunas bases de datos relacionadas con el estudio de la biomecánica de la marcha humana.

2.1. Sensores utilizados en el análisis de la marcha

A lo largo de los últimos años se han desarrollado múltiples algoritmos para el análisis de la marcha [31,32]. Los datos utilizados por estos algoritmos pueden provenir de diversos tipos de sensores clasificados como sensores directos e indirectos. Los sensores directos son aquellos que requieren de una conexión directa con el usuario, como por ejemplo las señales electromiográficas (EMG) [23]. Los sensores indirectos se caracterizan por la utilización de medidas cinemáticas y cinéticas. Las IMUs o los FSRs forman parte de los sensores indirectos [26]. La Tabla 2.1 muestra las características, ventajas e inconvenientes de estos sensores [33].

Tabla 2.1: Características, ventajas y desventajas de los datos obtenidos mediante señales EMG, FSR e IMU.

	Señales EMG	FSR	IMU
Características	Registro de la actividad eléctrica muscular.	Miden la presión aplicada.	Mediante acelerómetros, giróscopo y magnetómetros aporta información de la aceleración y velocidad de giro.
Ventajas	<ul style="list-style-type: none"> • Análisis de las funciones individuales del músculo. • Permite el estudio de músculos superficiales y profundos. 	<ul style="list-style-type: none"> • Facilidad de uso. • Pueden ser utilizados para diferentes tamaños de pie. 	<ul style="list-style-type: none"> • Sensores pequeños, ligeros y baratos. • No requieren que el estudio se lleve a cabo en un entorno controlado.
Desventajas	<ul style="list-style-type: none"> • Necesitan ser combinados con otros sistemas para el análisis de la marcha. • Las señales presentan gran cantidad de ruido. • Las señales requieren amplificación y filtrado. • Las señales pueden ser afectadas por medicación que relaje el sistema nervioso. • Difícil interpretación de los datos. 	<ul style="list-style-type: none"> • Necesitan ser combinados con otros sistemas para el análisis de la marcha. • Respuesta lenta, lo que puede generar interpretaciones incorrectas. 	<ul style="list-style-type: none"> • Complejos para estimar los parámetros necesarios para el análisis de la marcha.

Las ventajas referidas al pequeño tamaño y peso de las IMUs (ver Tabla 2.1) las hacen idóneas para ser portables. Además, no requieren que el estudio se realice en un entorno controlado. Sin embargo, no ofrecen información de la actividad muscular, para lo que es útil la señal EMG. Por su parte, los FSRs miden la presión aplicada, pero necesitan ser combinados con más sistemas para realizar un análisis completo de la marcha.

Este TFG se centra en las señales recogidas por los sensores inerciales debido a que se tratan de unos sensores baratos y ligeros, y no tan dependientes de su localización como los electrodos que recogen las señales EMG. Además, permiten obtener los ángulos de los segmentos o las articulaciones, lo que les otorga un valor añadido.

2.2. Trabajos previos sobre segmentación y análisis de la marcha con IMUs

Para desarrollar el estado del arte de este TFG se han realizado búsquedas en la herramienta *Google Scholar*, con las siguientes palabras claves: *inertial measurement unit*, *machine learning*, *deep learning*, IMU, ML, DL, *gait analysis*, *gait phase classification* y *gait segmentation*. Para la selección de los artículos se han elegido aquellos publicados en los últimos cinco años y que han recibido un alto número de citas.

La Tabla 2.2 y la Tabla 2.3 muestran un resumen de los artículos del estado del arte que se van a describir a continuación. Estas tablas categorizan los artículos por objetivos, sensor utilizado, métodos de ML aplicados y resultados obtenidos.

Tabla 2.2: Resumen del estado del arte.

Artículo	Objetivo	Señal	Métodos ML	Resultados
<i>A deep learning approach to emg-based classification of gait phases during level-ground walking</i> [23].	Clasificar fase de oscilación y fase de apoyo.	EMG	MLP	Tasa de acierto: <ul style="list-style-type: none"> Entrenamiento: 94.4 %. Test: 93.4 %.
<i>Recognition of gait phases with a single knee electrogoniometer: A deep learning approach</i> [13].	Realizar un reconocimiento de las fases de la marcha utilizando un único electrogoniómetro por pierna.	EMG+ IMU	MLP	Tasa de acierto: <ul style="list-style-type: none"> Datos de EMG + IMU: 95.6 %. Datos de IMU: 90.9 %.
<i>Gait phase detection for normal and abnormal gaits using IMU</i> [31].	Detectar fases de la marcha y determinar aquellos pacientes que presentan un paso anormal.	IMU+ FSR	-	Error medio: 30 ms. Tasa de acierto: 94.4 % en test.
<i>Wearables-based multi-task gait and activity segmentation using recurrent neural networks</i> [32].	Clasificación automática de actividades (andar, correr, montar en bicicleta...) y segmentación de las fases de la marcha.	IMU	Red neuronal convolucional (CNN) combinada con red neuronal recurrente (RNN).	Tasa de acierto: <ul style="list-style-type: none"> En clasificación de actividades 92 %. En segmentación de las fases de la marcha: 98 %.

Tabla 2.3: Resumen del estado del arte 2.

Artículo	Objetivo	Señal	Métodos ML	Resultados
<i>Machine learning based adaptive gait phase estimation using inertial measurement sensors</i> [34].	Segmentación de las fases de la marcha.	IMU	<ul style="list-style-type: none"> • <i>Logistic regression</i> (LR). • RF. • Redes neuronales artificiales (ANN). 	Tasa de acierto en RF en test: <ul style="list-style-type: none"> • 98.4 % marcha. • 98.45 % carrera. • 99.15 % subida de escaleras. • 99 % bajada de escaleras. • 99.63 % cuclillas.
<i>Gaitphase classification for a lower limb exoskeleton system based on a graph convolutional network model</i> [35].	Desarrollar un modelo de aprendizaje semi supervisado para clasificar las fases de la marcha con el fin de controlar un exoesqueleto.	Goniómetros + FSR	Red neuronal gráfica (GNN)	Tasa de acierto: 96 %.
<i>Continuous locomotion mode recognition and gait phase estimation based on a shank-mounted imu with artificial neural networks</i> [26].	Reconocimiento de modos de locomoción y fases de la marcha.	IMU+ FSR	ANN.	Tasa de acierto en test: <ul style="list-style-type: none"> • 97 % entornos controlados. • 95 % ambiente real.
<i>Design, development, and evaluation of a local sensor-based gait phase recognition system using a logistic model decision tree for orthosis-control</i> [25].	Segmentación de las fases de la marcha.	IMU	<ul style="list-style-type: none"> • Árbol de decisión logístico (LMT). • <i>Transition sequence verification and correction</i> (TSVC). 	Tasa de acierto: <ul style="list-style-type: none"> • LMT: 98.38 % en entrenamiento y 90.6 % en test. • TSVC: 98.72 % en entrenamiento y 98.61 % en test.
<i>Gait phase detection from thigh kinematics using machine learning techniques</i> [24].	Segmentación de las fases de la marcha utilizando únicamente información de la cinemática del muslo y el ángulo de la rodilla.	IMU	<ul style="list-style-type: none"> • Árbol de decisión • RF. • MLP. • SVM. 	<ul style="list-style-type: none"> • Mejor tasa de acierto: 98 % en test con RF. • Mejor error cuadrático medio: 0.014 con el árbol de decisión.
<i>Recognition of the gait phase based on new deep learning algorithm using multisensor information fusion</i> [36].	Segmentación de las fases de la marcha.	IMU+ FSR	<i>Deep memory convolutional neural network</i> (DM-CNN).	Tasa de acierto: 97.1 %.
<i>Walking gait phase detection based on acceleration signals using voting-weighted integrated neural network</i> [37].	Segmentación de las fases de la marcha.	IMU	<i>Voting - weighted integrated neural network</i> (VWI-DNN).	Tasa de acierto: 99.5 %.
<i>Adaptive algorithm for gait segmentation using a single IMU in the thigh pocket.</i> [38].	Segmentación de las fases de la marcha.	IMU	<i>Hidden-Markov.</i>	<i>F1-score</i> : <ul style="list-style-type: none"> • 75.57 % en evaluación entre sujetos. • 95.5 % en evaluación intra sujetos.

En general, tal y como se puede observar en la Tabla 2.2 y en la Tabla 2.3, el estudio de la marcha centrado en su segmentación es un tema bastante investigado en los últimos años. Además, estos estudios utilizan diferentes tipos de señales y modelos de aprendizaje automático, entre los que se encuentran redes neuronales, árboles de decisión o SVM; incluyendo en ocasiones la combinación de varios de estos. Generalmente las métricas utilizadas son la tasa de acierto o *Accuracy* (*Acc*), y el *F1-score* (*F1*), obteniendo resultados que superan el 90 % de predicciones correctas.

En relación con la segmentación de la marcha, los resultados obtenidos con otros tipos de sensores no difieren mucho a los de las IMUs, que son los sensores que se utilizan en este TFG. A continuación, se muestran algunos estudios relacionados con la utilización de señales EMG para el análisis de la marcha.

En [23] se recogen señales EMG de 23 voluntarios sanos mientras caminan en condiciones normales. El objetivo de este estudio es proponer una red neuronal artificial (ANN) que clasifique la fase de oscilación y la fase de apoyo. Cada uno de los voluntarios es dotado de tres sensores de presión y cuatro electrodos que recogen la señal EMG. Una vez recogidas estas señales, se les aplica un filtro paso alto, seguido de un filtro de respuesta finita al impulso (FIR), para terminar de nuevo con un filtro paso alto. Estos datos se normalizan y se divide cada una de las señales en ventanas de 20 muestras. Se utiliza un modelo MLP, el cual se evalúa con métricas como la tasa de acierto, precisión, sensibilidad y *F1-score*. Finalmente se obtiene una tasa de acierto de un 94.4 % en entrenamiento y un 93.4 % en test.

Algunos estudios combinan el uso de señales EMG con medidas inerciales. En [13] se define un estudio donde los voluntarios son equipados con un electrogoniómetro de rodilla, tres sensores en el pie y 4 electrodos que recogen la señal EMG. Estos datos se procesan de la siguiente manera: por una parte, los datos de los ángulos de la rodilla recogidos por el electrogoniómetro se les aplica un filtro paso bajo para a continuación normalizarlos; por otra parte, las señales EMG se amplifican para aplicarles un filtro paso alto, uno paso bajo y por último un filtro paso bajo Butterworth de orden dos. A partir de aquí, se estudian dos posibles aproximaciones para realizar un reconocimiento de las fases de la marcha: utilizando únicamente los datos obtenidos del electrogoniómetro, o utilizando además los datos obtenidos de las señales EMG. Se les aplica una arquitectura de red MLP con tres capas ocultas. Para evaluar los resultados se utilizan métricas como precisión, sensibilidad o *F1-score*. En cuanto a la tasa de acierto, se obtiene un 95.6 % en el caso de la utilización de datos del electrogoniómetro y de las señales EMG, y un 90.9 % en el caso de que se utilice únicamente los datos del electrogoniómetro. El objetivo del estudio es demostrar que se puede simplificar el análisis de la marcha utilizando un único electrogoniómetro por pierna, así que una tasa de acierto de mas del 90 % ya se considera útil.

En el estado del arte se encuentran también artículos que recogen el estudio de señales

grabadas con IMUs para la segmentación de las fases de la marcha. En [31], el máximo local, mínimo local y los pasos por cero de la velocidad angular y la aceleración son utilizados para detectar las fases de la marcha y determinar aquellos pacientes que presentan un paso anormal, lo cual puede estar relacionado con alguna patología. Los datos se obtienen a partir de dos sensores inerciales con cuatro FSRs que recogen velocidad angular y aceleración del muslo y de la pierna. Para clasificar las diversas fases de la marcha se estudian las formas de la señal, la velocidad angular y la aceleración. La tasa de acierto media es de un 94.4%, lo que no supone una mejoría con respecto a otros estudios. Sin embargo, se logra obtener un error medio de 30ms, lo que sí supone una mejora en comparación con investigaciones anteriores.

Otra aproximación en el estudio de la biomecánica de los miembros inferiores tiene relación con la clasificación automática de diversas actividades tales como andar, correr, montar en bicicleta o saltar. En [32] se propone una red neuronal para la realizar esta clasificación y además lograr la segmentación de las fases de la marcha (detección de fase de apoyo y fase de balanceo) en aquellas actividades relacionadas con la marcha (andar y correr). La idea es combinar una red neuronal convolucional (CNN) para la detección de los límites en la segmentación y una red neuronal recurrente (RNN) que modele la dependencia temporal de los datos. El modelo propuesto consta de una capa convolucional, normalización batch y dos capas recurrentes. Una vez creado el modelo, este se entrena y se postprocesan los resultados, eliminando los pasos demasiado largos o demasiado cortos. En este caso utilizan bases de datos de estudios anteriores para entrenar la red, obteniendo resultados de *F1-score* de más del 92% en el caso de clasificación de actividades y más del 98% en la segmentación de las fases de la marcha.

En [34] se lleva a cabo la segmentación de señales medidas con IMUs en diferentes condiciones: estado estacionario, caminar, correr, subir o bajar escaleras y cuclillas. Como entradas se utilizan la aceleración, velocidad angular, ángulo de Euler, aceleración resultante y velocidad angular resultante. Se emplean tres métodos para construir los modelos de detección: regresión logística, RF y ANNs. Se demuestra que el evento más importante para separar las fases es el contacto inicial y despegue (ocurre cuando el pie de referencia deja de tocar el suelo). El modelo que mejores resultados obtiene es RF con unas tasas de acierto de 98.94% en marcha, 98.45% en carrera, 99.15% en subida de escaleras 99%, en bajada de escaleras y 99.63% en cuclillas.

Entre las aplicaciones de la segmentación de las fases de la marcha, se encuentra el control de dispositivos de rehabilitación como exoesqueletos. En [35] se tiene como objetivo desarrollar un modelo de aprendizaje semi supervisado para clasificar las fases de la marcha con el fin de controlar un exoesqueleto mientras el paciente camina en entornos llanos, cuesta arriba y cuesta abajo. Los datos se recogen en tiempo real a partir de dos tipos de sensores: goniómetros y FSRs. Además, con el fin de validar la efectividad

de la recogida de datos a tiempo real, se utiliza un sistema de análisis de movimiento 3D. Para solucionar el problema de clasificación se utiliza una red neuronal gráfica (GNN). En primer lugar, se transfieren los datos al dominio no euclídeo, para luego entrenarlos con el fin de clasificar las fases de la marcha. Finalmente, en los resultados obtenidos se logran tasas de acierto de entorno al 96 % utilizando GNN, lo cual es significativamente mejor que las tasas de acierto obtenidas con otros modelos como LSTM (entorno al 94 %) o *deep convolutional neural network*, DCNN (entorno al 90 %).

También, para mejorar el control de robots portátiles para asistencia en la marcha, se presenta una aproximación para el reconocimiento de diversos modos de locomoción y fases de la marcha basados en ANNs [26]. En este estudio sólo se utilizan las medidas inerciales provenientes del plano sagital (midiéndose aceleración y velocidad angular), obtenidas a partir de dos experimentos. En primer lugar, datos recogidos en un laboratorio, y en segundo lugar datos recogidos en un ambiente no controlado. Para esto último se desarrolla un dispositivo *force resistor IMU* (FRIMU2) que recoge información de sensores inerciales y sensores de presión. Se busca minimizar el error cuadrático medio, lográndose mejores resultados con los datos obtenidos en el laboratorio que con los datos obtenidos en la realidad. Como métrica se utiliza la tasa de acierto, obteniéndose valores de entorno a 97 % en los datos de laboratorio y 95 % en los datos recogidos de ambiente real.

El reconocimiento óptimo de las fases de la marcha se ha estudiado además para proporcionar a las órtesis SCKAFO el control suficiente de la rodilla en el momento idóneo [25]. Este estudio logra que se identifiquen 4 fases de la marcha: respuesta a la carga, apoyo terminal, oscilación media y oscilación final. En este artículo, se utiliza un modelo de clasificación de árbol de decisión en señales provenientes de sensores inerciales de 30 participantes. El uso de ML con señales de los sensores inerciales situados en el muslo y en la rodilla permite distinguir las fases de la marcha en distintas condiciones (velocidades o inclinación de la superficie). Además, con el fin de mejorar el desempeño del modelo de árbol de decisión, se utiliza un algoritmo denominado *transition sequence verification and correction* (TSVC). En relación con los resultados, se obtiene una tasa de acierto del 98.38 % en entrenamiento y un 90.6 % en test. Tras aplicar el algoritmo TSVC se consigue una tasa de acierto del 98.72 % en entrenamiento y un 98.61 % en test. Con esto se logra reducir la complejidad del sistema de sensores que se utilizan en otros estudios.

En otro artículo [24] se pretende determinar si la detección de las fases de la marcha puede lograrse utilizando clasificadores de árboles de decisión, RF, MLP o SVM utilizando únicamente información de la cinemática del muslo y el ángulo de la rodilla de 31 pacientes voluntarios. Para la identificación de la fase de la marcha se utilizan IMUs. Se prueban diferentes técnicas de ML, para las cuales se dividen los datos en entrenamiento y test con validación cruzada. Las métricas utilizadas para este clasificador son la tasa de acierto y el error cuadrático medio, así como la tasa de verdaderos positivos y la tasa de falsos

positivos. La mejor tasa de acierto se consigue con el clasificador RF (98 %) y el menor error cuadrático medio (0.014) se consigue con el árbol de decisión.

También se han propuesto algoritmos novedosos, como *deep memory convolutional neural network* (DM-CNN), para reconocer las fases de la marcha [36]. En primer lugar, se recogen datos de 10 pacientes sanos provenientes de sensores inerciales y sensores de presión plantar (FSR402). A continuación, se realiza un procesamiento de las señales, donde se elimina el ruido, para luego extraer las características necesarias utilizando una DM-CNN. La ventaja de introducir una red de memoria es que al entrenar el modelo, no solo se utiliza la información del momento presente sino que también utiliza información observada en el pasado. Para la evaluación de esta red neuronal se utilizan las siguientes métricas: precisión, sensibilidad y tasa de acierto, obteniéndose un resultado de 97.1 % en esta última.

Otra propuesta para la detección de las fases de la marcha pasa por la creación de un modelo de *Voted-Weighted Integrated Neural Network* (voted-weighted integrated neural network). Este modelo utiliza como entradas las señales de aceleración multilínea de 20 voluntarios de los que se recogen datos de tres sensores inerciales colocados en la pierna [37]. Esta información se sintetiza utilizando el método *principle component analysis* (PCA). El artículo divide la marcha humana en cuatro fases: *Heel Strike* (el pie toca el suelo, también llamado contacto inicial), *Foot Flat* (toda la planta del pie está apoyada en el suelo), *Heel Off* (último contacto del talón con el suelo) y *Swing* (fase de balanceo, el pie no está en contacto con el suelo). El objetivo es crear una red neuronal formada por tres subredes que permitan clasificar las fases de la marcha gracias a un sistema de votación. Las métricas que se utilizan para evaluar el modelo son la precisión, sensibilidad, y *F1-score*. Los resultados experimentales demuestran que este método logra mejores tasas de acierto, obteniéndose valores de hasta el 99.5 %. Este modelo genera gran cantidad de parámetros, lo que hace que tenga gran coste computacional, por lo que una posible línea de investigación sería comprimir los parámetros para reducir el tiempo.

Por último, en [38] se centran en la segmentación de 4 fases de la marcha: *Heel Strike*, *Toe Off* (el pie pierde el contacto con el suelo, también llamado contacto final), *Foot Flat* y *Heel Off*. La idea es recoger las señales provenientes de un único sensor inercial situado en el bolsillo (atado al smartphone) que recoge el movimiento. Además de este sensor, se sitúan otros 3 sensores en la pierna del sujeto, para obtener las etiquetas de las fases de la marcha. Se utiliza un modelo de *Hidden-Markov* del que se obtienen unos resultados en la métrica *F1-score* de 75.57% en evaluación entre sujetos (se estudian los primeros 20 segundos de todos los sujetos), y un *F1-score* de 95.5% en evaluación de todos los eventos intra sujetos (se estudia los primeros 20 segundos de un único sujeto), lo que demuestra mejores resultados que estudios anteriores.

2.3. Bases de datos relacionadas con la marcha

Aunque existen diferentes bases de datos relacionadas con la marcha humana, no todas están enfocadas en la segmentación de las diferentes fases de la misma. Con este fin, es fundamental contar con un *ground truth*, es decir, las etiquetas que sirvan como referencia para validar que la segmentación se ha realizado de manera correcta y fiable. En la búsqueda de la base de datos utilizada para esta investigación se destacan las bases de datos descritas a continuación.

En [39] se presenta una base de datos (BD) formada por datos recogidos de sensores inerciales y grabaciones de cámaras con el objetivo de dividir cada zancada en 4 fases según las bases de la biomecánica: empuje (*Push Off*), oscilación (*Swing*), frenada (*Heel Strike*) y apoyo (*Foot Flat*). Esta BD recoge datos de la marcha de 22 sujetos sanos grabados con sensores inerciales, concretamente con un giróscopo de 3 ejes y un magnetómetro de 3 ejes. Todos estos datos se etiquetan manualmente gracias al software ELAN. Además, el trabajo plantea un método de segmentación de la marcha a partir de las señales de las IMUs, basado en *multi-dimensional subsequence Dynamic Time Warping* (msDTW). Por otro lado, como el método anterior depende de un valor umbral, se propone un nuevo método mediante el cual se genera una plantilla a partir de las muestras y se calcula la mínima distancia de cada muestra a esa plantilla localizando los puntos de inicio y fin de la señal y así reduciendo el coste computacional. Para evaluar este modelo se utilizan las métricas de precisión, sensibilidad y *F1-score* obteniéndose resultados de entorno al 80% en esta última, y demostrándose que el nuevo método es mejor que el modelo DTW convencional. Finalmente, esta BD no se utiliza para este TFG, puesto que no se encontraba disponible para su descarga al inicio del desarrollo de este trabajo.

La BD utilizada en este proyecto se detalla en [28], donde se describe la recogida de datos relacionados con la marcha de 20 voluntarios sanos. Estos voluntarios son equipados con un traje de lycra con 17 sensores inerciales alrededor del cuerpo, 8 sensores de presión

por pie, y un rastreador ocular. Cada uno de los sujetos describe tres trayectorias distintas, cada una de las cuales consiste en varios minutos de caminata donde tienen que atravesar rampas, escaleras y pavimentos.

Esta última BD es la seleccionada para el desarrollo de esta investigación, debido a que los sensores FSRs pueden ser empleados para obtener un *ground truth* de las fases de la marcha y porque, entre los sensores inerciales colocados en los voluntarios, uno se encuentra en el pie. La información obtenida por este sensor inercial es la que se utiliza como entrada para los modelos de aprendizaje automático. La posición de este sensor inercial es interesante, puesto permitiría la aplicación de los resultados en nuevas tecnologías como zapatillas inteligentes [40]. Además, otro de los motivos para la selección de esta BD es que se encuentra disponible públicamente para la comunidad científica, y es fácilmente accesible para realizar investigaciones acerca del análisis de la marcha.

Una de las principales limitaciones que se han encontrado en las investigaciones sobre segmentación de las fases de la marcha de los últimos años es la falta de una comparativa rigurosa entre los diferentes métodos y algoritmos utilizados en la literatura. Aunque se han aplicado diferentes técnicas de aprendizaje automático para la segmentación de las fases de la marcha, no se ha realizado una evaluación exhaustiva de las mismas en un conjunto de datos comunes, lo que dificulta la comparación. La elección del método óptimo tiene diversas implicaciones, entre las que destaca el coste computacional asociado a algunos algoritmos de ML.

Otra limitación importante de los métodos de aprendizaje automático utilizados para la segmentación de las fases de la marcha es que se han basado en técnicas de clasificación. Esta aproximación puede provocar errores en la identificación de las fases de la marcha o la inclusión de datos correspondientes a fases diferentes dentro de la fase de estudio. En este sentido, es importante destacar la necesidad de detectar de manera precisa los tiempos de cambio de fase en la marcha. Para abordar esto, se propone la evaluación de métodos de regresión que permitan identificar los instantes temporales de cambio de fase, lo cual brinda información adicional en el análisis de la marcha.

El objetivo de este TFG es la aplicación de diversos algoritmos de ML para realizar la segmentación de las fases de la marcha, incluyendo una comparativa de los mismos. Como valor adicional, se pretende llevar a cabo la predicción del tiempo restante hasta el inicio de cada una de las fases de la marcha, y utilizar estas predicciones para la segmentación de las mismas. De esta manera, se busca determinar la mejor aproximación y plantear también una alternativa de detección de cambios de fase.

Capítulo 3

Marco Teórico y Fundamentos Matemáticos

En este capítulo se definen los conceptos, teorías y herramientas matemáticas que se utilizan en el desarrollo de los experimentos de este TFG. En la Sección 3.1 se definen los principales conceptos relacionados con la marcha y sus fases. Las Secciones 3.2 y 3.3 aportan un resumen del funcionamiento de las IMUs y los FSRs. Por otra parte, la Sección 3.4 proporciona una visión general de la IA, incluyendo los tipos de aprendizaje automático existentes. Además, la Sección 3.5 se centra en los modelos utilizados para la clasificación de las fases de la marcha y la estimación del tiempo hasta los eventos relevantes de marcha. La Sección 3.6 aborda la estandarización y balanceo de datos. Por último, la Sección 3.7 explica la teoría relativa a la validación de modelos de aprendizaje automático.

3.1. Biomecánica de la marcha humana

Durante la marcha se realizan movimientos coordinados con las diferentes estructuras corporales [1,41]. Algunas de las estructuras involucradas en la marcha son las siguientes:

- Huesos de la pierna y del pie: el fémur, la tibia y el peroné aportan soporte y estabilidad durante la marcha. Por su parte, el pie es una estructura de vital importancia para la marcha, puesto que proporciona una base estable y amortiguación durante el impacto contra el suelo.
- Articulaciones de la cadera, rodilla y tobillo: permiten la flexión y extensión de la pierna y el pie durante la marcha.
- Músculos de la pierna: como el cuádriceps, músculos isquiotibiales, músculo tibial posterior o el sóleo; aportan estabilidad y equilibrio en el movimiento de las piernas.

- Musculatura y huesos de los miembros superiores: la marcha humana incluye el balanceo de los miembros superiores, flexionando el miembro superior opuesto al miembro inferior oscilante. Esto es necesario para mantener el equilibrio y lograr una marcha simétrica y estable.
- Sistema nervioso: es el responsable de coordinar los movimientos de las piernas y los pies durante la marcha, asegurando que los músculos se contraigan y relajen de manera apropiada.

Durante la marcha los movimientos se repiten en lo que se conoce como el ciclo de la marcha. La Figura 3.1 ilustra el movimiento realizado a lo largo de un ciclo de marcha junto con el nombre de cada una de las fases. Este ciclo de marcha incluye desde que se produce el contacto del talón con el suelo hasta que vuelve a tocar el suelo el mismo talón.

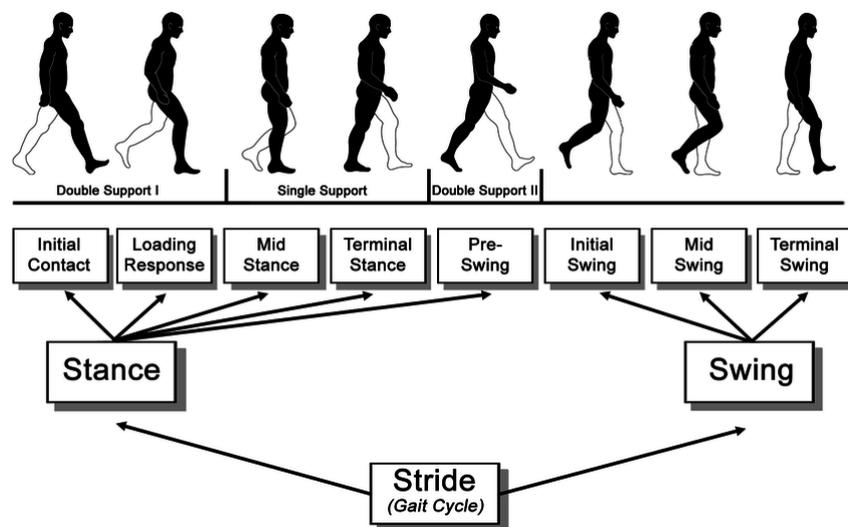


Figura 3.1: Se presentan las fases de la marcha [42], donde *initial contact* y *loading response* se corresponden con HS, *mid stance* y *terminal stance* con FF y *pre-swing* con PO.

El ciclo de la marcha se divide en dos fases: **fase de apoyo** y **fase de vuelo**. La fase de apoyo se define como el periodo de tiempo en el cual el pie de referencia está apoyado en el suelo y normalmente ocupa un 60% del tiempo del ciclo. La fase de oscilación se define como el periodo de tiempo en el que el pie de referencia no está tocando el suelo y ocupa el 40% restante del tiempo del ciclo.

Las fases de apoyo y de vuelo, a su vez, se dividen en diferentes subfases, como se ilustra en la Figura 3.1. La fase de apoyo incluye las siguientes subfases:

- Carga (*Load*): comienza con el evento *Heel Strike* (HS), que hace referencia al primer contacto del pie con el suelo. Se produce flexión dorsal del tobillo y flexión de la cadera. En cuanto a los músculos involucrados, en el tobillo se encuentran activos

los flexores dorsales (tibial anterior), en la rodilla isquiotibiales y cuádriceps y en la cadera los extensores (glúteo mayor y medio). En este TFG tanto al evento HS como a la fase de carga se les denomina HS, ilustrando así que en esta fase sólo hay apoyo del talón. Esta fase termina con el inicio de la respuesta a la carga, conocido como *Toe Strike* (TS), que da paso a la siguiente fase.

- Apoyo plano (*Foot Flat*, FF): comprende desde que se realiza el contacto de los dedos del pie con el suelo (TS), hasta que comienza la elevación del talón. En los primeros instantes, el peso del cuerpo se transfiere al pie de referencia, que es sobre el que se describe el ciclo de marcha. Esta fase coincide con el comienzo de la fase de oscilación del pie opuesto. En el pie de referencia se estabiliza la cadera y la rodilla, interviniendo el músculo sóleo. En el final de esta fase el peso del cuerpo pasa a los metatarsianos del pie de referencia, dando comienzo al despegue del talón del suelo. Los músculos estabilizadores en esta fase son los gemelos, el sóleo y el tríceps sural. La fase FF finaliza con el evento denominado *Heel Off* (HO), que corresponde al despegue del talón.
- Preoscilación (*Push Off*, PO): son los últimos instantes de contacto del pie de referencia con el suelo, preparando el miembro para la oscilación. Se activan los flexores de la cadera, que impulsan el muslo hacia adelante y se mantienen contraídos los gemelos, induciendo una flexión de la rodilla. El último instante en el que el talón deja de estar en contacto con el suelo se denomina *Toe Off* (TO). El evento TO da paso a la fase de oscilación o *swing* (SW).

Por otra parte, la **fase de oscilación** incluye las siguientes fases [43]:

- Fase inicial de oscilación: el pie se levanta del suelo y se mueve hacia adelante para prepararse para el siguiente paso. En esta fase es importante asegurar la separación del pie con respecto al suelo, para lo cual se flexiona el primer metatarsiano. Interviene la musculatura pretibial (tibial anterior).
- Fase media de oscilación: el pie continúa moviéndose hacia adelante y la pierna se extiende, mientras el talón se eleva. Esto se produce gracias a la actuación de los músculos flexores dorsales.
- Fase final de oscilación: el pie se mueve hacia abajo y hacia adelante para tocar el suelo y comenzar de nuevo la fase de apoyo. Se lleva a cabo una extensión completa de la rodilla.

Normalmente, se considera la fase de oscilación como una única fase. Esta es la perspectiva que se toma en este trabajo.

La Tabla 3.1 muestra un resumen de los músculos involucrados en la marcha humana.

Tabla 3.1: Resumen de los músculos involucrados en la marcha humana.

	Contacto inicial	Tibial anterior Isquiotibiales, cuádriceps Glúteo mayor y medio
Fase de apoyo	Apoyo plano	Sóleo Tibial posterior Peroneos, gemelos Triceps sural
	Preoscilación	Flexores de cadera Gemelos
	Fase inicial de oscilación	Flexor del primer metatarsiano
Fase de oscilación	Fase media de oscilación	Flexores dorsales
	Fase final de oscilación	Cuádriceps Flexo-extensores dorsales

La segmentación de estas fases desempeña un papel fundamental en el estudio detallado de la marcha. Además de analizar los ángulos de las piernas en los momentos relevantes descritos anteriormente, resulta relevante examinar la diferencia en la proporción de la fase de oscilación y la fase de apoyo con respecto al 60-40%. También es importante considerar la proporción de tiempo dedicada a cada una de las subfases.

3.2. Unidades de medición inercial (IMUs)

Una IMU es un dispositivo electrónico que permite obtener mediciones de velocidad angular y fuerza específica, usando una combinación de acelerómetros, giróscopos y, en ocasiones, magnetómetros [44]. Los sistemas de navegación de los teléfonos móviles o el estudio del movimiento humano son algunas de las aplicaciones que utilizan estos dispositivos.

El funcionamiento de las IMUs se basa en las dos primeras leyes de Newton. La primera ley de Newton establece que un cuerpo no cambiará su movimiento a menos que actúe sobre él una fuerza. Por otra parte, la segunda ley de Newton (Ecuación (3.1)) afirma que la fuerza ejercida sobre este cuerpo producirá en él una aceleración proporcional a ella [45]:

$$\vec{F} = m \cdot \vec{a}, \quad (3.1)$$

siendo \vec{F} la fuerza, m la masa del cuerpo y \vec{a} la aceleración.

Una IMU consta, comúnmente, de un acelerómetro triaxial y un giróscopo triaxial, como se ilustra en la Figura 3.2. En primer lugar, un acelerómetro triaxial es una combinación de tres acelerómetros uniaxiales colocados ortogonalmente. Este proporciona información de la aceleración lineal, calculada como la derivada segunda de la posición.

Esta aceleración es una combinación de la aceleración lineal instantánea (\vec{a}), que es la que tiene el cuerpo en un momento específico, y la gravedad (\vec{g}). En segundo lugar, un giróscopo triaxial es una combinación de tres giróscopos uniaxiales perpendiculares entre sí. Cada uno de estos giróscopos permite medir la velocidad angular ($\vec{\omega}$) del cuerpo con respecto al sistema de referencia, consiguiendo así conocer la rotación del cuerpo en el espacio 3D. La información obtenida del acelerómetro, el giróscopo y el magnetómetro puede combinarse mediante diferentes algoritmos para obtener la orientación de la IMU en el espacio tridimensional.

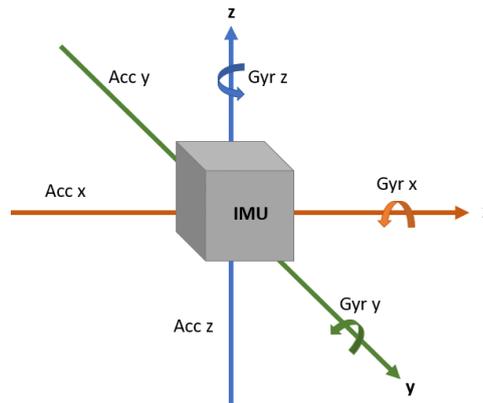


Figura 3.2: Se presentan los tres ejes de acelerómetro triaxial y del giróscopo triaxial.

En ocasiones, las IMUs también incluyen magnetómetros. Un magnetómetro triaxial es un elemento sensible al campo magnético, que permite, como una brújula, obtener información de la orientación del cuerpo respecto al norte magnético terrestre.

En la actualidad existen variedad de IMUs, entre las que se encuentran dispositivos de pequeño tamaño. Este tamaño posibilita que determinadas IMUs se puedan colocar en distintas partes del cuerpo, tales como cabeza, extremidades y tronco, sin intervenir en la ejecución del movimiento natural. De esta forma, se puede medir el movimiento humano y estimar las posiciones articulares del cuerpo fuera de las condiciones controladas de laboratorio.

La portabilidad de las IMUs en entornos no controlados sin interferir en el movimiento las hace idóneas para el análisis de la marcha. Es por ello que este TFG se centra en desarrollar algoritmos de segmentación de marcha basados únicamente en IMUs.

3.3. Sensores de fuerza resistivos (FSRs)

Un FSR, es un material que cambia su resistencia cuando se le aplica una fuerza o presión [46]. La relación entre la fuerza aplicada y la resistencia de un FSR se puede ver en la Figura 3.3.

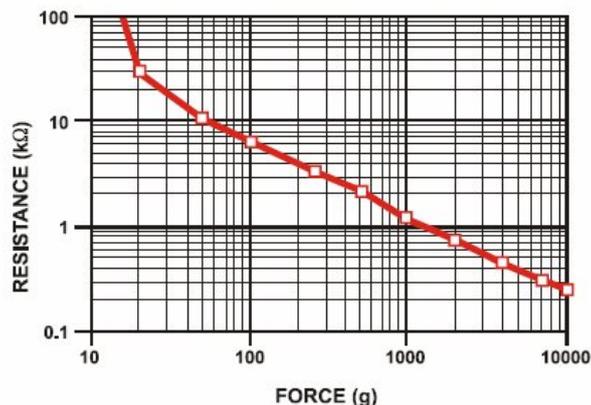


Figura 3.3: Representación de la resistencia vs fuerza de un FSR [47].

Los FSRs se caracterizan por ser sensores sencillos de fabricar y de bajo coste, aunque no son muy precisos. Estos suelen estar compuestos por dos capas conductoras y una película delgada. La estructura de estos sensores se puede observar en la Figura 3.4. Cuando se aplica una fuerza externa al sensor, la película se deforma contra la capa conductora, cambiando la resistencia eléctrica del sensor. Esta variación de la resistencia se mide y se utiliza para determinar la magnitud de la carga aplicada.

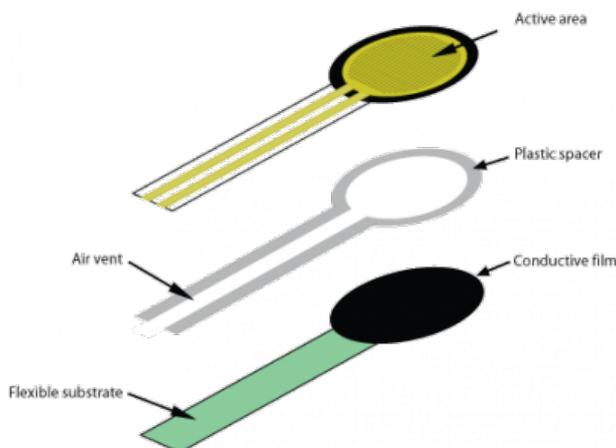


Figura 3.4: Representación de la estructura de un FSR [48].

Este tipo de sensores son utilizados para múltiples aplicaciones, como pueden ser botones sensibles al tacto [49], estudios del movimiento del paciente durante el sueño [50], o evaluación de la marcha humana [51].

3.4. Inteligencia artificial

Según la RAE, la **IA** es la “disciplina científica que se ocupa de crear programas informáticos que ejecutan operaciones comparables a las que realiza la mente humana, como el aprendizaje o el razonamiento lógico”. El aprendizaje automático, o ML, y el

aprendizaje profundo, o DL, son dos enfoques fundamentales en el campo de la IA que comparten objetivos comunes pero se diferencian en su estructura y complejidad [52].

Por una parte, **ML** hace referencia a un subconjunto de la IA donde las máquinas se “entrenan” para reconocer patrones basados en datos y hacer sus predicciones. El objetivo del ML es conocer la relación existente entre los datos de entrada y los datos de salida para, de esta manera, poder realizar predicciones. Los métodos basados en ML utilizan técnicas estadísticas y matemáticas para identificar patrones y realizar predicciones o clasificaciones basadas en estos patrones.

Por otro lado, **DL** es un subconjunto de ML que se basa en redes neuronales más complejas y profundas que imitan el funcionamiento del cerebro humano. Estas redes tienen múltiples capas ocultas que les permiten inferir una relación entre las características de entrada y la variable a predecir. El aprendizaje profundo es capaz de abordar problemas más complejos y extraer información más abstracta.

3.4.1. Tipos de aprendizaje automático

Los algoritmos de aprendizaje automático se basan en la utilización de datos de entrada y salida para entrenar y realizar predicciones o clasificaciones. Los datos de entrada, también conocidos como conjunto de características representan la variable predictiva (denominada como x) y se proporcionan al algoritmo para que pueda aprender y realizar inferencias. Por otro lado, los datos de salida constituyen la variable respuesta (denominada como y) y representan la respuesta esperada que el algoritmo debe producir para cada conjunto de características de entrada. Durante el entrenamiento, el algoritmo ajusta los parámetros del modelo utilizando los datos de entrada y de salida (si estos están disponibles). Una vez entrenado, el algoritmo es capaz de tomar nuevos datos de entrada y producir predicciones basadas en el conocimiento adquirido durante el entrenamiento [53].

Según si se tienen o no las etiquetas asociadas a los datos, se distinguen dos tipos de aprendizaje automático: aprendizaje supervisado y aprendizaje no supervisado. El **aprendizaje supervisado** consiste en aprender la relación entre los datos y las etiquetas asociadas a estos datos. Por otro lado, el **aprendizaje no supervisado** consiste en aprender la estructura de los datos agrupando los ejemplos en grupos consistentes con características similares.

Los métodos de aprendizaje supervisado utilizan bases de datos etiquetadas con el objetivo de inferir relaciones entre la variable de entrada (x) y la variable de salida (y). A partir de los pares (x, y) , se logra realizar las predicciones. Según cómo son las etiquetas relacionadas con los datos, se pueden diferenciar dos tipos de aprendizaje automático: clasificación y regresión. En clasificación el conjunto de etiquetas es numerables y el objetivo

es asignar una clase; en cambio, en regresión el conjunto de etiquetas no es numerable y el objetivo es predecir valores continuos.

En el caso del aprendizaje no supervisado, se trata de encontrar patrones a partir de los datos de entrada sin ninguna referencia de los datos etiquetados. Una de las técnicas más utilizadas en aprendizaje no supervisado es el *clustering*. El *clustering* es una técnica empleada para agrupar datos similares en conjuntos. Los algoritmos de *clustering* evalúan la similitud entre los datos en función de sus características y los agrupan en base a esta similitud.

Existe un tercer paradigma dentro del campo del aprendizaje automático. Este es el **aprendizaje por refuerzo** y se diferencia del aprendizaje supervisado y del aprendizaje no supervisado. En el aprendizaje por refuerzo un agente interactúa con el entorno tomando decisiones secuenciales para maximizar una recompensa obtenida en base al conjunto correcto de acciones. El agente aprende mediante ensayo y error, a través de una interacción continua con el entorno.

En el desarrollo de este TFG, se utilizan diferentes métodos basados en aprendizaje supervisado, puesto que al entrenar los modelos de ML se tienen en cuenta las etiquetas asociadas a los datos. Además, se realizan dos tareas: segmentación de las fases de la marcha (clasificación) y predicción del tiempo restante hasta los eventos relevantes a lo largo del siguiente ciclo de marcha (regresión).

3.5. Algoritmos de ML

En esta sección se explican los modelos utilizados para realizar la clasificación en la segmentación de las fases de marcha y la regresión para estimar los tiempos hasta los eventos de la misma.

3.5.1. Regresión Lineal

La **regresión lineal** es un método de aprendizaje automático supervisado para regresión que busca el hiperplano que mejor describa la relación entre las variables predictivas (denominadas variables de entrada o variables independientes) con la variable dependiente, que se corresponde con la variable respuesta.

La regresión lineal puede ser simple o múltiple. En el caso de la regresión lineal simple, sólo hay una variable predictiva para realizar la predicción de la variable respuesta de la siguiente forma:

$$y = ax + b. \tag{3.2}$$

donde x es la variable predictiva, y la variable respuesta, b el sesgo o *bias* y a la pendiente de la recta de regresión.

Sin embargo, en la regresión lineal múltiple, hay varias variables predictivas (x_1, x_2, \dots, x_n) que contribuyen en mayor o menor medida a la variable respuesta [54]:

$$y = a_1x_1 + a_2x_2 + \dots + a_nx_n + b. \quad (3.3)$$

Los coeficientes a y b de la regresión simple y (a_1, a_2, \dots, a_n) y b de la regresión múltiple se determinan durante el proceso de entrenamiento.

El objetivo de la regresión lineal es minimizar el error cometido en la estimación. Cuanto menor sea este error, mejor será la bondad del ajuste de la recta. Para calcular los coeficientes que minimicen este error se utiliza el método de mínimos cuadrados.

El método de mínimos cuadrados busca los mejores coeficientes de regresión minimizando la suma de los errores al cuadrado. Este error, conocido como *Sum of Squares Error* (SSE) se muestra en la Ecuación (3.4):

$$SSE = \sum_i^N (y_i - \hat{y}_i)^2, \quad (3.4)$$

siendo y_i el valor real y \hat{y}_i el valor estimado.

Algunas de las ventajas de utilizar este error para calcular los coeficientes de regresión es que es fácil de analizar e interpretar y tiene bajo coste computacional. Los coeficientes del regresor en este problema pueden ser determinados por la expresión analítica:

$$\hat{a} = (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t Y, \quad (3.5)$$

donde \mathbf{X}^t denota a la matriz transpuesta de \mathbf{X} y \mathbf{X} es:

$$X = \begin{bmatrix} x_1^1 & x_2^1 & x_3^1 & \dots & x_n^1 \\ x_1^2 & x_2^2 & x_3^2 & \dots & x_n^2 \\ \dots & \dots & \dots & \dots & \dots \\ x_1^N & x_2^N & x_3^N & \dots & x_n^N \end{bmatrix}, \quad (3.6)$$

siendo n el número de características que componen la variable de entrada, y N el número de observaciones.

3.5.2. Regresión Logística

La **regresión logística** es un método que trata de clasificar las observaciones en clases encontrando fronteras que establecen los límites entre unas y otras [55]. Para calcular las

fronteras de decisión, el primer paso es transformar los datos a través de la Ecuación (3.7):

$$z_k = \vec{\omega}_{kp} \cdot \vec{x}_p + b_k, \quad 0 \leq k \leq K, \quad (3.7)$$

siendo K el número de clases en las que clasificar el problema, p el número de características, \vec{x} las variables independientes x_1, x_2, \dots, x_p para la clase k , $\vec{\omega}$ los coeficientes de regresión asociados con las variables independientes y b_k el sesgo. Los pesos y el sesgo se aprenden durante el entrenamiento mediante el método del descenso por gradiente, o cualquier otro método iterativo.

Para obtener la clasificación, los valores estimados en la transformación son convertidos en probabilidades mediante la función de activación *softmax* tal como se muestra en la Ecuación (3.8):

$$p_i = \frac{e^{z_k}}{\sum_{j=0}^K e^{z_j}}. \quad (3.8)$$

La clase asignada será aquella que tenga una mayor probabilidad.

Durante el entrenamiento, se utiliza el método del descenso por gradiente para encontrar los coeficientes \vec{w}_k y \vec{b}_k . Para ello, en primer lugar se define la función de error, que es la entropía cruzada, detallada en la Ecuación (3.9):

$$error = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K y_{ik} \log(p_{ik}), \quad (3.9)$$

donde N es el número total de datos utilizados para la predicción, K el número total de clases, y_{ik} la etiqueta real de la observación i para la clase k y p_{ik} la probabilidad predicha de la observación i para la clase k .

El objetivo será minimizar esta función de coste. Para ello, se escoge un valor aleatorio de ω , y en cada iteración se actualiza este valor gracias a la tasa de aprendizaje α . Esto se repite hasta que se encuentra el mínimo de la función de coste.

Es importante destacar que en la regresión logística la frontera de decisión siempre es lineal, por lo que este modelo no resulta eficiente para aquellos problemas donde la distribución es compleja y las observaciones no pueden ser separadas por una frontera lineal.

Otro aspecto importante es la regularización en los modelos de regresión lineal o logística. La regularización consiste en añadir penalizaciones al modelo a medida que aumenta su complejidad [56].

En este TFG se utiliza la regularización *Ridge* o L_2 que añade un término de penali-

zación haciendo que la función de pérdida quede como la Ecuación(3.10):

$$SSE = \sum_i (y_i - \hat{y}_i)^2 + \lambda \sum_j \beta_j^2. \quad (3.10)$$

Esta regularización mantiene en la función de pérdida las variables menos importantes, pero minimiza su impacto. Esta es la razón por la que se ha escogido esta regularización para este proyecto, ya que es interesante mantener todas las variables, y no eliminarlas como realiza la regularización *Lasso* o L_1 . Gracias a la regularización se consigue limitar la complejidad del modelo, evitando así el sobreajuste de los datos.

3.5.3. Support Vector Machine

A continuación se muestra el funcionamiento del modelo SVM para clasificación (*Support Vector Classifier*, SVC) y para regresión (*Support Vector Regressor*, SVR).

SVC es un algoritmo de aprendizaje automático supervisado que trata de encontrar un hiperplano óptimo que separe las clases [57]. Un hiperplano es un subespacio de una dimensión menos que su espacio natural que separa el espacio en dos medios espacios. Por ejemplo, un hiperplano de un espacio n -dimensional es un subconjunto con dimensión $n - 1$.

Para este modelo, el mejor hiperplano es aquel que maximiza la distancia al elemento más cercano de cada etiqueta, es decir aquel que deja el margen máximo. Se denominan vectores soporte a los puntos utilizados para maximizar la distancia entre las etiquetas y calcular así el hiperplano. Estas son las observaciones verdaderamente importantes, puesto que son aquellas que “apoyan” la ubicación de la línea.

Si las clases no se pueden separar por un hiperplano lineal, el modelo SVC utiliza un *kernel* para transformar los datos en un espacio de mayor dimensión donde las clases pueden ser separadas linealmente. Los *kernel* más comunes son el *kernel* lineal, el *kernel* polinómico y el *kernel* radial.

Este mismo procedimiento se puede utilizar para clasificación multiclase, teniendo en cuenta el enfoque *one vs all* (uno contra todos). Esto consiste en entrenar un clasificador para cada clase, tratando de separar esa clase del resto de las clases. Cuando se recibe una muestra, se ejecutarán todos los clasificadores y se elige la clase con la mayor probabilidad de pertenencia.

En el caso de regresión, se utiliza el **SVR**. El modelo SVR trata de encontrar una función que se ajuste a los datos y minimice el error de predicción.

3.5.4. K-Nearest Neighbors

El algoritmo **KNN**, es un método no paramétrico que utiliza la proximidad de las observaciones para obtener las predicciones [58]. A continuación se detalla su funcionamiento para clasificación y regresión.

En clasificación, dado un conjunto etiquetado T y una nueva observación x , representada con una estrella roja en la Figura 3.5, el clasificador KNN busca los k vecinos más próximos de x en T . Entre los vecinos más cercanos, KNN selecciona la clase más frecuente [59].

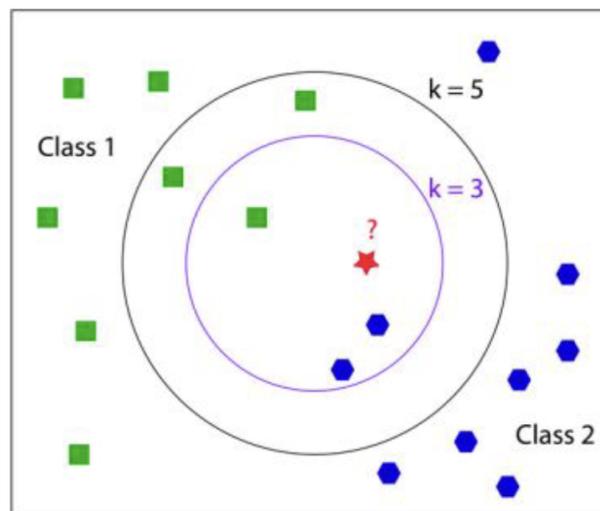


Figura 3.5: Ejemplo de clasificador KNN [59].

La proximidad se define según una función de similitud basada en la distancia, de tal manera que a mayor distancia, menor similitud y viceversa. El valor de k se selecciona según las necesidades del problema, teniendo en cuenta los siguientes aspectos:

- Si k es igual al número de observaciones, no existirá frontera de clasificación, de tal manera que cada vez que aparece una nueva observación, esta se clasifica dentro de la clase mayoritaria.
- Si k es demasiado alto, la frontera será más suave y menos sensible al ruido.
- Si $k = 1$, la frontera de decisión será demasiado flexible, es decir, cada vez que aparece una nueva observación, se le otorgará la etiqueta de su vecino más próximo.

En el ejemplo de la Figura 3.5, para $k = 1$, la observación roja sería clasificada en la clase azul; sin embargo, para $k = 5$, la observación roja sería clasificada en la clase verde. Otro ejemplo de la influencia del valor de k en la frontera de clasificación se muestra en la Figura 3.6, donde se observa que cuanto mayor es el valor de k , más suave es la frontera. Al contrario, a menor valor de k , la frontera se ajusta más a los datos.

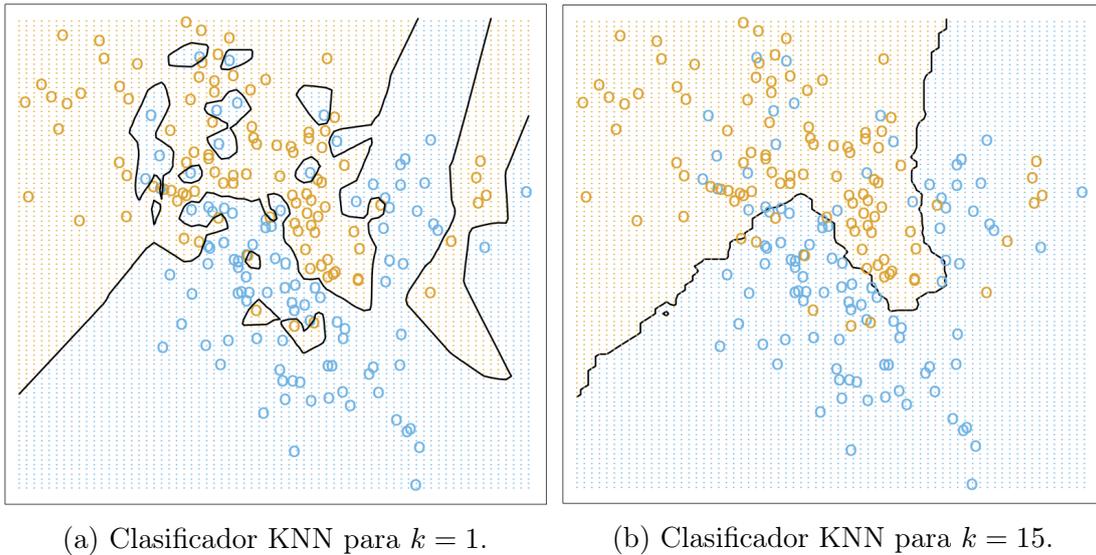


Figura 3.6: Comparativa entre un clasificador KNN para $k = 1$ y $k = 15$ [60].

En resumen, es importante encontrar un equilibrio en el valor del hiperparámetro k , de tal manera que sea lo bastante grande como para que tome en cuenta suficientes observaciones, pero lo suficientemente pequeño como para que sea flexible y se ajuste a los datos de entrada.

En el caso de regresión, el funcionamiento es muy similar al de clasificación. Su diferencia de funcionamiento recae en que, en la predicción de la regresión, en lugar de etiquetar la nueva observación con la clase más numerosa entre las k seleccionadas, se calcula la media de las k observaciones más cercanas.

3.5.5. Árbol de decisión

El **árbol de decisión** consiste en un modelo de clasificación en forma de árbol para predecir el valor de la variable respuesta mediante reglas sencillas. Estas reglas se obtienen a partir de un proceso de aprendizaje con ejemplos. Las diferencias en su implementación para problemas de clasificación y regresión se detallan a continuación.

Para la tarea de clasificación durante el entrenamiento se crean ramas en los árboles tratando de separar los ejemplos de distintas clases de la mejor forma posible. Se comienza generando el primer nodo también conocido como nodo raíz, que es la primera división en función de la variable más importante. Después se continúa creando ramas en los nodos internos o intermedios. Finalmente, los nodos terminales u hojas indican la clasificación definitiva [61].

Para seleccionar la variable a elegir para obtener la mejor división, se puede considerar

el índice Gini:

$$GINI(t) = 1 - \sum_{i=1}^N (P_i)^2, \quad (3.11)$$

o la entropía:

$$H = - \sum_{i=1}^N P_i \cdot \log_2 P_i, \quad (3.12)$$

donde P_i es la probabilidad de que una observación sea de la clase i .

El índice Gini es una medida que indica el grado de impureza de un nodo. A menor índice de Gini mayor pureza y viceversa. Si el índice de Gini es igual a cero el nodo se considera puro, es decir, únicamente contiene datos de una categoría. A mayor índice menor pureza, es decir, contiene datos de más de una categoría. Por lo tanto, se seleccionarán las variables con menor índice de Gini.

Por otra parte, la entropía se define como la medida del desorden de un sistema. En el caso de que un nodo sea puro, su entropía es 0, solo contiene datos de una categoría. Sin embargo, si la entropía es 1, significa que existe la misma frecuencia para cada una de las clases de observaciones.

En el caso de los árboles de decisión de un problema de regresión, el procedimiento es similar al de los árboles de clasificación, pero la diferencia reside en que una vez dividido el árbol en nodos, los valores de salida se calculan tomando el promedio de los valores que forman parte de cada subconjunto [61].

Además, en el caso de los problemas de regresión se utiliza el SSE que es una medida de la diferencia entre los datos reales y los predichos por el modelo, definido en la Ecuación (3.4). El objetivo es minimizar el SSE, puesto que a menor SSE, mejor ajuste del modelo a los datos.

Tanto para problemas de clasificación como para problemas de regresión, es importante tener en cuenta el concepto “profundidad del árbol”, que hace referencia al número máximo de nodos del árbol, es decir, cuantas veces se divide una muestra hasta que se llega al valor predicho. Para evitar que se produzca un sobreajuste de los datos se realiza una poda del árbol, que consiste en establecer la profundidad del árbol como un hiperparámetro y calcularlo minimizando el error de validación.

3.5.6. Métodos Ensemble

Combinando varios árboles de decisión se pueden generar modelos predictivos más complejos. Estos métodos se denominan *ensemble* y los más conocidos son *bagging* y *boosting* [62].

- *Bagging*: se combinan múltiples modelos, cada uno de ellos entrenado con un subconjunto distinto de los datos de entrenamiento. Como valor final, se toma la media de todas las predicciones (en problemas de regresión), o la clase más numerosa (en problemas de clasificación). Un ejemplo de esto es el modelo RF.
- *Boosting*: el modelo final se ajusta a partir de múltiples *weak learners*, que son modelos sencillos que predicen ligeramente mejor que lo esperado por azar. Como valor final se toma la media de todas las predicciones (en el caso de problemas de regresión), o la clase más numerosa (en problemas de clasificación). Dos ejemplos de esto son *AdaBoost* y *XGBoost*.

Para entender la diferencia entre estos dos métodos es importante conocer los conceptos de varianza y sesgo, vea la Figura 3.7. La varianza hace referencia a la sensibilidad de un modelo a pequeñas variaciones en los datos de entrenamiento. Si la varianza es alta, significa que el modelo se ajusta demasiado a los datos de entrenamiento. Por otra parte, el sesgo se refiere a la capacidad de un modelo de aprender patrones generales a partir de datos. El objetivo será disminuir la varianza y el sesgo.

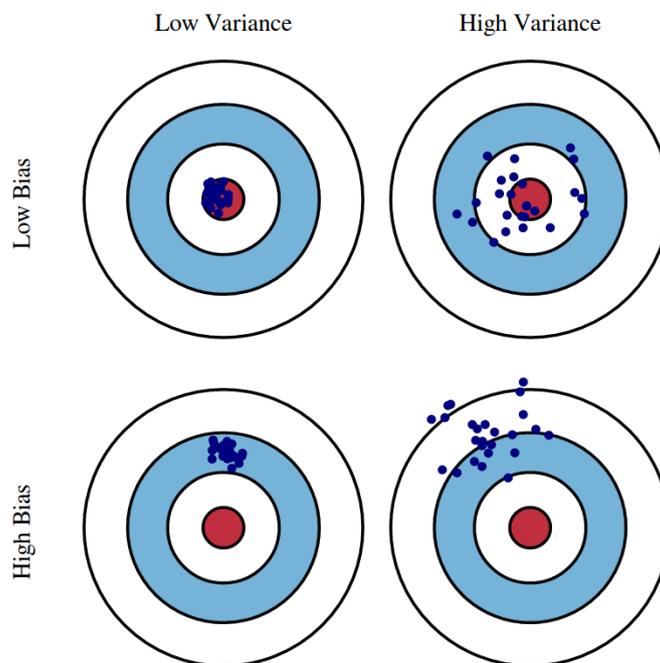


Figura 3.7: Representación visual de la varianza y el sesgo [63].

Las diferencias entre los modelos *bagging* y los *boosting* se reflejan en la Tabla 3.2.

Tabla 3.2: Tabla comparativa de los modelos *bagging* y *boosting*.

Bagging	Boosting
El modelo final es la combinación de varios árboles, cada uno de los cuales es entrenado con un subconjunto distinto de los datos de entrenamiento	El modelo final se ajusta aprendiendo de los errores del modelo anterior
Se emplean modelos con muy poca <i>bias</i> pero mucha varianza, y combinándolos se consigue reducir la varianza sin apenas aumentar el <i>bias</i>	Se emplean modelos con muy poca varianza pero mucho <i>bias</i> , ajustando secuencialmente los modelos se reduce el <i>bias</i> sin apenas modificar la varianza.

3.5.6.1. Random Forest

Un modelo **RF** está formado por un conjunto de árboles de decisión, donde cada uno de ellos está entrenado con una muestra aleatoria de los datos de entrenamiento. Estas muestras se eligen mediante *bootstrap* (técnica estadística que toma múltiples muestras con reemplazo de una sola muestra aleatoria), y las observaciones no estimadas para entrenar los árboles (conocidas como “*out of the bag*”) se utilizan para validar el modelo. De esta manera, la predicción de una nueva observación se obtiene agregando las predicciones de todos los árboles individuales que forman el modelo. En este caso, uno de los hiperparámetros a optimizar es el número de árboles aleatorios que se crean. Generalmente, a mayor número de árboles, mejor resultado, pero a partir de un cierto número, los resultados dejan de mejorar significativamente, y aumenta considerablemente el coste computacional [62].

3.5.6.2. AdaBoost

Otro de los modelos utilizados es **AdaBoost**. Este se basa en el ajuste iterativo de un modelo *weak learner* (también denominado *base learner*). Se comienza creando un modelo *weak learner*, que predice la variable respuesta con un porcentaje de acierto ligeramente superior a lo esperado por azar. Este *weak learner* suele ser un árbol con apenas unos pocos nodos [62]. A partir de aquí, dependerá de si el problema a abordar es de regresión o de clasificación.

En el caso de que el problema a resolver sea una clasificación, se establece un peso inicial igual para todas las observaciones que forman el conjunto de entrenamiento. A partir de ahí comienza el ajuste iterativo detallado a continuación.

En la primera iteración, se ajusta el *weak learner* empleando el conjunto de entrenamiento y el peso inicial establecido. El peso inicial es igual para todas las observaciones, como muestra Ecuación (3.13):

$$\omega_i = \frac{1}{N}, \quad 1 \leq i \leq N, \quad (3.13)$$

donde N es el número de observaciones del entrenamiento. Con este *weak learner* se predicen las observaciones de entrenamiento y se distinguen las que están mal clasificadas de las que están bien clasificadas.

A continuación se actualizan los pesos dando mayor importancia a las observaciones mal clasificadas, y disminuyendo en el caso de las observaciones bien clasificadas. Se asigna un peso total al *weak learner*, proporcional al número de aciertos. A mayor número de aciertos, mayor influencia tendrá en el conjunto *ensemble*.

El error del *weak learner* se define en la Ecuación (3.14):

$$err_m = \frac{\sum_{i=1}^N \omega_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^N \omega_i}, \quad (3.14)$$

siendo M el número de iteraciones del aprendizaje (número de *weak learners*), G_m el *weak learner* de la iteración m y ω_i el peso de la observación i

Para calcular el peso asignado al *weak learner* G_m , denotado como α_m , se utiliza la Ecuación (3.15):

$$\alpha_m = \log \left(\frac{1 - err_m}{err_m} \right), \quad (3.15)$$

y, con la Ecuación (3.16), se actualizan los pesos de las observaciones.

$$\omega_i = \omega_i \exp [\alpha_m I(y_i \neq G_m(x_i))], \quad 1 \leq i \leq N \quad (3.16)$$

En la siguiente iteración, se toma un nuevo *weak learner* y se ajusta teniendo en cuenta los pesos de la iteración anterior. Este nuevo *weak learner* se almacena, teniendo así otro modelo para el *ensemble*.

Este proceso se repite M veces, obteniendo de esta manera M *weak learners* para realizar el *ensemble*. Cuando llega una nueva observación, esta se clasifica obteniendo la predicción para cada uno de los *weak learners* que forman el *ensemble*, y se pondera cada uno de ellos teniendo en cuenta el peso asignado en el ajuste [64]. En clasificación binaria se emplea la Ecuación (3.17):

$$G(x) = \text{sign} \left[\sum_{m=1}^M \alpha_m G_m(x) \right], \quad (3.17)$$

mientras que en clasificación multiclase se utiliza la Ecuación (3.18):

$$G(x) = \arg \max_k \left[\sum_{m=1}^M \alpha_m I(G_m(x) = k) \right], \quad (3.18)$$

donde k es la clase del conjunto $k \in 1, 2, \dots, K$.

Si el problema es de regresión, se comienza de la misma manera, seleccionando un conjunto inicial de datos y asignando pesos iguales a cada una de las muestras. A continuación, se entrena un modelo de regresión débil en el conjunto de datos de entrenamiento aumentando los pesos de las muestras predichas incorrectamente y disminuyendo los pesos de las muestras predichas correctamente. Luego se entrena otro modelo de regresión débil teniendo más en cuenta las muestras difíciles de predecir. Esto se repite hasta que se alcanza un número determinado de modelos o hasta que se alcanza una precisión determinada.

Por último se combinan los modelos de regresión débil en un modelo de regresión fuerte, definido en la Ecuación (3.19):

$$G(x) = \arg \max \left[\sum_{m=1}^M \alpha_m G_m(x) \right]. \quad (3.19)$$

Este modelo es el que se utiliza para realizar las predicciones.

3.5.6.3. XGBoost

Otro de los algoritmos más importantes de *boosting* es **XGBoost**. Este algoritmo se basa en *gradient boosting*. La diferencia con *AdaBoost* es que este se enfoca en los datos que se han predicho incorrectamente, y *gradient boosting* se enfoca en los errores de los modelos previos. Además, *XGBoost* tiene un algoritmo de selección de características incorporado, que permite seleccionar las características más importantes para resolver la tarea de predicción. Esto, unido al hecho de que lleve a cabo el aprendizaje en paralelo, hace que se reduzcan de manera considerable los tiempos de entrenamiento [65].

3.5.7. Perceptrón multicapa

El modelo MLP, también denominado algoritmo de retropropagación, es un modelo matemático cuya forma de aprendizaje se basa en el funcionamiento de las neuronas biológicas. Un perceptrón está formado por los componentes detallados a continuación e ilustrados en la Figura 3.8 [66]:

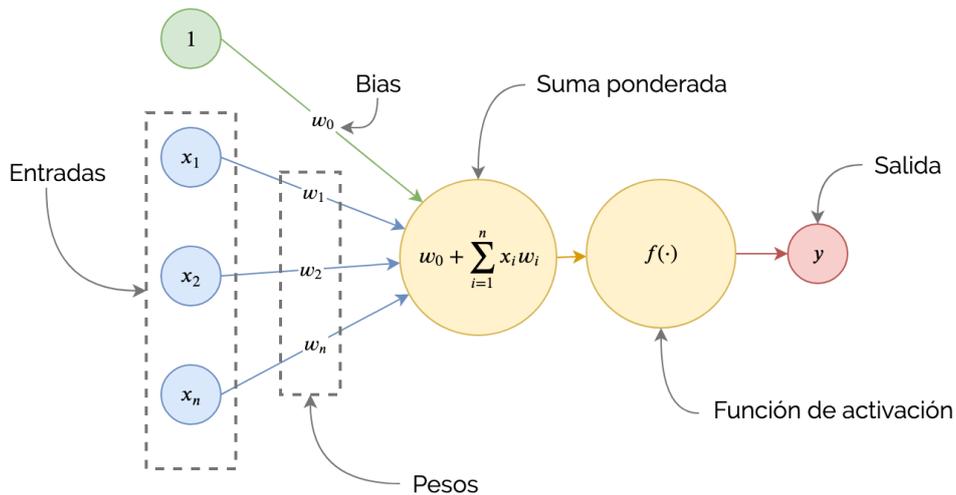


Figura 3.8: Representación de la estructura de una neurona [67].

- Entrada: denotan las características del modelo $x_1, x_2, x_3, \dots, x_n$.
- Pesos: son parámetros ajustables que se utilizan para determinar la fuerza de la conexión entre las entradas y la salida del perceptrón. Cada entrada está asociada con un peso que refleja su contribución a la salida del perceptrón. Se denotan como $\vec{\omega} = (\omega_1, \omega_2, \dots, \omega_n)$.
- Sesgo: es un parámetro adicional que se utiliza para ajustar el resultado de la suma ponderada antes de aplicar la función de activación.
- Función de activación: es una función matemática que transforma la entrada ponderada del perceptrón en una salida no lineal. Algunas de las funciones de activación más comunes, como son la sigmoide, la tangente hiperbólica y la *Rectified Linear Unit* (ReLU), se pueden observar en la Figura 3.9. Estas funciones se caracterizan por lo siguiente:
 - Función sigmoidea: produce una salida suave entre 0 y 1. El principal problema de esta función es que puede saturar a partir de cierto valor.
 - Función tangente hiperbólica: produce una salida suave entre -1 y 1. Al igual que la anterior, puede saturar a partir de cierto valor.
 - Función ReLU: produce una salida lineal si el valor de entrada es positivo, y 0 en caso contrario. La ventaja de esta función respecto a las anteriores es que no satura y tiene un menor coste computacional.

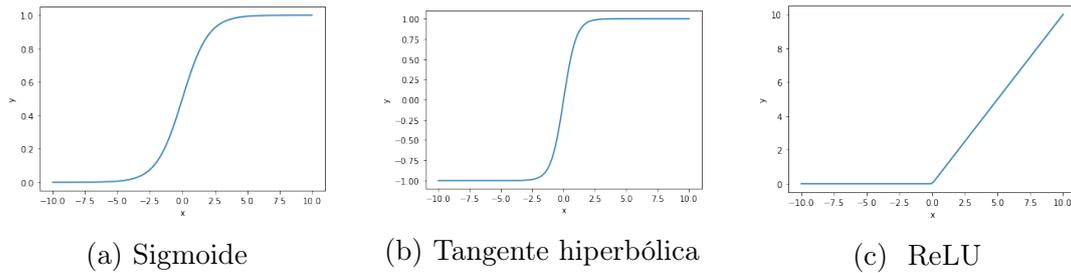


Figura 3.9: Principales funciones de activación.

- **Suma ponderada:** es la operación matemática que se realiza para combinar las entradas del modelo con los pesos asociados a cada entrada. Se calcula multiplicando cada entrada por su peso correspondiente y sumando los resultados.
- **Salida:** una vez calculada la suma ponderada, esta pasa por la función de activación para obtener la salida predicha.

Un modelo MLP tiene múltiples capas ocultas, cada una de ellas con múltiples neuronas artificiales, como se muestra en la Figura 3.10.

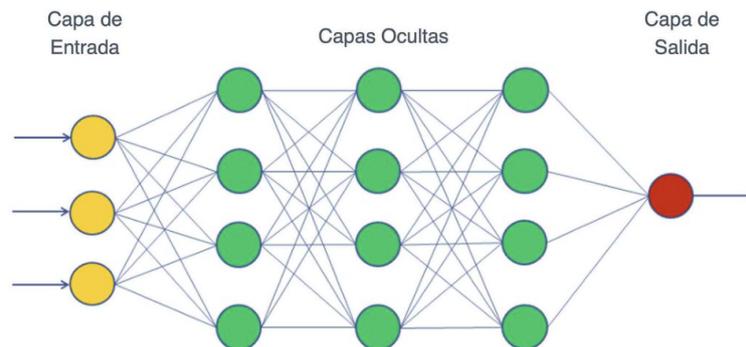


Figura 3.10: Representación de la estructura del perceptrón multicapa [66].

En la última capa (denominada capa de salida), el perceptrón llevará a cabo la predicción. En el caso de que el problema sea de clasificación la última capa estará compuesta por múltiples neuronas, cada una de ellas representando una clase diferente. Sin embargo, si el problema es de regresión, la capa de salida estará formada por una sola neurona que produce un valor de salida continuo.

3.5.8. Long Short-Term Memory

Las **LSTM** son un tipo de RNN que se caracterizan por su capacidad de recordar un dato relevante en la secuencia y preservarlo durante varios instantes de tiempo. Por lo tanto, tienen memoria a corto y a largo plazo.

La arquitectura de las LSTM cuenta con varias partes [68, 69]:

- Puerta de olvido: decide mantener u olvidar la información. Sólo se mantiene la información proveniente de capas previamente ocultas y la entrada actual, esto se regula con la función sigmoide, ver Figura 3.9a.
- Puerta de entrada: controla las nuevas entradas en la célula de la memoria. Utiliza la función sigmoide y tangente hiperbólica (Figura 3.9b) para elegir qué valores conservar y cuáles descartar.
- Puerta de salida: contiene información sobre entradas anteriores y determina el valor del siguiente estado oculto. Primero, los valores del estado actual y el estado oculto previo se pasan por una función sigmoide para después pasarlo por la función tangente hiperbólica. Basándose en esto, la red decide qué información debe llevar el estado oculto. Este estado oculto se utilizará para la predicción.

El esquema de funcionamiento se muestra en la Figura 3.11

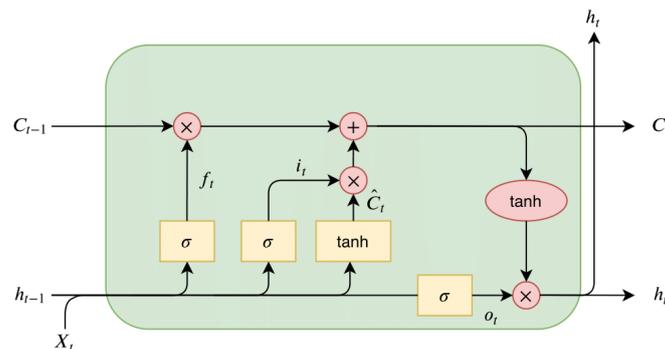


Figura 3.11: Representación de la arquitectura de la red LSTM [70].

Para predecir a partir de estos estados ocultos, se puede incluir una capa densa o completamente conectada que produzca la salida deseada. En el caso de clasificación la salida será una capa densa con el número de neuronas igual al número de clases en las que se quiere clasificar. Sin embargo, si el problema a resolver es de regresión la capa de salida será una capa densa con una sola neurona y una función de activación lineal para obtener la salida continua.

3.6. Estandarización y balanceo de datos

La estandarización de variables es un proceso mediante el cual las variables originales son transformadas a una escala común para así facilitar su comparación o análisis conjunto. La estandarización de variables es una técnica fundamental en el análisis de datos, que permite comparar variables con unidades de escalas diferentes, facilitando así la aplicación

de modelos de IA. Para la estandarización de variables se utiliza la Ecuación (3.20):

$$z = \frac{x - \mu}{\sigma}, \quad (3.20)$$

siendo x la observación original, μ la media de todas las observaciones y σ la desviación estándar.

Por otra parte, el desequilibrio de clases consiste en que las observaciones de una o mas clases de un conjunto de datos están desproporcionalmente representadas en comparación con otras clases, es decir, cuando una o varias clases están subrepresentadas en relación con las demás. El desequilibrio en las muestras de las clases supone un problema en la aplicación de modelos de aprendizaje automático, ya que el modelo puede tener dificultades para predecir la clase minoritaria.

Para abordar el problema del desequilibrio de clases, se utilizan técnicas como el *over-sampling* y el *under-sampling*, que buscan equilibrar las clases para mejorar el rendimiento del modelo:

- *Under-sampling*: consiste en eliminar observaciones de la clase mayoritaria con el objetivo de crear un conjunto de datos equilibrado.
- *Over-sampling*: consiste en la creación de nuevos objetos de la clase minoritaria para producir un nuevo conjunto de datos equilibrado.

Uno de los métodos más conocidos de *over-sampling* es SMOTE (*Synthetic Minority Over-sampling Technique*). SMOTE es un algoritmo que genera nuevas muestras sintéticas a partir de los datos de la clase minoritaria con el fin de solucionar el desbalanceo de los datos. Para ello se enfoca en el espacio de características para generar nuevas observaciones con la ayuda de la interpolación entre las instancias que se encuentran juntas [71], ver Figura 3.12:

1. Se selecciona una instancia de la clase minoritaria al azar.
2. Se obtienen los k vecinos más próximos (5 por defecto) para esta instancia.
3. Se eligen N de estas k instancias para interpolar nuevas observaciones sintéticas. Para esto se calcula la distancia entre el vector de características y sus vecinos. Esta diferencia se multiplica por cualquier valor aleatorio entre 0 y 1 y se agrega al vector de características anterior.

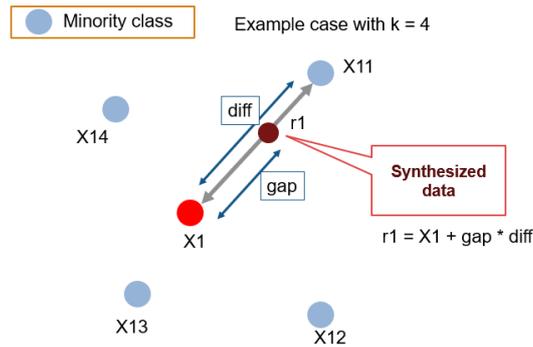


Figura 3.12: Funcionamiento del algoritmo SMOTE [71].

3.7. Validación de modelos de aprendizaje automático

Al entrenar un modelo de aprendizaje automático, los datos se dividen en dos: entrenamiento y test. En ocasiones el modelo se ajusta demasiado a los datos de entrenamiento y pierde así su capacidad de generalizar a datos nuevos. Para detectar este sobreajuste se utiliza la validación cruzada. En la Figura 3.13 se muestra el funcionamiento de la validación cruzada de tipo k -fold con cuatro subconjuntos para el conjunto de entrenamiento.

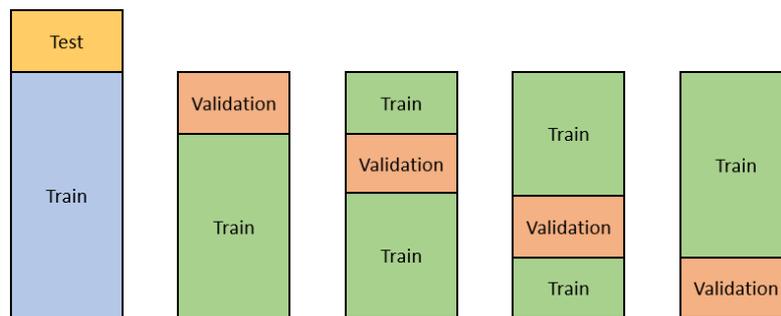


Figura 3.13: Funcionamiento de la validación cruzada de tipo k -fold [72].

La validación cruzada utilizada para evaluar el rendimiento de un modelo predictivo consiste en lo siguiente. Los datos de entrenamiento son divididos a su vez en dos subconjuntos: entrenamiento y validación. A continuación, se realizan diferentes combinaciones de los datos pertenecientes a estos conjuntos para luego combinar los resultados obtenidos y tener así una estimación más precisa del rendimiento del modelo. Esto permite evaluar la capacidad de generalización de un modelo y evitar problemas de sobreajuste.

Otro de los usos de la validación cruzada es la selección de hiperparámetros (valores utilizados para configurar el modelo durante el entrenamiento). Para seleccionar los valores óptimos de los hiperparámetros, se realiza una validación cruzada que divide el conjunto de entrenamiento en entrenamiento y validación. A continuación se entrena el modelo con diferentes valores de hiperparámetros en el conjunto de entrenamiento, y se evalúa el rendimiento en el conjunto de validación. Este proceso se repite varias veces

con diferentes combinaciones de datos en el conjunto de entrenamiento y validación. Al final se selecciona el valor del hiperparámetro con el que se haya obtenido mejores resultados. La correcta selección de los hiperparámetros puede mejorar considerablemente el rendimiento del modelo y su capacidad de predicción. Aún así hay que tener en cuenta el coste computacional asociado a la validación cruzada, puesto que esta requiere entrenar y evaluar el modelo varias veces.

Capítulo 4

Metodología y Desarrollo

Este capítulo describe la metodología seguida y el desarrollo del proyecto. La Figura 4.1 resume el flujo del trabajo realizado, que se describe con detalle en las siguientes secciones. En la Sección 4.1 se describe la base de datos que utilizada, y la Sección 4.2 muestra las señales pertenecientes a la base de datos y la forma en la que se etiquetan dichas señales. Por su parte, la Sección 4.3 explica la preparación de los datos previa al comienzo del entrenamiento, incluyendo la segmentación de la señal y la extracción de características. La validación de los hiperparámetros y selección del ancho de ventana óptimo, así como la descripción de los experimentos realizados se precisa en la Sección 4.4, y las métricas utilizadas para evaluar dichos experimentos se detallan en la Sección 4.5. Por último, la Sección 4.6 define las herramientas utilizadas para este TFG.

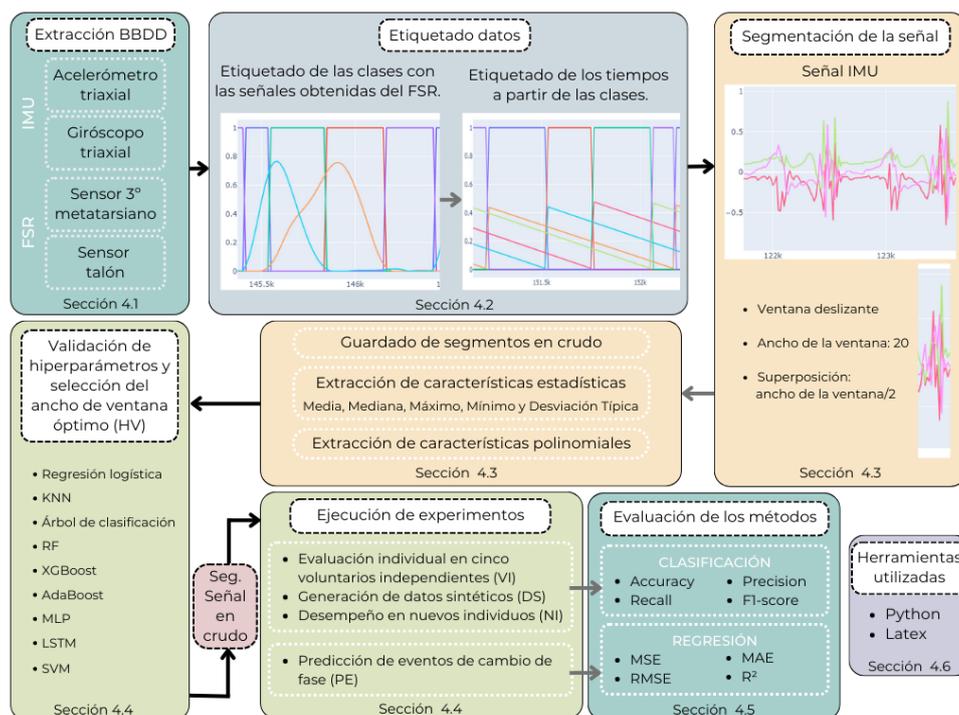


Figura 4.1: Representación del flujo de trabajo seguido.

4.1. Base de Datos

Este TFG utiliza la BD publicada en el trabajo de investigación [28], que contiene información de la marcha recogida en un ambiente urbano. Esta BD incluye medidas del movimiento de la marcha de 20 voluntarios sanos. De estos, para este TFG solo se utilizan los datos de 5 voluntarios seleccionados al azar, específicamente los voluntarios 1, 3, 4, 5 y 8. La edad, el sexo y las características antropométricas de estos cinco voluntarios elegidos se encuentran detalladas en la Tabla 4.1.

Tabla 4.1: Información antropométrica de los participantes.

ID	Edad [años]	Sexo	Peso [kg]	Altura [cm]	Altura de la cadera [cm]
1	50-59	Femenino	59	171	91
3	18-29	Masculino	80	185	112
4	18-29	Masculino	78	184	111
5	30-39	Masculino	61	172	98
8	30-39	Femenino	65	168	94
Resumen	36.8 (± 10.75)	2f, 3m	68.6 (± 9.762)	176 (± 7.906)	101.2 (± 9.731)

La BD contiene las medidas adquiridas por un conjunto de IMUs y FSRs durante diferentes pruebas de marcha. En la grabación de esta BD, a cada uno de los voluntarios se le coloca un traje de lycra con 17 IMUs, 8 FSRs en cada pie y un dispositivo de seguimiento ocular. Los sensores y sus colocaciones se muestran en la Figura 4.2.



Figura 4.2: Sensores empleados y su colocación en los voluntarios [28].

Los sensores de presión colocados en los pies son las plantillas *IEE ActiSense Smart Footwear Sensor*. Los sensores están colocados debajo del hallux, de los dedos en las

cabezas del primer, tercer y quinto metatarsiano, de la bóveda plantar y en la parte izquierda y derecha del talón. Las plantillas de presión están controladas por *electronic control units* (ECUs) que se sujetan a los zapatos de los participantes y tienen IMUs incorporadas.

Tanto los sensores inerciales situados en el traje de lycra como los sensores inerciales integrados en las plantillas de presión constan de un acelerómetro, un giróscopo y un magnetómetro triaxiales. Los datos recogidos por las plantillas de presión se muestrean con una frecuencia de 200 Hz.

Los voluntarios de la BD completan tres recorridos distintos que incluyen diferentes elementos como rampas, escaleras o pavimentos. El esquema de estos recorridos se muestra en la Figura 4.3. De esta forma, los datos incluidos en la BD son las medidas inerciales en los diferentes segmentos corporales, así como la presión en las plantas de los pies, a lo largo de los estos tres recorridos.

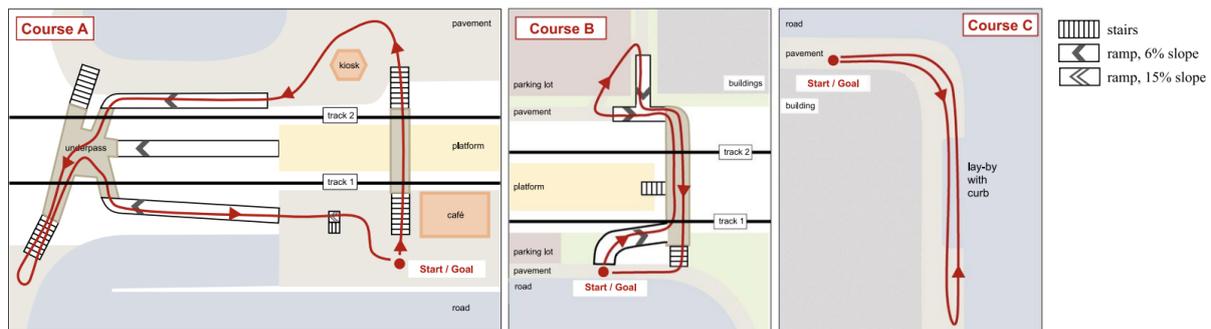


Figura 4.3: Recorridos realizados por los voluntarios [28].

Es importante destacar que los datos de marcha de la BD están recogidos en un entorno natural. Esto es relevante puesto que este trabajo analiza la marcha en un entorno no controlado. Esto hace que los resultados de este trabajo sean útiles para posibles aplicaciones futuras, ya que los datos utilizados son de entornos reales. En contraposición, si se obtuviesen a partir de datos simulados en el laboratorio, habría que estudiar su extrapolación a entornos reales.

Dado que este trabajo se centra en analizar la segmentación de la marcha normal y la estimación de sus eventos relevantes, únicamente utiliza los datos del recorrido C. Esto se debe a que es el único recorrido de la BD que no contiene rampas ni escaleras (ver Figura 4.3). Asimismo, los sensores que se emplean son los que se encuentran situados en los pies en la grabación de la BD: las plantillas que contienen los FSRs y las IMUs incorporadas a dichas plantillas.

4.2. Descripción de las señales utilizadas

En esta sección se describen las señales de los FSRs y de las IMUs. También se aborda el etiquetado de las fases de la marcha y los tiempos restantes hasta cada una de las fases.

4.2.1. Descripción y etiquetado de las señales FSR

Las señales recogidas por los FSRs se emplean como referencia para etiquetar las fases de la marcha, puesto que la BD no cuenta con esta información. En concreto, se utilizan las señales de los FSRs colocados en el tercer metatarsiano y en el talón. Un ejemplo de las señales recogidas por estos sensores se muestra en la Figura 4.4.

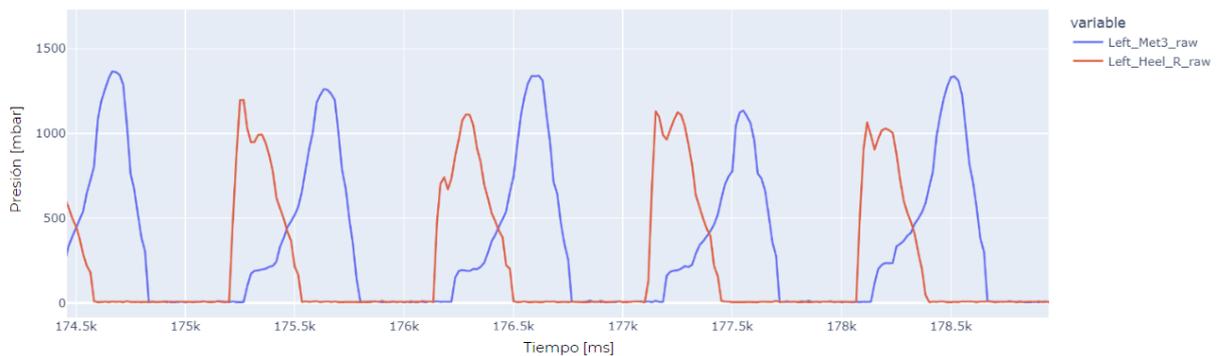
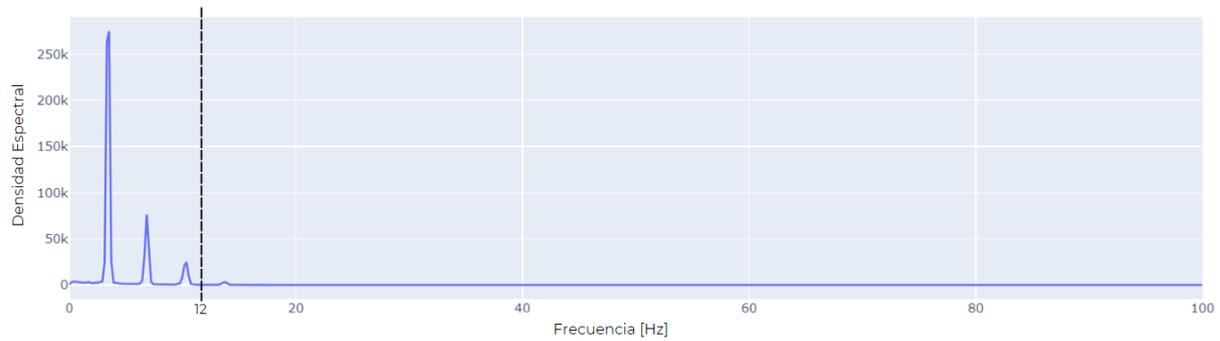


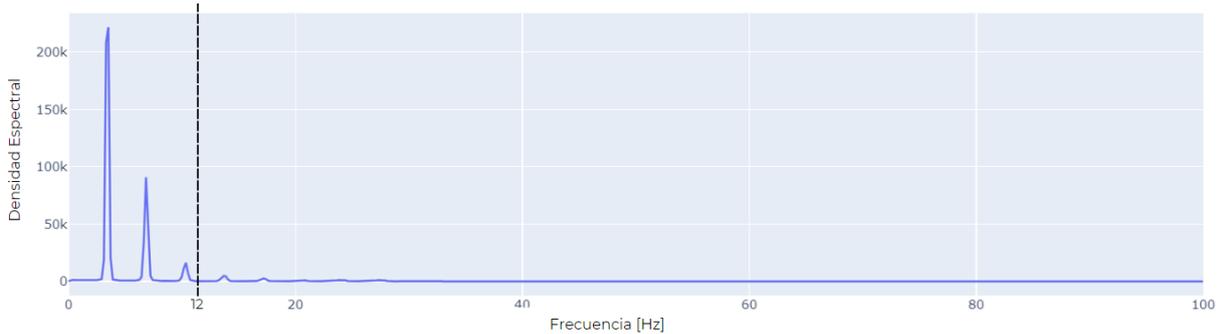
Figura 4.4: Señal recogida por los sensores de presión en el pie izquierdo del voluntario 1 mientras realiza el recorrido C.

Partiendo de un pie de referencia, en el caso de la Figura 4.4 el pie izquierdo, en rojo se indica la señal del sensor de presión colocado en el talón y en azul la señal de presión del sensor colocado en el tercer metatarsiano. La presión captada por el sensor del talón (en rojo) aumenta al inicio de un paso hasta llegar a un máximo. Esto significa que el peso del cuerpo que recaía en el pie contrario comienza a transmitirse al pie de referencia. Por su parte, llegado un momento, la presión del sensor colocado en el tercer metatarsiano (en azul) comienza a aumentar. Esto quiere decir que el peso se transfiere a la parte delantera del pie. Cuando ninguno de los sensores detecta presión significa que el pie de referencia no se encuentra apoyado en el suelo. Este es el criterio utilizado para el etiquetado de las fases de la marcha.

Debido a la importancia de estas señales recogidas por los FSRs al usarlas como referencia, y puesto que presentan ruido de alta frecuencia, en primer lugar se les aplica un filtrado. Para ello, se visualiza el periodograma de Welch de las señales de presión del metatarsiano y el talón con el objetivo de determinar la frecuencia de corte del filtro, obteniéndose lo mostrado en la Figura 4.5.



(a) Representación del periodograma de Welch de la señal recogida en el metatarsiano.



(b) Representación del periodograma de Welch de la señal recogida en el talón.

Figura 4.5: Periodograma de Welch de la señal recogida por los sensores de presión del pie izquierdo del voluntario 1 mientras realiza el recorrido C.

Como se puede apreciar en la Figura 4.5, la información relevante se encuentra en frecuencias inferiores a los 12 Hz, mientras que por encima de esta frecuencia solo se observa ruido. Por lo tanto, se utiliza un filtro paso bajo con una frecuencia de corte de 12 Hz, en concreto un filtro Butterworth de tercer orden. La representación de la respuesta al impulso del filtro Butterworth de orden 3 con una frecuencia de corte de 12 Hz se muestra en la Figura 4.6.

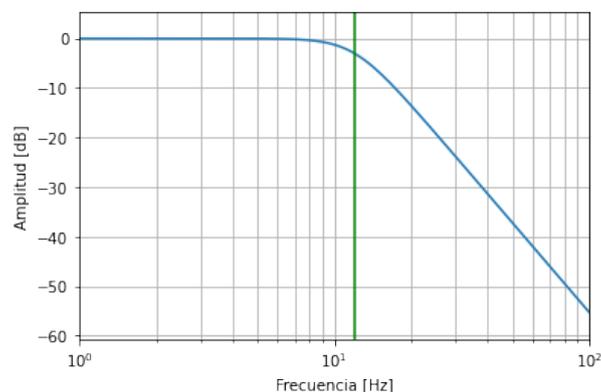


Figura 4.6: Respuesta al impulso del filtro Butterworth.

El proceso de eliminación de ruido de alta frecuencia y suavizado de las señales tiene el propósito de utilizar estas señales como base para el etiquetado de las fases de la marcha.

4.2.1.1. Etiquetado de las fases de la marcha

El etiquetado de las fases de la marcha se realiza en base a las señales recogidas por los FSRs filtradas. Para etiquetar las fases de la marcha primeramente se establece un umbral que tiene en cuenta el posible ruido inherente de los sensores. Si la señal recogida es menor que este umbral, se considera que la señal es similar a 0. Observando las gráficas de las señales de los sensores de presión filtradas se considera un umbral de 25 milibar.

Las directrices que se tienen en cuenta para el etiquetado de las fases de la marcha son las siguientes:

- Si la señal recogida por el sensor de presión colocado en el talón y la señal recogida por el sensor de presión colocado en el metatarsiano es similar a 0 (menor que el umbral de 25 milibar) se etiqueta como *Swing*, SW: el pie no está apoyado, está en fase de oscilación.
- Si la señal recogida por el sensor de presión colocado en el talón es mayor que el umbral y la señal recogida por el sensor de presión colocado en el metatarsiano es similar a 0 (menor que el umbral de 25 milibar) se etiqueta como *Heel Strike*, HS: el pie se apoya en el talón, es el contacto inicial o *load*.
- Si la señal recogida por el sensor de presión colocado en el talón y la señal recogida por el sensor de presión colocado en el metatarsiano son mayores que el umbral se etiqueta como *Foot Flat*, FF: todo el pie está apoyado en el suelo, es el fase de apoyo plano.
- Si la señal recogida por el sensor de presión colocado en el talón es similar a 0 (menor que el umbral de 25 milibar) y la señal recogida por el sensor de presión colocado en el metatarsiano es mayor que el umbral se etiqueta como *Push Off*, PO: el pie se apoya en los metatarsianos, es la fase de preoscilación.

El etiquetado final de las señales siguiendo este criterio se observa en la Figura 4.7. Las señales de presión del sensor colocado en el metatarsiano y en el talón normalizadas respecto al máximo se muestran en color naranja y azul claro respectivamente. Las etiquetas representan en azul oscuro la fase HS, en rojo la fase PO, en verde la fase FF y en morado la fase SW.

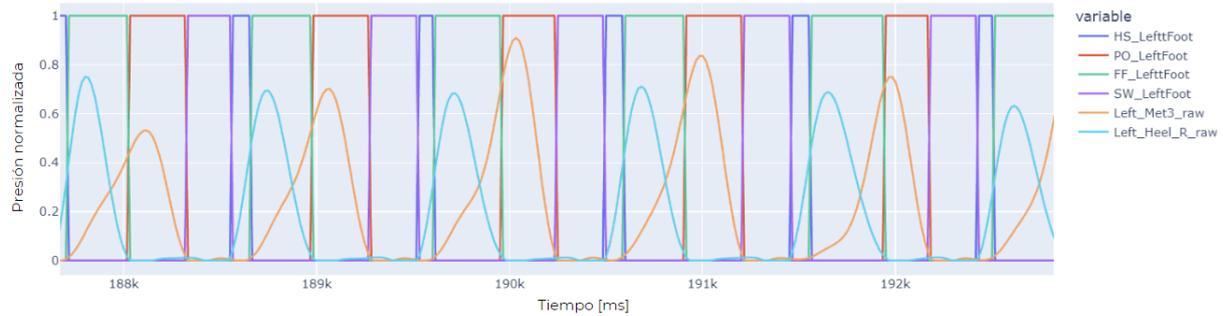


Figura 4.7: Etiquetado de la señal filtrada de los sensores de presión en el pie izquierdo del voluntario 1 mientras realiza el recorrido C.

4.2.1.2. Etiquetado de los tiempos

Otro de los objetivos de este TFG es predecir el tiempo restante desde cada instante temporal hasta cada una de las cuatro fases siguientes. El inicio de las fases de la marcha se corresponde con cada uno de los eventos relevantes de la marcha (HS, TS, HO y TO). Para predecir el tiempo restante hasta cada uno de estos eventos, se estiman los tiempos de inicio de fase utilizando las etiquetas de las fases calculadas previamente.

En primer lugar, se calculan los instantes de tiempo de cada cambio de fase. Es decir, el instante en el que comienza cada una de las fases. En la Figura 4.8 se muestran con un punto rosa los instantes en los que se inicia la fase SW, con un punto naranja los instantes en los que se inicia la fase FF, con un punto azul los instantes en los que se inicia la fase PO y con un punto verde los instantes en los que se inicia la fase HS.

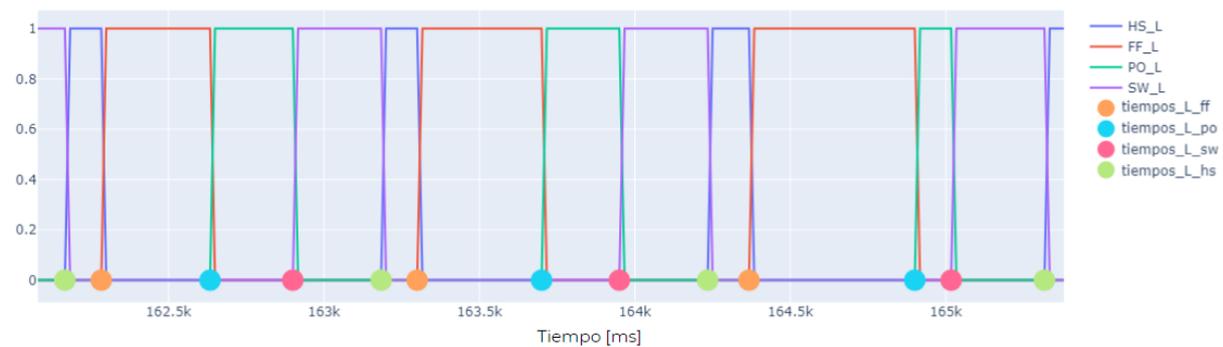


Figura 4.8: Representación de los tiempos de cambio de fase.

Una vez etiquetados estos tiempos, se calcula la diferencia desde cada instante de tiempo de la señal hasta el próximo cambio de fase. En la Figura 4.9 se muestra en verde claro un ejemplo del tiempo restante normalizado hasta el comienzo de la fase HS. Para la normalización se ha dividido cada valor entre el máximo. Debido a la presencia de pasos de aproximadamente del doble de duración, en la figura mostrada los picos máximos tienen un valor de 0.5. Para calcular el tiempo restante a las otras fases, el procedimiento es similar.

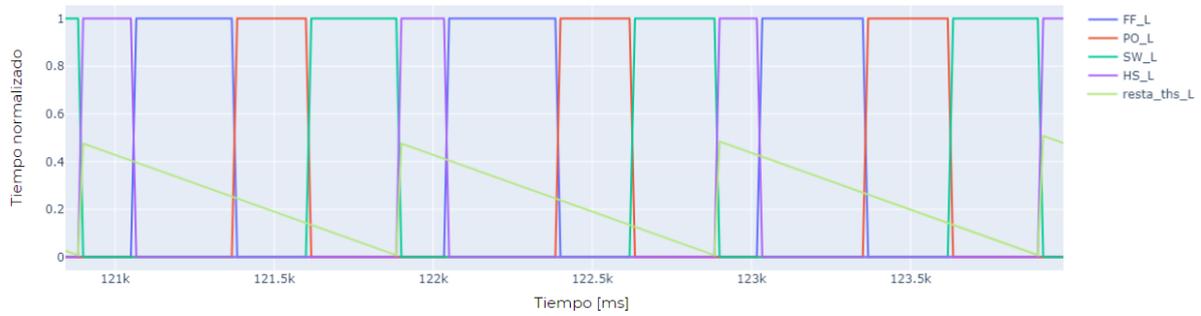


Figura 4.9: Se muestran el tiempo restante hasta HS y las etiquetas de las fases de la marcha con respecto a este tiempo.

Como se puede observar en la Figura 4.9, la señal del tiempo restante hasta el comienzo de HS es periódica y los picos coinciden con el comienzo de dicha fase. Al comenzar la fase HS el tiempo hasta la próxima HS es máximo, y este tiempo va disminuyendo hasta que termina la fase SW cuando es 0 puesto que vuelve a comenzar el ciclo de la marcha.

4.2.2. Descripción de las señales de las IMUs

La idea de este proyecto es predecir las fases de la marcha a partir de las señales recogidas de un sensor inercial colocado en el pie. Para esto se utilizan únicamente las señales del acelerómetro y del giróscopo. Estas señales son seleccionadas debido a que proporcionan información sobre los movimientos lineales y angulares de las piernas durante la marcha. Con respecto al magnetómetro, este es muy sensible a las interferencias electromagnéticas. Esto, unido al hecho de que no todos los sensores inerciales lo incluyen, hace que se opte por no utilizar esta información.

En la Figura 4.10 se muestran las señales del acelerómetro y del giróscopo. Para mejor visualización de estas, se muestran únicamente la señal del acelerómetro en el eje x y z , que se corresponden con los ejes contenidos en el plano sagital del cuerpo, y la señal del giróscopo en el eje y , perpendicular al plano sagital. Ambas señales están normalizadas entre 0 y 1.



Figura 4.10: Señal del acelerómetro en el eje x y z (medido en m/s^2) y del giróscopo (medido en dps) en el eje y , ambas normalizadas entre 0 y 1.

En la Figura 4.11 se muestra un ejemplo de las señales etiquetadas. Se puede observar que las etiquetas son correctas por la relación de la velocidad angular \vec{w} con los movimientos que ocurren en las diferentes fases:

- Durante la fase de carga (HS) se produce un pico de aceleración angular, puesto que el tobillo pasa de un ángulo definido por el apoyo único del talón al instante previo a la fase FF, donde el pie está en posición plana.
- Durante la fase de apoyo plano (FF) no se produce prácticamente aceleración angular, el pie está en horizontal.
- Previo a la fase de balanceo (SW) hay un mínimo que se corresponde con el despegue del talón (PO). Esto es debido a que es esta fase, el tobillo realiza un giro, desde que despegamos el talón hasta el último apoyo de los metatarsianos.

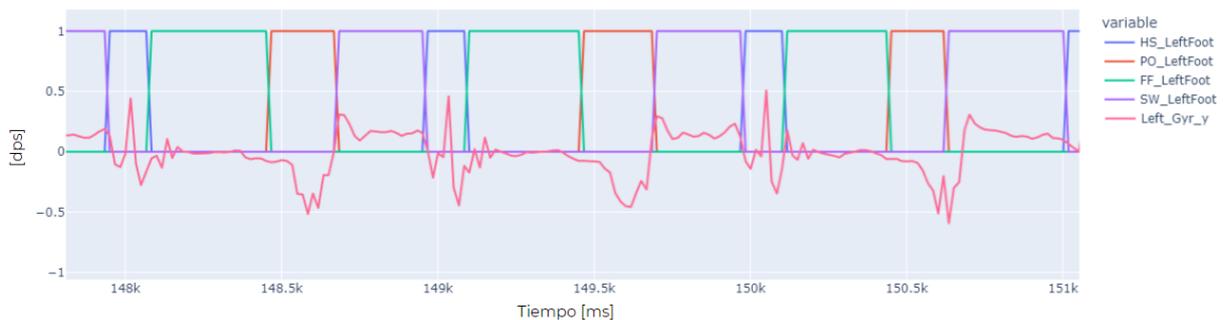


Figura 4.11: Representación de la señal del giróscopo en el eje y .

4.3. Preparación de los datos para el entrenamiento

En esta sección se describe la preparación de los datos para el posterior entrenamiento de los algoritmos de aprendizaje automático.

Una vez etiquetadas las señales, se realiza un preprocesado del *dataset* para obtener el archivo que se utiliza más adelante para entrenar los modelos de ML. Para ello, se toman ventanas deslizantes con un determinado ancho de ventana (W muestras) y una superposición (*stride*) de la mitad del ancho de la ventana.

El etiquetado de la tarea de clasificación para ventanas de 20 muestras se representa en la Figura 4.12. En esa figura se muestra en azul la fase SW, en rojo la fase PO, en verde la fase FF y en morado la fase HS.

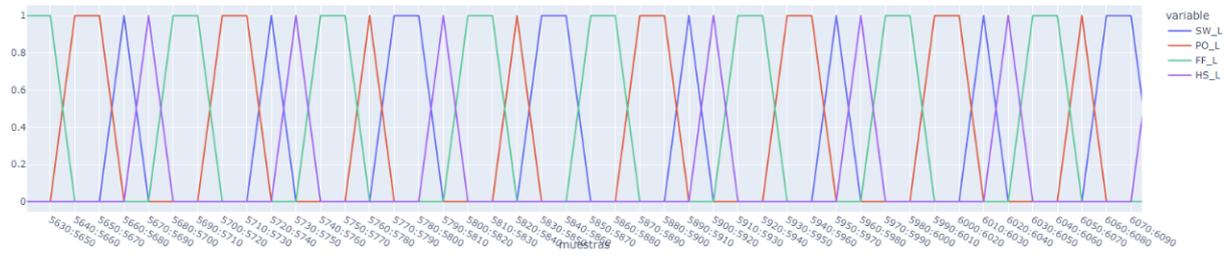


Figura 4.12: Etiquetado de la señal muestreada de los sensores de presión en el pie derecho del voluntario 1 mientras realiza el recorrido C.

Por su parte, para la tarea de regresión, cada ventana es etiquetada con los cuatro tiempos correspondientes a la última muestra, tal como se observa en la Figura 4.13. En azul se indican los tiempos restantes hasta FF, en rojo los tiempos restantes hasta PO, en verde los tiempos restantes hasta SW y en morado los tiempos restantes hasta HS.

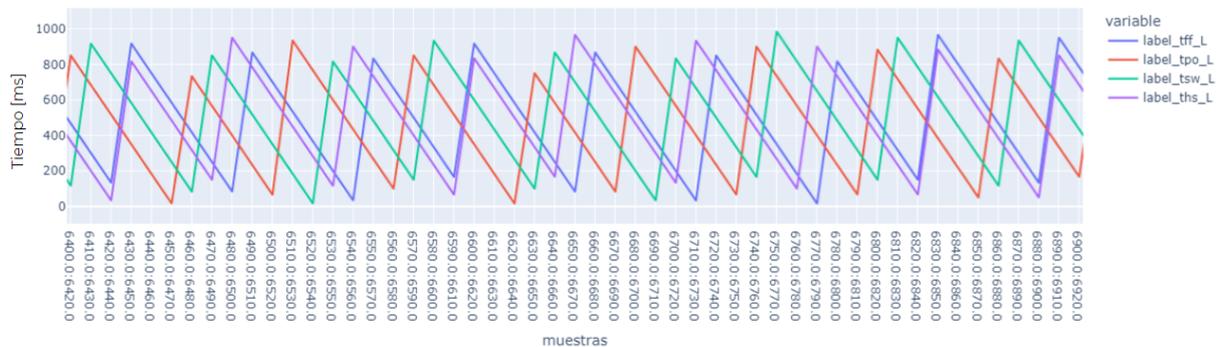


Figura 4.13: Etiquetas de los tiempos restantes hasta cada una de las fases.

Por último, se concatenan los valores de la señal del acelerómetro en el eje x , y y z y la señal del giróscopo en el eje x , y y z obteniendo para cada ventana un vector de características con una longitud de 6 veces el ancho de ventana que es lo que se utilizará para entrenar los diferentes modelos. Cada una de estas ventanas tiene asociada una etiqueta correspondiente a la fase de la marcha, y cuatro etiquetas correspondientes a los tiempos restantes hasta cada uno de los eventos a predecir. Además, se concatenan los datos del pie izquierdo con los datos del pie derecho para así obtener una mayor cantidad y variabilidad de los datos; y se divide el conjunto de datos en entrenamiento (80 %) y test (20 %). También se utiliza un tercio de los datos de entrenamiento para realizar validación cruzada en los modelos que requieren *fitting* de hiperparámetros, como se explica en la Sección 4.4.1.

A partir de las ventanas deslizantes etiquetadas se estudian dos posibles líneas de investigación basadas en el uso de las señales en crudo o en la extracción de características a partir de estas señales. Ambas líneas se describen en las Secciones 4.3.1 y 4.3.2, respectivamente.

4.3.1. Datos en crudo

La primera línea de investigación consiste en realizar el entrenamiento de los algoritmos de ML con los datos de las señales en crudo. Para ello, se utiliza la matriz de datos de dimensión $N \times (6 \cdot W)$. El valor de N representa el número de ventanas, y se calcula según la Ecuación (4.1):

$$N = 2 \cdot 2 \cdot \frac{L}{W} \quad (4.1)$$

donde L corresponde a la longitud total de la señal y W al ancho de ventana. La cantidad de ventanas se determina multiplicando por dos la división de L entre W debido al *stride* establecido al 50%. Además, este valor se multiplica de nuevo por dos debido a la inclusión de datos tanto del pie derecho como del pie izquierdo. Cada una de las filas de esta matriz se encuentra etiquetada con la fase de la marcha correspondiente.

4.3.2. Análisis y Extracción de Características

La segunda alternativa consiste en extraer características de cada ventana de la señal y entrenar los modelos con esta información. La extracción de características es un proceso en el que se utilizan métodos para transformar los datos originales a un espacio de menor dimensión, con el objetivo de obtener la información más importante y relevante. En numerosas ocasiones se utiliza este método para el procesamiento de señales, puesto que esto ayuda a reducir la complejidad de las mismas. Estas características pueden ser de diferentes tipos, como características temporales, frecuenciales, estadísticas o morfológicas, entre otras. En este TFG se utilizan características estadísticas y polinomiales.

4.3.2.1. Extracción de Características Estadísticas

Las características estadísticas empleadas son las siguientes: media, mediana, máximo, mínimo y desviación típica. La media se calcula mediante la suma de todos los datos divididos entre el número de datos, según se muestra en la Ecuación (4.2):

$$\bar{X} = \frac{1}{N} \sum_{i=1}^N x_i. \quad (4.2)$$

La mediana es el valor medio de todos los datos ordenados. Su cálculo depende de si el número de datos es impar o par, tal como se muestra en las Ecuaciones (4.3) y (4.4) respectivamente:

$$Me = x_{\frac{N+1}{2}}, \quad (4.3)$$

$$Me = \frac{x_{\frac{N}{2}} + x_{\frac{N}{2}+1}}{2}. \quad (4.4)$$

El máximo y el mínimo se asocian con el mayor y el menor valor dentro de un conjunto de valores. Por último, la desviación típica es una medida de la dispersión de un conjunto de datos. Indica cuanto se alejan las observaciones individuales del conjunto de datos de la media aritmética del conjunto, tal como muestra la Ecuación (4.5):

$$\sigma = \sqrt{\frac{1}{N} \sum_i^N (x_i - \bar{X})^2}. \quad (4.5)$$

Las características estadísticas extraídas varían entre las diferentes fases de la marcha. Para analizar la distribución de estas características en cada fase, se realiza un estudio observacional mediante gráficas *boxplot*. Sin embargo, debido a la gran cantidad de gráficas que se generarían al mostrar los *boxplot* de los tres ejes del acelerómetro y los tres ejes del giróscopo, en este documento se opta por mostrar la norma de la aceleración lineal y la velocidad angular para cada estadístico en cada fase de la marcha. Estas representaciones se muestran en las Figuras 4.14, 4.15, 4.16, 4.17 y 4.18.

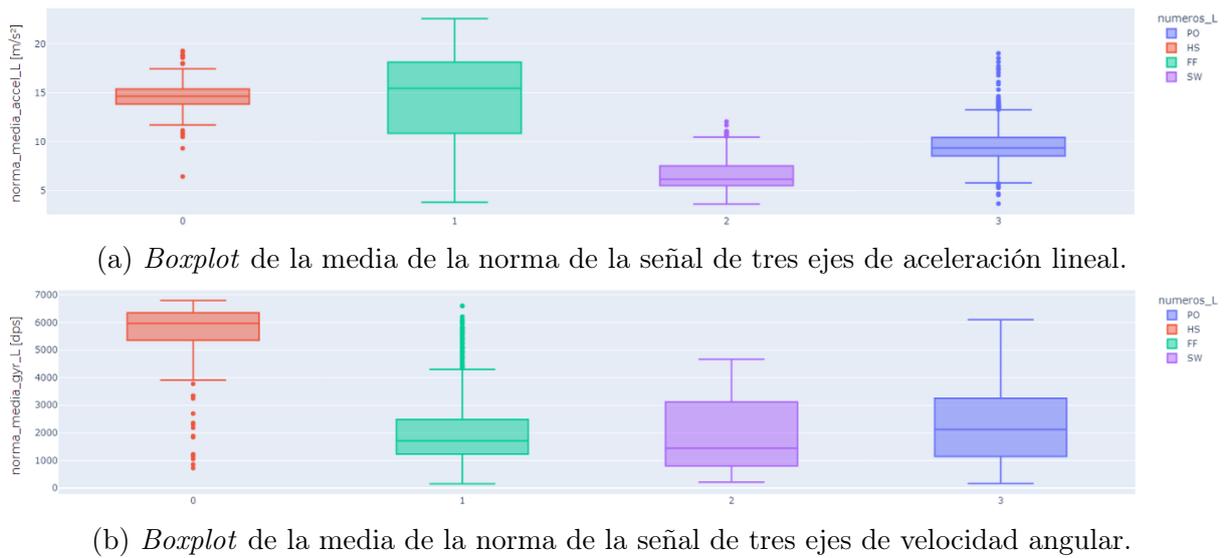


Figura 4.14: *Boxplot* de la media de la norma de las señales de tres ejes de aceleración lineal y velocidad angular.

Como se puede observar en la Figura 4.14 la media de la norma de la señal del acelerómetro permite diferenciar PO y SW del resto. Por su parte, la media de la norma de la señal del giróscopo es útil para diferenciar HS. Sin embargo, esto no es suficiente para separar todas las clases de manera individual.

Los *boxplots* de la desviación típica, el máximo y el mínimo y la mediana de la norma de la aceleración lineal y la velocidad angular se muestran en las Figuras 4.15, 4.16, 4.17 y 4.18, respectivamente. Puede observarse que con estas características se diferencian las fases SW, FF y HS.

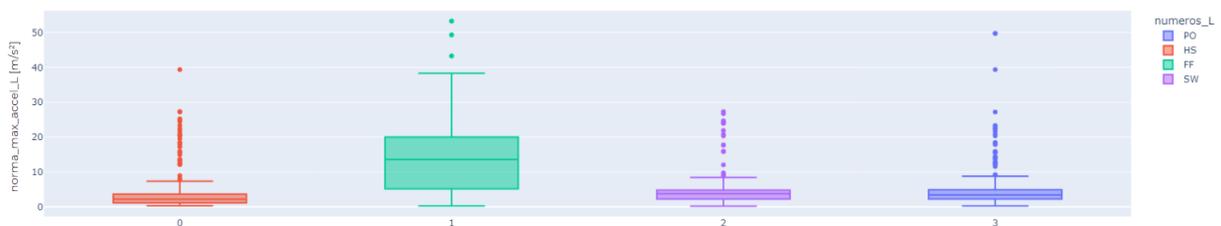


(a) *Boxplot* de la desviación típica de la norma de la señal de tres ejes de aceleración lineal.

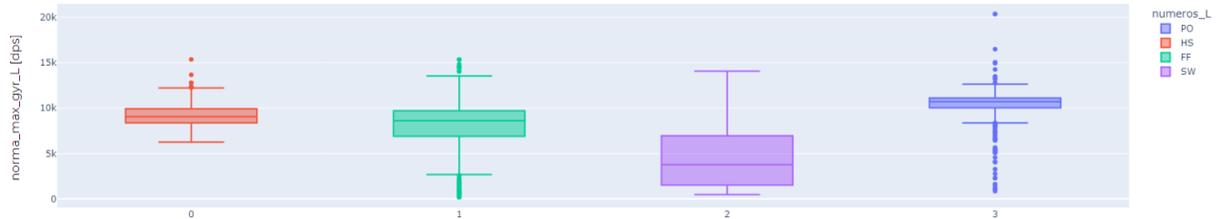


(b) *Boxplot* de la desviación típica de la norma de la señal tres ejes de velocidad angular.

Figura 4.15: *Boxplot* de la desviación típica de la norma de las señales de tres ejes de aceleración lineal y velocidad angular.



(a) *Boxplot* del máximo de la norma de la señal de tres ejes de aceleración lineal.

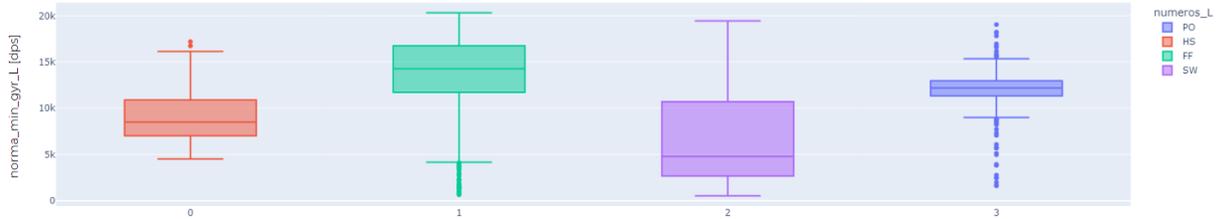


(b) *Boxplot* del máximo de la norma de la señal de tres ejes de velocidad angular.

Figura 4.16: *Boxplot* del máximo de la norma de las señales de tres ejes de aceleración lineal y velocidad angular.



(a) Boxplot del mínimo de la norma de la señal de tres ejes de aceleración lineal.



(b) Boxplot del mínimo de la la norma de la señal de tres ejes de velocidad angular.

Figura 4.17: *Boxplot* del mínimo de la norma de las señales de tres ejes de aceleración lineal y velocidad angular.

(a) *Boxplot* de la mediana de la norma de la señal de tres ejes de aceleración lineal.(b) *Boxplot* de la mediana de la norma de la señal de tres ejes de velocidad angular.

Figura 4.18: *Boxplot* de la mediana de la norma de las señales de tres ejes de aceleración lineal y velocidad angular.

La Figura 4.15 muestra que la desviación típica de la norma de las señales del acelerómetro y del giróscopo permiten diferenciar SW del resto. En la Figura 4.16 se observa como el máximo de la norma de la señal del acelerómetro permite diferenciar FF y el máximo de la norma de la señal del giróscopo SW. Como muestra la Figura 4.17, el mínimo de la norma de las señales del acelerómetro y del giróscopo es bueno para diferenciar las fases FF y SW. Por último, la mediana de la norma de la señal del giróscopo permite distinguir HS del resto, como se observa en la Figura 4.18.

Con estas características se crea una matriz de datos. Para ello se concatenan las características de cada ventana en una lista de 30 valores correspondientes a la media,

desviación típica, máximo, mínimo y mediana del acelerómetro y el giróscopo en cada uno de los tres ejes. La matriz de datos creada tiene una dimensión de $N \times 30$, siendo N el número de ventanas calculado según la Ecuación (4.1). Además cada ventana se encuentra etiquetada con la fase de la marcha correspondiente.

4.3.2.2. Extracción de Características Polinomiales:

La extracción de características polinomiales es una técnica que permite ajustar datos a una curva utilizando funciones polinomiales de diferentes grados. La idea es encontrar los coeficientes del polinomio de grado n que mejor se ajusta a los datos. Es importante la correcta selección del grado del polinomio, puesto que un grado de polinomio demasiado alto generará demasiadas características aumentando por tanto la complejidad del modelo; mientras que un grado de polinomio demasiado bajo puede no ser suficiente para captar toda la complejidad de los datos. Una vez se han generado estas características, pueden ser utilizadas para entrenar modelos de aprendizaje automático.

En este TFG, para realizar la extracción de características polinomiales se utiliza la herramienta *Polyfit*. *Polyfit* es una herramienta utilizada para ajustar una curva polinómica a un conjunto de datos mediante regresión polinómica. Se elige un polinomio de grado 3, y se entrenan los modelos con los coeficientes de dicho polinomio. La matriz de datos con la que se entrenan los modelos tiene una dimensión de $N \times 4$ (siendo N el número de ventanas calculado según la Ecuación (4.1)), correspondiente a los cuatro coeficientes de la curva polinómica ajustada. Los *boxplot* correspondientes a cada uno de los coeficientes para cada fase de la marcha se muestran en la Figura 4.19.

(a) *Boxplot* del coeficiente 0 de cada una de las clases.(b) *Boxplot* del coeficiente 1 de cada una de las clases.(c) *Boxplot* del coeficiente 2 de cada una de las clases.(d) *Boxplot* del coeficiente 3 de cada una de las clases.Figura 4.19: *Boxplot* de los coeficientes 0, 1, 2 y 3 obtenidos con *Polyfit*.

En general, estos coeficientes no proporcionan una discriminación clara entre clases, lo que sugiere que los resultados obtenidos con esta aproximación no serán óptimos. En términos generales, las medias de los coeficientes de las cuatro clases son similares para todos los coeficientes, excepto para el coeficiente 3, donde se observa una mayor capacidad para distinguir la clase HS del resto.

4.4. Descripción de los experimentos

En esta sección se presentan los detalles de los experimentos realizados en este TFG. El primer experimento, denominado HV, se aborda en la Sección 4.4.1, y se enfoca en la determinación de los hiperparámetros y el ancho de ventana óptimo. El segundo experimento, identificado como VI, se describe en la Sección 4.4.2, y se centra en la evaluación individual de cinco voluntarios independientes. El tercer experimento, designado como

DS, se lleva a cabo para mejorar el rendimiento de los modelos mediante la estandarización y creación de datos sintéticos, y se encuentra descrito en la Sección 4.4.3. El cuarto experimento, denominado NI, se realiza para evaluar el desempeño en nuevos individuos, y se encuentra definido en la Sección 4.4.4. Finalmente, el último experimento, PE, se detalla en la Sección 4.4.5 y consiste en la predicción de los eventos de cambio de fase.

Todos estos experimentos quedan resumidos en la Tabla 4.2. Esta tabla categoriza los experimentos según tarea a resolver, voluntarios implicados, conjunto de datos utilizado, algoritmos de ML empleados, ancho de ventana definido y uso de estandarización y SMOTE.

Tabla 4.2: Resumen de los experimentos realizados.

Exp	Tarea	Voluntario	Conjunto de datos	Modelos	W	Estandarización y SMOTE
HV	Clasificación	1	Datos en crudo Ext. Caract Estadísticas Ext. Caract. Polinomiales	R. Logística KNN Árbol RF	20	No
				AdaBoost XGBoost MLP LSTM SVC		
W óptimo	Clasificación	1	Datos en crudo	RF	[6-20]	No
VI	Clasificación	1,3,4,5,8	Datos en crudo	R. Logística KNN Árbol RF	8	No
				AdaBoost XGBoost MLP LSTM SVC		
DS	Clasificación	1	Datos en crudo	R. Logística KNN Árbol RF	8	Si
				AdaBoost XGBoost MLP LSTM SVC		
NI	Clasificación	V. Cruzada 1, 3, 4, 5, 8	Datos en crudo	R. Logística KNN Árbol RF	8	Si
				AdaBoost XGBoost MLP LSTM SVC		
PE	Regresión + Clasificación	1	Datos en crudo	R. Lineal KNN Árbol RF	8	Estandarización
				AdaBoost XGBoost MLP LSTM SVR		

4.4.1. Selección de hiperparámetros y ancho de ventana óptimo

El objetivo de este primer experimento, HV, es determinar que entradas a los modelos de ML, y que ancho de ventana proporcionan las métricas de clasificación más altas en segmentación. La configuración que proporcione mayores métricas será la adecuada para segmentar y estimar los eventos que componen el ciclo de la marcha. Dado que se utilizan tres conjuntos distintos de entrada y la ventana es variable, en primer lugar se establece un ancho de ventana inicial para reducir las posibles combinaciones y limitar el espacio de búsqueda. De esta forma, en la Figura 4.20 se presenta un resumen del flujo de trabajo seguido para este experimento y su repercusión en los experimentos posteriores:

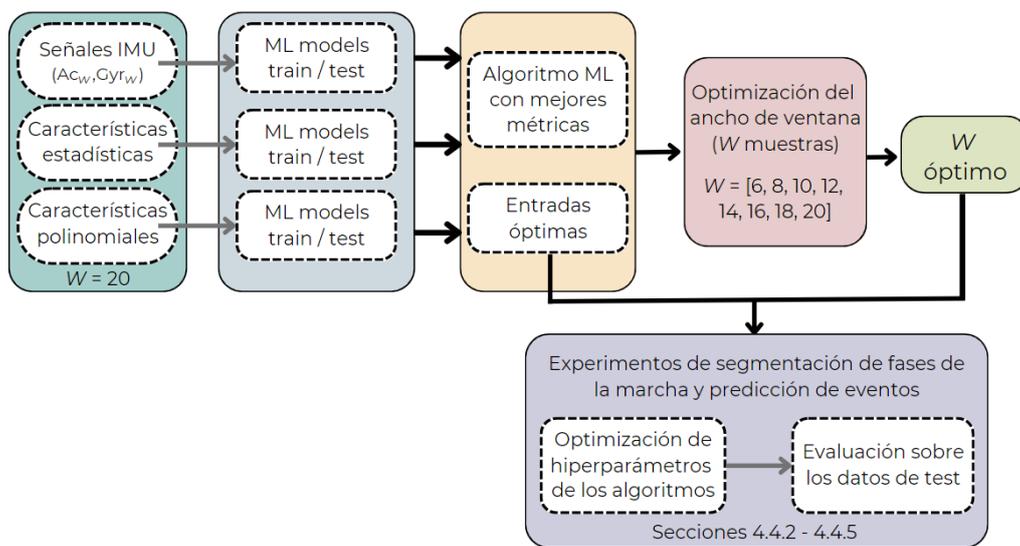


Figura 4.20: Flujo de trabajo seguido para realizar el experimento HV.

En primer lugar se fija un ancho de ventana y se determina un desplazamiento de la mitad de este ancho de ventana. Con ello, se optimizan los hiperparámetros y se entrenan los modelos de ML con los datos de los tres conjuntos: datos en crudo, características estadísticas y características polinomiales del primer voluntario. Cada uno de estos conjuntos de datos se divide en entrenamiento y test, en una proporción de 80% y 20%, respectivamente. Con el conjunto de entrenamiento se realiza la optimización de hiperparámetros, para lo que se utiliza la técnica de validación cruzada *k-fold*, descrita en la Sección 3.7.

A continuación, se describen las configuraciones de los modelos empleados y la determinación de sus hiperparámetros. En todos los experimentos se realiza la optimización de los mismos hiperparámetros.

En el caso del modelo de **regresión logística** se emplean los parámetros predefinidos por la función *LogisticRegression* de la librería *sklearn*. El único parámetro que se configura es el tipo de regularización, seleccionándose la regularización *Ridge* L_2 .

En lo que respecta al problema de regresión, se emplea la **regresión lineal**. Para el entrenamiento de este modelo no se lleva a cabo un proceso de validación cruzada para determinar los hiperparámetros del modelo. La configuración del modelo se basa en los parámetros predefinidos por la función *LinearRegression* de la librería *sklearn*.

Para el modelo **SVM**, tampoco se seleccionan los hiperparámetros con validación cruzada, sino que se establecen los parámetros predefinidos por la función de *sklearn*, entre los que está el *kernel* gaussiano.

En el ajuste de hiperparámetros del modelo **KNN**, para estimar el valor de k , se dividen los datos de entrenamiento en entrenamiento y validación utilizando validación cruzada de tipo k -fold con 3 folds. Se utiliza un tercio de los datos de entrenamiento para realizar la validación cruzada. Se evalúan valores de k desde 1 vecino hasta 41 vecinos con un intervalo de dos unidades. En la Figura 4.21 se observa la determinación del hiperparámetro k de un experimento ejemplo. Se toma el valor de k para el cual hay un mejor resultado en la métrica del área bajo la curva (AUC), en ese caso, se establecería $k = 7$ vecinos.

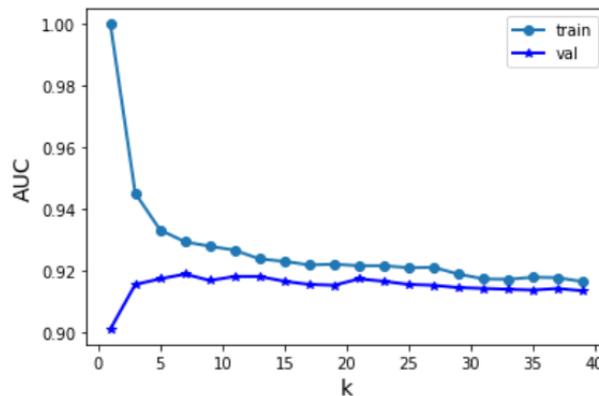


Figura 4.21: Determinación del valor de k mediante validación cruzada.

Por otra parte, para aplicar el modelo de **árbol de decisión**, al igual que para KNN, se utiliza validación cruzada de tipo k -fold con 3 folds para estimar el hiperparámetro. En este caso, el hiperparámetro a definir es la máxima profundidad del árbol, es decir, el número máximo de niveles que tiene el árbol. En la Figura 4.22 se muestra la determinación del mejor hiperparámetro de un experimento ejemplo. Se evalúan tamaños de árbol desde un nivel hasta 10 niveles. Al igual que en KNN, el mejor hiperparámetro será aquel que maximice el valor de AUC, en este caso la máxima profundidad sería de 6 niveles.

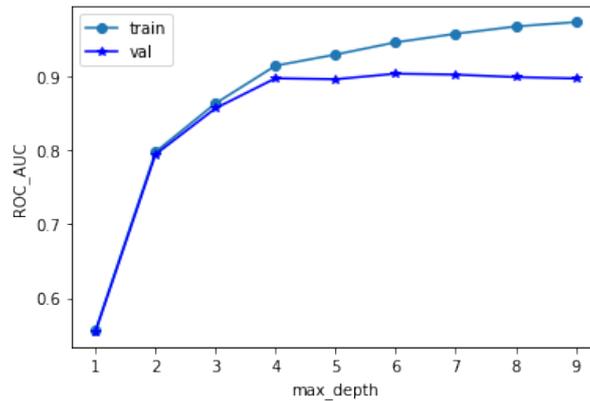


Figura 4.22: Determinación de la máxima profundidad del árbol mediante validación cruzada.

Para **RF**, el hiperparámetro a definir es el número de árboles utilizados para el ensemble, para lo que se utiliza validación cruzada de tipo k -fold con 3 folds. En la Figura 4.23 se muestra el cálculo del valor del hiperparámetro para un experimento ejemplo. Se evalúan valores del hiperparámetro desde un único árbol hasta 100 árboles con intervalos de 10 unidades. Como en los anteriores modelos, el objetivo será maximizar el valor de AUC, obteniendo en este ejemplo un valor del hiperparámetro de 71 árboles de decisión.

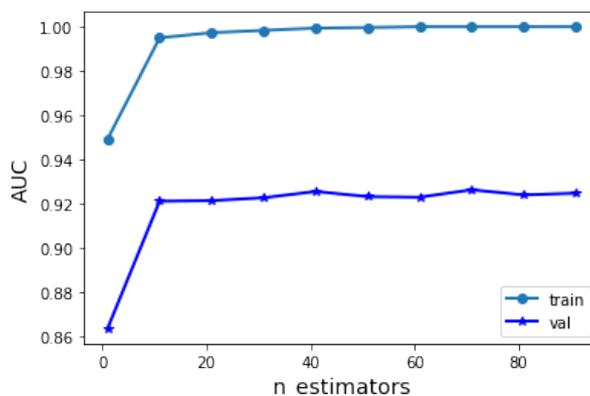


Figura 4.23: Determinación del número de árboles mediante validación cruzada.

Para el modelo **XGBoost**, con el fin de simplificar, no se definen hiperparámetros. Se entrena el modelo con los parámetros predefinidos entre los cuales está el número máximo de árboles (100 árboles) y la profundidad máxima de cada árbol (3 niveles).

Al igual que en el caso de **XGBoost**, en el modelo **AdaBoost** no se definen hiperparámetros para simplificar el proceso. En su lugar, el modelo se entrena utilizando los parámetros predefinidos, que en el caso del número máximo de árboles es 50 árboles.

Para el modelo **MLP** se define como hiperparámetro el número de neuronas ocultas, y se trata de maximizar el valor de la tasa de acierto mediante validación cruzada de tipo k -fold con 3 folds. Se evalúan valores del hiperparámetro de entre 2 y 152 neuronas

ocultas con intervalos de 10 unidades, distribuidas en una única capa. En la Figura 4.24 se muestra la estimación del número de neuronas ocultas para un experimento ejemplo.

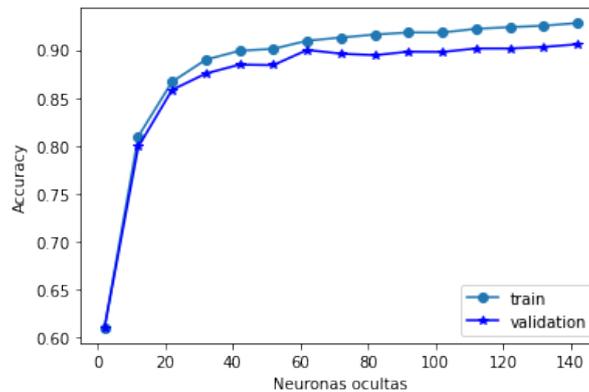


Figura 4.24: Determinación del número de neuronas ocultas mediante validación cruzada.

En relación a la red **LSTM**, se crea una arquitectura muy sencilla que consta de una capa de entrada, una capa oculta, una capa de *dropout* al 20% y una capa de salida. Durante el entrenamiento, la capa de *dropout* desconecta el 20% de las neuronas para que no sean entrenadas, simplificando así el modelo y reduciendo el *overfitting*.

En el caso del problema de clasificación, la capa de salida se compone de 4 neuronas con una función de activación *softmax*. La estructura de la red LSTM para clasificación se muestra en la Figura 4.25.

La red es entrenada con 50 épocas y un tamaño de batch de 128. Un ejemplo del entrenamiento de esta red en uno de los experimentos de clasificación se muestra en la Figura 4.26.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 120, 64)	16896
lstm_1 (LSTM)	(None, 32)	12416
dropout (Dropout)	(None, 32)	0
dense (Dense)	(None, 4)	132

=====
Total params: 29,444
Trainable params: 29,444
Non-trainable params: 0

Figura 4.25: Estructura de la red LSTM para clasificación.

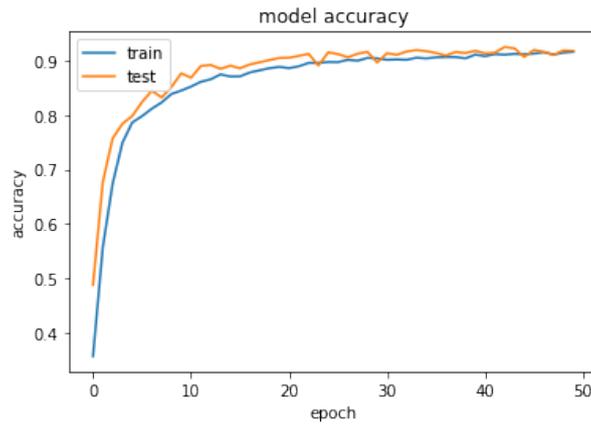


Figura 4.26: Ejemplo de entrenamiento de la red LSTM para clasificación.

En el caso del problema de regresión, la estructura de la red es similar, excepto la capa de salida que en este caso contiene una única neurona con una función de activación lineal. Debido a que el problema a resolver es más complejo, la red LSTM es entrenada con 100 épocas y un tamaño de batch de 256.

Todos los modelos de clasificación descritos se entrenan con tres conjuntos de datos distintos: los datos en crudo, las características estadísticas extraídas de la ventana y las características polinomiales del polinomio de tercer grado que ajusta los valores de la ventana. Con este experimento se obtienen métricas de clasificación de cada modelo para los tres conjuntos de datos.

En base a las métricas reportadas, se determina la aproximación más adecuada considerando el conjunto de datos de entrada y el modelo de ML. Utilizando los datos de entrada óptimos en el modelo más adecuado, se determina el ancho de ventana óptimo utilizado en la segmentación de las señales IMU. Para ello, se estudian tamaños de ventana de entre 6 y 20 muestras, con intervalos de dos unidades y con un salto de ventana del 50 % del ancho de ventana. Como resultado de este experimento, se identifican el conjunto de entradas más adecuado y el tamaño de ventana óptimo para el estudio de segmentación de la marcha, como se muestra en la Figura 4.20. El conjunto de datos determinado, junto con el tamaño de ventana óptimo, se utiliza en todos los experimentos posteriores.

4.4.2. Evaluación individual en cinco voluntarios independientes

Con el propósito de evaluar el rendimiento de los modelos en la segmentación de las fases de la marcha en datos de diferentes voluntarios, se realiza un experimento basado en la evaluación independiente, VI. Esta evaluación se lleva a cabo en cinco voluntarios distintos. Para ello, se utilizan todos los modelos planteados y cada uno de ellos se entrena utilizando los datos correspondientes a cada uno de los cinco voluntarios seleccionados.

Para cada voluntario se dividen los datos en conjuntos de entrenamiento y test con una proporción de 80 % y 20 %, respectivamente. En aquellos modelos que requieren ajuste de hiperparámetros, estos se ajustan tal y como se describe en la Sección 4.4.1. Finalmente, se evalúa el rendimiento de cada modelo utilizando los datos pertenecientes al conjunto de test de cada voluntario. Los resultados mostrados para este experimento hacen referencia al rendimiento de los modelos para cada uno de los voluntarios, así como al promedio del rendimiento en todos los voluntarios.

4.4.3. Generación de datos sintéticos

La base de datos presenta un desequilibrio significativo entre las clases. Como se puede apreciar, el recuento de ventanas por fase de la marcha, representado en la Figura 4.27 en color azul, evidencia un desequilibrio entre las clases. Los algoritmos de ML, requieren que las clases se encuentren balanceadas para así lograr un rendimiento óptimo.

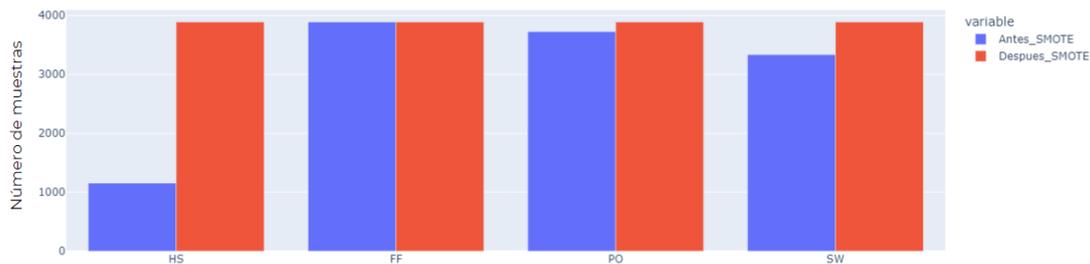


Figura 4.27: Número de muestras de cada fase de la marcha antes y después de aplicar el algoritmo SMOTE.

La distribución mostrada se debe a que la fase de contacto inicial (HS) tiene menor duración; sin embargo, las fases con mayor duración son apoyo medio (FF) y preoscilación (PO).

Con el objetivo de balancear las clases, en este experimento, DS, se plantea la creación de datos sintéticos que permitan balancear las clases de un voluntario. Para ello se emplea el algoritmo SMOTE, que permite equilibrar las clases logrando el mismo número de muestras para cada una de las cuatro clases. La diferencia entre el número de clases antes y después de aplicar SMOTE se refleja en la Figura 4.27. En esta figura, se representa en rojo el recuento del número de ventanas de cada fase después de aplicar el algoritmo SMOTE.

Tras aplicar el algoritmo SMOTE, el número de ventanas en cada clase se iguala al número de ventanas correspondientes a la fase con mayor cantidad de muestras. Es decir, en el conjunto de datos balanceado, cada fase contiene un número de ventanas equivalente al número de ventanas de la clase FF, puesto que esta fase es la que contenía mayor cantidad de datos.

En este experimento se emplean datos en crudo del voluntario 1, para así poder comparar los resultados obtenidos en el experimento anterior con el primer voluntario. Para ello, se entrenan todos los modelos planteados y se dividen los datos en entrenamiento y test en una proporción de 80 % y 20 %, respectivamente. Para los modelos que requieren optimización de hiperparámetros, estos se optimizan como se detalla en la Sección 4.4.1. Por último, el rendimiento de cada uno de los modelos se evalúa en el conjunto de test.

4.4.4. Desempeño en nuevos individuos

Con el objetivo de ver cómo de extrapolables son los modelos ya entrenados a la marcha de un voluntario nuevo, se realiza una validación cruzada “*leave-one-out*” utilizando cinco voluntarios. Para este experimento, denominado NI, se emplean los datos en crudo de cinco voluntarios, en concreto los voluntarios 1, 3, 4, 5 y 8. Se realizan cinco iteraciones en las que se entrenan todos los modelos planteados con los datos de cuatro voluntarios, dejando aparte el voluntario restante. En cada iteración, se selecciona un voluntario diferente para ser excluido.

Para el entrenamiento de los algoritmos en este experimento, se estandarizan las variables con el objetivo de tener la certeza de que se encuentran en el mismo rango para todos los voluntarios; y se utiliza SMOTE para balancear las clases a predecir. Para los modelos que requieren ajuste de hiperparámetros, estos se optimizan como se detalla en la Sección 4.4.1. Finalmente, se evalúa el rendimiento de los modelos utilizando los datos del voluntario independiente que no fue utilizado en el proceso de entrenamiento. Los resultados que se obtienen en este experimento se muestran como un promedio de las cinco iteraciones realizadas.

4.4.5. Predicción de eventos de cambio de fase

En este experimento, PE, con el objetivo de predecir el momento en el que ocurren los principales eventos que componen el ciclo de la marcha, y segmentar con ello la marcha, se estima el tiempo restante hasta cada una de las fases. Para esta aproximación inicial, se utilizan exclusivamente los datos en crudo del voluntario 1. Estos datos se estandarizan y se dividen en entrenamiento y test en una proporción de 80 % y 20 %, respectivamente.

Para el entrenamiento se emplean los modelos de regresión con las configuraciones descritas en la Sección 4.4.1. Finalmente, el rendimiento de los modelos de regresión se evalúa con el conjunto de test. Los resultados de este experimento se muestran mediante las métricas referidas a la diferencia entre los tiempos estimados para el cambio de fase y los tiempos reales.

Con el fin de realizar una comparativa con los experimentos previos, una vez obtenidas las predicciones de los tiempos restantes hasta cada una de las fases, se utiliza un sistema de reglas. Este sistema de reglas permite clasificar las observaciones en cuatro clases: HS, FF, PO, SW. Las reglas de clasificación en función de los tiempos a las fases de la marcha son las siguientes:

- Si al comparar los cuatro tiempos predichos se determina que el mayor corresponde al tiempo restante hasta la fase HS, se concluye que el voluntario se encuentra en la fase HS actualmente. Por lo tanto, el mayor tiempo registrado representa el restante hasta la próxima HS.
- Si al comparar los cuatro tiempos predichos se determina que el mayor corresponde al tiempo restante hasta la fase FF, se concluye que el voluntario se encuentra en la fase FF actualmente. Por lo tanto, el mayor tiempo registrado representa el restante hasta la próxima FF.
- Si al comparar los cuatro tiempos predichos se determina que el mayor corresponde al tiempo restante hasta la fase PO, se concluye que el voluntario se encuentra en la fase PO actualmente. Por lo tanto, el mayor tiempo registrado representa el restante hasta la próxima PO.
- Si al comparar los cuatro tiempos predichos se determina que el mayor corresponde al tiempo restante hasta la fase SW, se concluye que el voluntario se encuentra en la fase SW actualmente. Por lo tanto, el mayor tiempo registrado representa el restante hasta la próxima SW.

Es importante destacar que este experimento representa una primera aproximación, por lo que el sistema de reglas solo se aplica a las predicciones generadas por los dos modelos de regresión que logran mejores métricas en términos de rendimiento.

4.5. Métricas de evaluación

En esta sección se detallan las métricas utilizadas para la evaluación de los modelos de clasificación y regresión. Las métricas de clasificación se detallan en la Sección 4.5.1 y las métricas de regresión se detallan en la Sección 4.5.2.

4.5.1. Métricas de Clasificación

Las métricas empleadas en este trabajo se obtienen a partir de a matriz de confusión de las clasificaciones. La matriz de confusión binaria es una representación matricial de

los resultados de las predicciones obtenidas de un modelo de clasificación binario sobre un conjunto de datos cuyos valores reales son conocidos [73], tal como se muestra en la Tabla 4.3. Lo que representa en este trabajo es cómo se clasifican las fases de la marcha relacionando las predicciones obtenidas con las etiquetas.

Tabla 4.3: Matriz de confusión binaria.

		Predicción		Total
		Positivo	Negativo	
Real	Positivo	TP	FN	$TP + FN$
	Negativo	FP	TN	$FP + TN$
	Total	$TP + FP$	$FN + TN$	N

Cada uno de los elementos de la matriz se clasifican según una de estas cuatro posibilidades. En primer lugar, *True Positive* (TP) se refiere al caso en que la fase de la marcha real se corresponde con la fase de la marcha estimada. En segundo lugar, *True Negative* (TN) se corresponde al caso en que la ventana no se corresponde con la fase estudiada y se clasifica como cualquiera de las otras fases. En tercer lugar, *False Positive* (FP), hace referencia al caso en el que se ha estimado la fase que se está evaluando pero realmente se trata de otra fase. Por último, *False Negative* (FN) se corresponde al caso en el que se estima una de las fases que no se está evaluando, cuando en realidad se trata de la fase a evaluar.

A partir de los elementos de la matriz de confusión, se obtienen métricas que permiten evaluar el rendimiento de los modelos de segmentación de la marcha. Algunas de estas métricas son las siguientes: *Accuracy* (Acc), *Recall* (Rec), *Precision* ($Prec$) y *F1-score* ($F1$).

En primer lugar, la tasa de acierto de un modelo de clasificación o Acc , es la proporción de observaciones que se han clasificado de manera correcta. En este problema hace referencia a la proporción de ventanas cuya predicción coincide con la fase etiquetada:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + FN}{N}. \quad (4.6)$$

En segundo lugar, la sensibilidad, denominada en este trabajo Rec , es la probabilidad de clasificar correctamente una fase concreta de la marcha:

$$Rec = \frac{TP}{TP + FN}. \quad (4.7)$$

En tercer lugar, la métrica $Prec$ hace referencia al porcentaje de ventanas pertenecientes a una fase que fueron clasificadas correctamente. Se expresa matemáticamente de la

siguiente manera:

$$Prec = \frac{TP}{TP + FP}. \quad (4.8)$$

Por último, el *F1-score*, denominado en este trabajo *F1* indica el promedio entre *Prec* y *Rec*:

$$F1 = 2 \cdot \frac{Prec \cdot Rec}{Prec + Rec}. \quad (4.9)$$

En el caso de este TFG, el problema a resolver no se trata de clasificación binaria, sino multiclase, ya que se pretenden segmentar cuatro fases de la marcha. Para ello, la matriz de confusión y las métricas se calculan de manera similar. De esta manera, para el problema de clasificación de cuatro fases, la matriz de confusión de la fase HS quedará según se muestra en la Tabla 4.4.

Tabla 4.4: Matriz de confusión multiclase.

		Predicción				Total
		Fase HS	Fase FF	Fase PO	Fase SW	
Real	Fase HS	<i>TP</i>	<i>FN</i>	<i>FN</i>	<i>FN</i>	<i>TP + FN</i>
	Fase FF	<i>FP</i>	<i>TN</i>	<i>TN</i>	<i>TN</i>	<i>FP + TN</i>
	Fase PO	<i>FP</i>	<i>TN</i>	<i>TN</i>	<i>TN</i>	<i>FP + TN</i>
	Fase SW	<i>FP</i>	<i>TN</i>	<i>TN</i>	<i>TN</i>	<i>FP + TN</i>
	Total	<i>TP + FP</i>	<i>FN + TN</i>	<i>FN + TN</i>	<i>FN + TN</i>	<i>N</i>

Por lo tanto, para cada clase habrá un valor de *Rec*, un valor de *Prec* y un valor de *F1*. Para simplificar este análisis, se estudian las medias de estas métricas en lugar de las métricas individuales. Para calcular las medias de cada una de las métricas, hay dos opciones. Por un lado, “*weighted average*”, que tiene en cuenta el número de observaciones reales de cada clase; y por otro lado, “*macro average*” que otorga el mismo peso a todas las clases.

Siendo *k* el número de clases del problema a resolver y *N* el número total de muestras, en la Ecuación (4.10) se muestra la manera matemática de calcular el *Rec* “*weighted average*”, y en la Ecuación (4.11) como se calcula el *Rec* “*macro average*”.

$$Rec(\textit{weighted}) = \sum_{i=0}^k \frac{TP(i)}{TP(i) + FN(i)} \cdot \frac{TP(i) + FN(i)}{N} = \frac{\sum_{i=0}^k TP(i)}{N} = Acc \quad (4.10)$$

$$Rec(\textit{macro}) = \sum_{i=0}^k \frac{TP(i)}{TP(i) + FN(i)} \cdot \frac{N}{k} \quad (4.11)$$

Como se puede observar, al calcular el *Rec* “*weighted average*”, el resultado es el mismo que la *Acc*.

Para la *Prec*, tanto “*weighted average*” como “*macro average*”, se calculan de manera similar al caso del *Rec*. Las Ecuaciones (4.12) y (4.13) muestran los cálculos correspondientes:

$$Prec(weighted) = \sum_{i=0}^k \frac{TP(i)}{TP(i) + FP(i)} \cdot \frac{TP(i) + FN(i)}{N} \quad (4.12)$$

$$Prec(macro) = \sum_{i=0}^k \frac{TP(i)}{TP(i) + FP(i)} \cdot \frac{N}{k} \quad (4.13)$$

Igualmente, para *F1*, las expresiones matemáticas quedan reflejadas en las Ecuaciones (4.14) y (4.15):

$$F1(weighted) = \sum_{i=0}^k 2 \cdot \frac{Prec(i) \cdot Rec(i)}{Prec(i) + Rec(i)} \cdot \frac{TP(i) + FN(i)}{N} \quad (4.14)$$

$$F1(macro) = \sum_{i=0}^k 2 \cdot \frac{Prec(i) \cdot Rec(i)}{Prec(i) + Rec(i)} \cdot \frac{N}{k} \quad (4.15)$$

4.5.2. Métricas de Regresión

Debido a que para problemas de regresión el objetivo es predecir un valor numérico continuo, las métricas de evaluación miden la diferencia entre las predicciones y el valor real. En este TFG, la regresión es empleada para estimar el tiempo restante hasta cada uno de los eventos que componen la marcha. Las principales métricas empleadas para regresión son *Mean Square Error (error cuadrático medio)* (MSE), *Root Mean Square Error (raíz del error cuadrático medio)* (RMSE), *Mean Absolute Error (error absoluto medio)* (MAE) y coeficiente de determinación (R^2) [74, 75].

En primer lugar, el error cuadrático medio, o MSE, hace referencia al promedio del cuadrado de la resta entre el tiempo real y el tiempo estimado hasta cada una de las fases de la marcha. Una de las ventajas de esta métrica es que al estar elevada al cuadrado hace que los errores grandes penalicen mucho el modelo:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (4.16)$$

donde N es el número total de ventanas, y_i es el tiempo real que falta hasta el evento de la marcha estudiado, y \hat{y}_i el tiempo estimado hasta dicho evento. Esta métrica trabaja con unidades de tiempo al cuadrado

En ocasiones, también se utiliza como métrica la raíz del error cuadrático medio o

RMSE:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}. \quad (4.17)$$

En segundo lugar, el error absoluto medio, o MAE, se calcula como el promedio de las diferencias absolutas entre el tiempo real y el tiempo estimado hasta cada una de las fases de la marcha. Esto permite proporcionar una medida uniforme de que tan bien está funcionando el modelo:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|. \quad (4.18)$$

Por último, R^2 mide la bondad del ajuste realizado por la recta de regresión. Cuantifica la proporción de variación de las variables independientes (señales del acelerómetro triaxial y giróscopo triaxial) que es explicado por las variables dependientes (etiquetas de los tiempos que se quieren estimar). Esto permite dar una indicación de que tan bueno es el ajuste del modelo. Sus valores están siempre entre -1 y 1 , y se calcula mediante la Ecuación (4.19).

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (4.19)$$

4.6. Herramientas utilizadas

4.6.1. Python

Se ha utilizado el lenguaje de programación *Python*, con el editor *Visual Studio Code* para la manipulación de los datos y el desarrollo de los modelos de ML. *Python* es un lenguaje de programación muy popular en la comunidad de ML debido a la gran cantidad de librerías disponibles. Algunas de las librerías que se han empleado para el desarrollo de este TFG son las siguientes:

- *Numpy*: esta librería proporciona herramientas para la realizar operaciones matemáticas de alto rendimiento. Se ha utilizado para el procesamiento numérico de los datos.
- *Pandas*: esta librería proporciona herramientas para la lectura y escritura de datos en diversos formatos. Se ha utilizado para el trabajo con los diferentes *dataframes* que contienen los datos.
- *Scikit-learn*: esta librería proporciona una gran cantidad de algoritmos de aprendizaje automático, herramientas de procesamiento de datos y utilidades para la evaluación de modelos.

- *Plotly Express* y *Matplotlib*: estas librerías permiten la visualización de datos, incluyendo funciones que permiten crear gráficos complejos y elaborados con el objetivo de representar bien los datos y hacerlos comprensibles.

4.6.2. LaTeX

LaTeX es un sistema de composición de textos que permite la creación de documentos de alta calidad y es ampliamente utilizado en la comunidad científica. Esta memoria ha sido redactada con LaTeX mediante su editor de texto *online Overleaf*.

Capítulo 5

Resultados y Discusión

En este capítulo se exponen y discuten los resultados obtenidos en los experimentos realizados. En la Sección 5.1 se describen los resultados de la selección de hiperparámetros y ancho de ventana óptimo, HV. En la Sección 5.2 se presentan los resultados de la evaluación individual en cinco voluntarios independientes, VI. En la Sección 5.3 se exponen los resultados de la generación de datos sintéticos, DS. En la Sección 5.4 se discuten los resultados del desempeño en nuevos individuos, NI. Por último, en la Sección 5.5 se analizan los resultados de la predicción de eventos de cambio de fase, PE.

5.1. Selección de hiperparámetros y ancho de ventana óptimo

Los resultados obtenidos en el primer experimento, HV, en el que se evalúan diferentes conjuntos de datos para realizar la segmentación de las fases de la marcha, se muestran en las Tablas 5.1, 5.2 y 5.3. Se han separado los resultados de acuerdo a las diferentes entradas utilizadas en cada caso. En la Tabla 5.1 se muestran los resultados obtenidos a partir del conjunto de datos en crudo. En la Tabla 5.2 se presentan los resultados obtenidos utilizando la extracción de características estadísticas. Por último, en la Tabla 5.3 se recogen los resultados obtenidos con la extracción de características polinomiales. En cada una de estas tablas están recogidas las métricas descritas en la Sección 4.5.1. En concreto, se incluyen las métricas *Acc*, *F1 (weighted)*, *Rec (weighted)* y *Prec (weighted)*. Además, se recoge el número de ventanas de cada fase predichas de manera correcta, es decir, el número de *TP* de cada fase.

En primer lugar se muestran los resultados obtenidos al utilizar los datos en crudo. En la Tabla 5.1 se presentan las métricas obtenidas para cada uno de los modelos empleados.

Tabla 5.1: Resultados del Experimento HV con los datos en crudo.

Modelos	<i>Acc</i>	<i>F1</i>	<i>Rec</i>	<i>Prec</i>	T-HS	T-FF	T-PO	T-SW
Regresión logística	0.892	0.892	0.892	0.893	85	261	262	256
SVM	0.930	0.930	0.930	0.932	87	288	273	253
KNN	0.928	0.928	0.928	0.930	88	285	271	255
Árbol de decisión	0.911	0.911	0.911	0.912	84	283	262	253
RF	0.936	0.936	0.936	0.937	86	292	273	255
AdaBoost	0.719	0.716	0.719	0.755	94	150	263	189
XGBoost	0.935	0.935	0.935	0.937	86	291	274	255
MLP	0.919	0.919	0.919	0.920	89	279	266	256
LSTM	0.155	0.138	0.155	0.167	9	27	98	17
Total	-	-	-	-	104	307	279	278

En la Tabla 5.1 se puede observar que el modelo que proporciona los resultados menos favorables es la red LSTM, que obtiene métricas de *Acc*, *F1*, *Rec* y *Prec* por debajo de 0.2. Por otra parte, los modelos que presentan mayores métricas son RF y *XGBoost*, con métricas de *Acc*, *F1*, *Rec* y *Prec* de más de 0.935. Además, los modelos SVC y KNN reportan métricas que superan el valor de 0.92 en *Acc*, *F1*, *Rec* y *Prec*. Esto se debe a su capacidad de generalización, que permite encontrar las relaciones no lineales entre las medidas de los sensores inerciales y las fases de la marcha.

Tras la obtención de estos primeros resultados, se realiza un análisis más detallado para examinar si existen diferencias significativas en las métricas obtenidas para cada una de las fases. La Figura 5.1 muestra la *Prec*, *Rec* y *F1* con las que se clasifican las fases de la marcha, HS, FF, PO y SW, empleando los diferentes modelos evaluados.

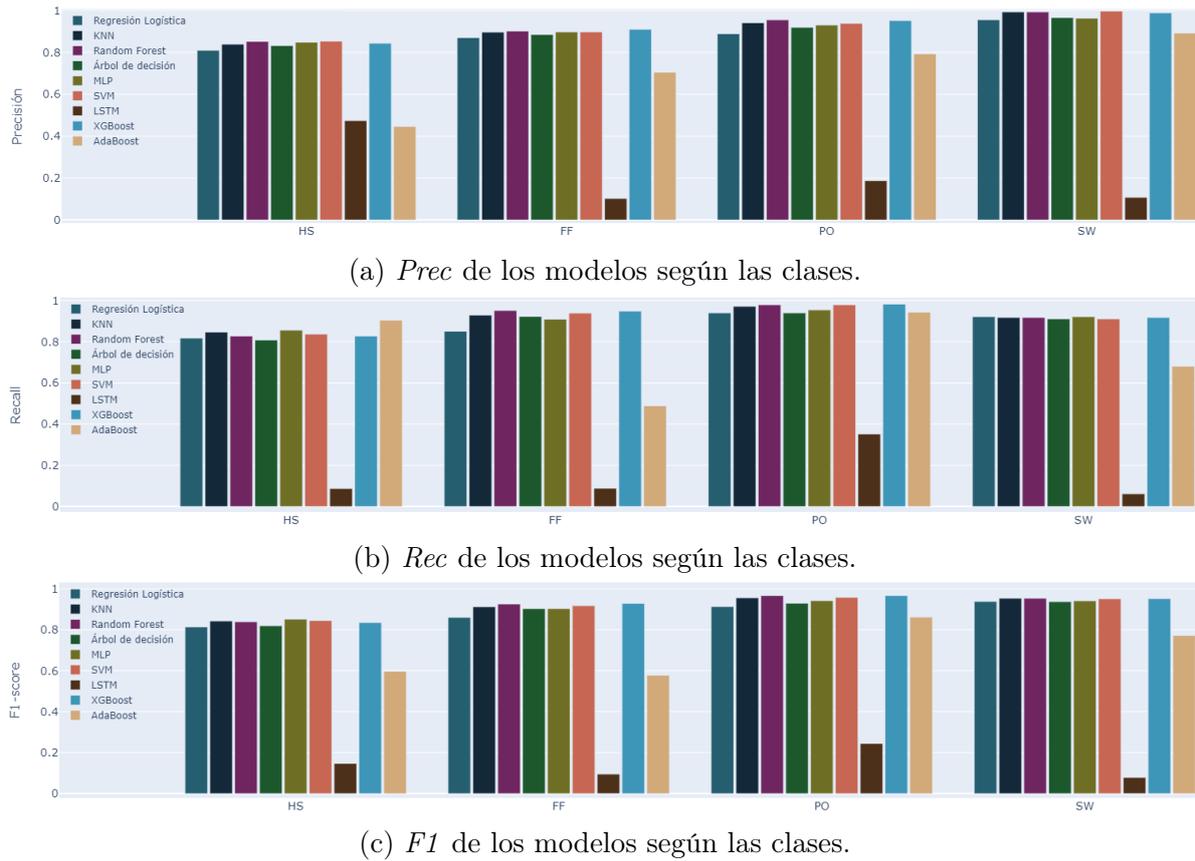


Figura 5.1: *Prec*, *Rec* y *F1* de los modelos evaluados para identificar cada una de las fases de la marcha.

Al analizar la Figura 5.1, se identifican diferencias en las métricas obtenidas para cada una de las clases. En términos generales, y sin tener en cuenta el modelo LSTM, la fase HS obtiene un rendimiento ligeramente inferior en comparación con las demás fases. Esto se debe a que esta fase tiene una duración más corta, por lo que presenta una representación menor en el conjunto de datos. Al estar subrepresentada con respecto al resto de clases, no permite a los modelos aprender sus patrones más característicos. Por otra parte, destaca la alta *Prec* de la fase SW, lo que indica una capacidad destacada para reconocer esta fase en particular. Además, la fase PO logra, tanto un *Rec*, como una *Prec*, elevados, superando ambos valores de 0.9. Esto indica que la detección de esta fase se logra de manera adecuada en la mayoría de los casos.

En relación con los algoritmos de ML empleados, destaca el rendimiento de los modelos RF y *XGBoost*. Estos algoritmos logran las métricas más altas, por encima de 0.9 para la clasificación de la mayoría de fases. Los siguientes modelos con métricas más elevadas son SVM y KNN, cuyos resultados son similares a los algoritmos previamente citados, variando alrededor de un 0.01. Por lo tanto, RF y *XGBoost* son los algoritmos más adecuados para segmentar las fases de la marcha a partir de las señales inerciales en crudo.

La Tabla 5.2 presenta los resultados obtenidos con extracción de características estadísticas. Como se puede apreciar, el modelo que menores métricas reporta es nuevamente la red LSTM. En este caso este modelo logra resultados ligeramente superiores a los obtenidos en el entrenamiento con los datos en crudo (ver Tabla 5.1), en concreto por encima de 0.25 en la métrica *Acc*. Las mayores métricas las obtiene el modelo *XGBoost*, con *Acc*, *F1*, *Rec* y *Prec* por encima de 0.93. Estos resultados no presentan diferencias significativas al compararlos con los resultados obtenidos en el caso del entrenamiento con datos en crudo, presentados en la Tabla 5.1. Los modelos RF y árbol de decisión también obtienen métricas altas, con *Acc* superiores a 0.91, siendo ligeramente mayores las métricas de RF. Por tanto, *XGBoost* y RF son los modelos más adecuados para clasificar las fases de la marcha cuando se emplean como entradas características estadísticas de las señales inerciales.

Tabla 5.2: Resultados del Experimento HV con extracción de características estadísticas.

Modelos	<i>Acc</i>	<i>F1</i>	<i>Rec</i>	<i>Prec</i>	T-HS	T-FF	T-PO	T-SW
Regresión logística	0.845	0.844	0.845	0.845	75	262	275	206
SVM	0.891	0.892	0.891	0.901	84	273	287	219
KNN	0.894	0.895	0.894	0.900	82	278	284	222
Árbol de decisión	0.910	0.910	0.910	0.910	79	297	280	225
RF	0.930	0.930	0.930	0.931	79	313	283	226
AdaBoost	0.671	0.663	0.671	0.667	11	174	253	212
XGBoost	0.932	0.932	0.932	0.933	78	315	282	228
MLP	0.857	0.857	0.857	0.861	77	265	273	215
LSTM	0.256	0.248	0.256	0.501	80	74	93	1
Total	-	-	-	-	89	329	299	251

Por último, la Tabla 5.3 muestra los resultados obtenidos con selección de características polinomiales. Acorde a esta tabla, igual que en el caso del entrenamiento con los datos en crudo (ver Tabla 5.1), el modelo que reporta métricas más favorables es RF. Este modelo, logra resultados por encima de 0.87 en las métricas de *Acc*, *F1*, *Rec* y *Prec*. Otro de los modelos que obtiene resultados por encima de 0.85 en la métrica *Acc* es *XGBoost*. En contraposición, el modelo que obtiene menor rendimiento es la regresión logística, reportando métricas de *Acc*, *F1*, *Rec* y *Prec* por debajo de 0.5. Además, destaca la mejora de las métricas en los resultados obtenidos por la red LSTM, que con este conjunto de datos logra un *Acc* de casi 0.8, frente al valor de 0.155 con los datos en crudo (ver Tabla 5.1) y el valor de 0.256 obtenido con la extracción de características estadísticas (ver Tabla 5.2). Por tanto, la red LSTM proporciona mayores métricas cuando las entradas son las características polinomiales, aunque en ningún caso supera las métricas de RF o *XGBoost*. De esta forma, nuevamente los modelos RF y *XGBoost* son los más adecuados para segmentar las fases de la marcha, utilizando en este caso las características polinomiales como entradas a los clasificadores.

Tabla 5.3: Resultados del Experimento HV con extracción de características polinomiales.

Modelos	<i>Acc</i>	<i>F1</i>	<i>Rec</i>	<i>Prec</i>	T-HS	T-FF	T-PO	T-SW
Regresión logística	0.473	0.473	0.473	0.588	53	175	147	83
SVM	0.594	0.580	0.594	0.625	19	203	178	175
KNN	0.743	0.743	0.743	0.747	72	221	232	195
Árbol de decisión	0.729	0.728	0.729	0.755	65	210	252	179
RF	0.875	0.874	0.875	0.876	83	269	257	238
Adaboost	0.559	0.549	0.559	0.571	22	159	195	166
XGBoost	0.854	0.854	0.854	0.856	80	266	253	228
MLP	0.716	0.710	0.716	0.734	48	204	262	180
LSTM	0.796	0.795	0.796	0.799	84	226	252	209
Total	-	-	-	-	104	307	279	278

De las Tablas 5.1, 5.2 y 5.3 destaca la métrica *Acc*, que en el 70% de los casos se encuentra por encima de 0.7, en el 60% de casos por encima de 0.8 y en el 30% de los casos por encima de 0.9. Esto indica que la mayor parte de los modelos evaluados con los diferentes planteamientos clasifican de forma correcta las muestras recogidas del análisis de la marcha y, por tanto, son adecuados para el problema a resolver.

Tanto con el conjunto de datos en crudo como con la extracción de características estadísticas se alcanzan resultados en las métricas de *Acc*, *F1*, *Rec* y *Prec* de más de 0.93. En concreto, destacan los resultados obtenidos al entrenar los modelos con el conjunto de datos en crudo, logrando métricas de *Acc* superiores a 0.935. Sin embargo, al emplear la extracción de características polinomiales, se observa una diferencia significativa, donde la mayor *Acc* alcanzada es de 0.875, por lo que esta no se considera la opción más adecuada. En base a estos resultados, el conjunto de datos seleccionado es el que contiene los datos en crudo de las señales del acelerómetro y el giróscopo.

En cuanto al método de clasificación más adecuado, en los tres conjuntos de datos las mayores métricas se obtienen con los métodos *ensemble*, específicamente con RF y *XGBoost*. Estos modelos obtienen un *Acc* superior a 0.93 con el conjunto de datos en crudo y con extracción de características estadísticas. Esto es debido a que combinar varios modelos de clasificación permite detectar patrones más complejos y por tanto mejorar la segmentación de las fases de la marcha.

Habiendo establecido los modelos que proporcionan las mayores métricas de clasificación y el conjunto de entradas más adecuado, se analiza el tamaño de ventana óptimo para la segmentación de las fases de la marcha. Para ello, se utilizan los datos en crudo y el modelo RF ya que esta es la configuración que obtiene mayores métricas.

En la Figura 5.2 se presentan las métricas *Acc*, *F1*, *Rec* y *Prec* para cada uno de los anchos de ventana estudiados. *F1*, *Rec* y *Prec* se han calculado como la media *weighted* obtenida para cada una de las fases de la marcha.

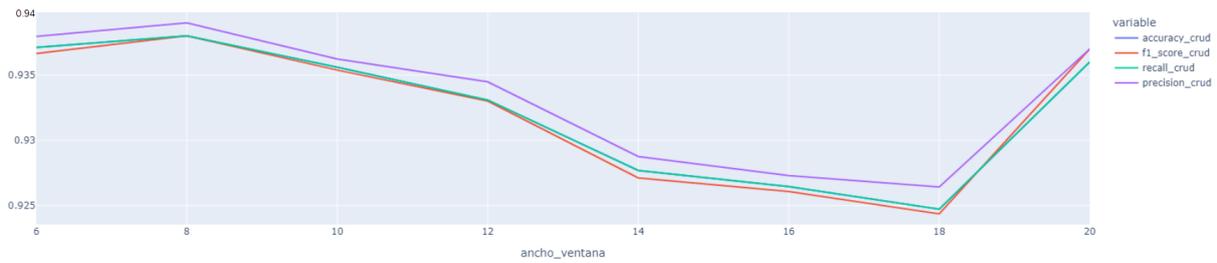


Figura 5.2: Métricas de clasificación en función del ancho de ventana.

Como se puede apreciar en la Figura 5.2, los resultados se acercan a 0.94 para las métricas de *Acc*, *F1*, *Rec* y *Prec* al reducir la ventana a 8 muestras. Este efecto se debe a que al reducir el ancho de ventana, se reduce la cantidad de información que contiene cada segmento de la señal. Esto permite detectar mejor los cambios rápidos de la señal, mejorando así la clasificación. Además, en la Figura 5.2 se observa que, de manera general, a medida que aumenta el ancho de ventana las métricas empeoran. Esto es debido a que cuando el ancho de ventana es demasiado grande, hay información característica de más de una clase, lo que dificulta la determinación de la fase correspondiente. En base a estos resultados, en los experimentos cuyos resultados se detallan a continuación, se utilizan ventanas de 8 muestras. El desplazamiento entre ventanas es de un 50% de su tamaño, es decir, 4 muestras.

5.2. Evaluación individual en 5 voluntarios independientes

En este experimento, VI, se evalúa la segmentación de las fases de la marcha en cinco voluntarios independientes. En base a los resultados del experimento HV, se evalúan los clasificadores empleando como entradas, únicamente, las señales en crudo segmentadas en ventanas de 8 muestras con un desplazamiento de 4 muestras. Para ello se toman los datos en crudo de los cinco voluntarios detallados en la Sección 4.1.

En primer lugar, se evalúan los resultados obtenidos para cada uno de los voluntarios. Estos resultados se muestran en las Tablas 5.4, 5.5, 5.6, 5.7 y 5.8, que se corresponden con los voluntarios 1, 3, 4, 5 y 8, respectivamente.

Tabla 5.4: Resultados del Experimento VI con los datos del voluntario 1.

Modelos	<i>Acc</i>	<i>F1</i>	<i>Rec</i>	<i>Prec</i>	T-HS	T-FF	T-PO	T-SW
Regresión Logística	0.811	0.810	0.811	0.837	167	734	501	562
SVC	0.886	0.887	0.886	0.898	186	758	620	583
KNN	0.928	0.928	0.928	0.929	192	744	730	582
Árbol de decisión	0.911	0.910	0.911	0.912	181	726	715	584
RF	0.938	0.937	0.938	0.939	192	755	733	590
AdaBoost	0.655	0.649	0.655	0.688	96	455	727	310
XGBoost	0.936	0.936	0.936	0.937	193	752	732	591
MLP	0.884	0.883	0.884	0.884	152	703	699	588
LSTM	0.560	0.535	0.560	0.538	4	575	435	342
Total	-	-	-	-	225	776	762	658

Tabla 5.5: Resultados del Experimento VI con los datos del voluntario 3.

Modelos	<i>Acc</i>	<i>F1</i>	<i>Rec</i>	<i>Prec</i>	T-HS	T-FF	T-PO	T-SW
Regresión Logística	0.826	0.827	0.826	0.830	193	318	577	663
SVC	0.883	0.883	0.883	0.886	210	365	616	680
KNN	0.903	0.903	0.903	0.906	213	405	622	673
Árbol de decisión	0.890	0.891	0.890	0.894	211	411	589	675
RF	0.917	0.917	0.917	0.919	209	427	632	676
AdaBoost	0.704	0.714	0.704	0.767	209	349	542	392
XGBoost	0.922	0.922	0.922	0.923	212	430	632	679
MLP	0.890	0.890	0.890	0.893	200	387	621	679
LSTM	0.487	0.462	0.487	0.454	14	119	341	559
Total	-	-	-	-	244	465	649	760

Tabla 5.6: Resultados del Experimento VI con los datos del voluntario 4.

Modelos	<i>Acc</i>	<i>F1</i>	<i>Rec</i>	<i>Prec</i>	T-HS	T-FF	T-PO	T-SW
Regresión Logística	0.701	0.708	0.701	0.737	329	352	386	485
SVC	0.862	0.862	0.862	0.870	353	594	434	529
KNN	0.885	0.885	0.885	0.888	354	584	492	531
Árbol de decisión	0.853	0.852	0.853	0.854	327	560	481	521
RF	0.899	0.899	0.899	0.901	350	590	507	545
AdaBoost	0.784	0.784	0.784	0.788	299	462	472	504
XGBoost	0.900	0.899	0.900	0.900	343	593	507	550
MLP	0.837	0.836	0.837	0.837	312	523	498	521
LSTM	0.574	0.535	0.574	0.574	46	268	450	508
Total	-	-	-	-	406	632	543	633

Tabla 5.7: Resultados del Experimento VI con los datos del voluntario 5.

Modelos	<i>Acc</i>	<i>F1</i>	<i>Rec</i>	<i>Prec</i>	T-HS	T-FF	T-PO	T-SW
Regresión Logística	0.500	0.484	0.500	0.512	155	131	218	531
SVC	0.834	0.833	0.834	0.841	261	313	546	607
KNN	0.868	0.868	0.868	0.872	274	371	547	605
Árbol de decisión	0.842	0.843	0.842	0.847	256	375	514	598
RF	0.879	0.879	0.879	0.884	282	378	547	613
AdaBoost	0.763	0.766	0.763	0.778	240	329	505	506
XGBoost	0.876	0.877	0.876	0.881	282	364	552	616
MLP	0.803	0.803	0.803	0.805	254	329	509	572
LSTM	0.445	0.386	0.445	0.425	103	5	530	285
Total	-	-	-	-	324	444	585	717

Tabla 5.8: Resultados del Experimento VI con los datos del voluntario 8.

Modelos	<i>Acc</i>	<i>F1</i>	<i>Rec</i>	<i>Prec</i>	T-HS	T-FF	T-PO	T-SW
Regresión Logística	0.821	0.823	0.821	0.830	243	644	525	593
SVC	0.908	0.908	0.908	0.911	256	707	608	646
KNN	0.915	0.915	0.915	0.917	263	696	633	642
Árbol de decisión	0.898	0.898	0.898	0.901	259	694	618	621
RF	0.922	0.923	0.922	0.925	267	711	632	642
AdaBoost	0.637	0.638	0.637	0.738	285	577	348	345
XGBoost	0.925	0.925	0.925	0.927	270	703	639	645
MLP	0.884	0.883	0.884	0.884	241	661	622	633
LSTM	0.458	0.392	0.458	0.521	208	639	236	35
Total	-	-	-	-	306	746	670	718

Los resultados varían según cada voluntario, pero las diferencias no son muy significativas, variando alrededor de 0.05. Los voluntarios 1, 3 y 8 muestran métricas más elevadas con tasas de acierto de entorno al 92%-93% (ver Tablas 5.4, 5.5 y 5.8). Por otro lado, el voluntario 4 presenta métricas de *Acc*, *F1*, *Rec* y *Prec* de entorno a 0.9 (ver Tabla 5.6), mientras que el voluntario 5 obtiene métricas cercanas a 0.88 (ver Tabla 5.7). Estas variaciones se deben a que cada persona puede presentar características únicas en su marcha, lo que influye en las métricas obtenidas por los modelos.

Con el objetivo de determinar los modelos más adecuados acorde a sus métricas de clasificación, se analizan las métricas medias para el conjunto de cinco voluntarios. En la Tabla 5.9 se muestra la media del *Acc*, *F1*, *Rec* y *Prec* obtenidos con los diferentes clasificadores para los cinco sujetos analizados

Tabla 5.9: Media de los resultados obtenidos en el Experimento VI.

Modelos	<i>Acc</i>	<i>F1</i>	<i>Rec</i>	<i>Prec</i>
Regresión Logística	0.732	0.730	0.732	0.749
SVC	0.875	0.875	0.875	0.881
KNN	0.900	0.900	0.900	0.902
Árbol de decisión	0.879	0.879	0.879	0.881
RF	0.911	0.911	0.911	0.913
AdaBoost	0.709	0.710	0.709	0.752
XGBoost	0.912	0.912	0.912	0.914
MLP	0.860	0.859	0.860	0.860
LSTM	0.505	0.462	0.505	0.507

El modelo que logra mayores resultados en promedio es el modelo *XGBoost*, con un 91.2% de predicciones correctas, seguido de RF, con un 91.1% y KNN con un 90%. Esto indica que estos modelos son aplicables a otros voluntarios, y se siguen obteniendo resultados aceptables. Por otra parte, los resultados menos favorables son los obtenidos por la red LSTM, la cual presenta una tasa de acierto de tan solo el 50.5%.

5.3. Generación de datos sintéticos

El efecto de la estandarización de los datos en crudo y el balanceo de las clases empleando la técnica SMOTE, se analiza sobre los datos de un único voluntario en el experimento DS. En este caso, se emplean los datos del voluntario 1 y los resultados obtenidos se muestran en la Tabla 5.10.

Tabla 5.10: Resultados del Experimento DS.

Modelos	<i>Acc</i>	<i>F1</i>	<i>Rec</i>	<i>Prec</i>	T-HS	T-FF	T-PO	T-SW
Regresión logística	0.637	0.637	0.637	0.639	688	378	489	430
SVC	0.934	0.934	0.934	0.935	716	730	766	696
KNN	0.935	0.935	0.935	0.935	754	684	749	724
Árbol de decisión	0.861	0.861	0.861	0.861	702	637	694	647
RF	0.942	0.942	0.942	0.943	734	735	766	698
AdaBoost	0.704	0.703	0.704	0.710	669	534	580	409
XGBoost	0.941	0.941	0.941	0.942	739	724	764	704
MLP	0.926	0.926	0.926	0.926	731	697	749	705
LSTM	0.911	0.912	0.911	0.914	724	708	719	687
Total	-	-	-	-	767	770	795	780

En la Tabla 5.10 se puede observar que el modelo que reporta mayores métricas es RF, que alcanza *Acc*, *F1*, *Rec* y *Prec* superiores a 0.94. Los modelos *XGBoost*, SVC y KNN también alcanzan métricas superiores a 0.93. Por otro lado, el modelo que proporciona resultados menos favorables es regresión logística, que obtiene una tasa de acierto del 63.7%.

Comparándolos con los resultados obtenidos en el primer experimento (ver Tabla 5.1), se observa una pequeña mejoría en los modelos KNN, RF, *XGBoost* y MLP. Si se centra la comparación en la métrica de *Acc*, el modelo KNN pasa de un 0.928 a un 0.935; el modelo RF mejora del 0.936 a 0.942, el modelo *XGBoost* aumenta de 0.935 a 0.941 y el modelo MLP de 0.919 a 0.926.

Sin embargo, lo más destacado es la mejoría en los resultados del modelo LSTM, que pasa de 0.155 de *Acc* al 0.911. Esta mejora se debe a varios factores. En primer lugar, la creación de datos sintéticos permite que haya un equilibrio entre los datos de las diferentes fases, lo que contribuye a un mejor rendimiento del modelo. Además, la estandarización de los datos y la reducción del ancho de ventana a 8 muestras han tenido un impacto positivo en los resultados obtenidos por el modelo. Al tomar segmentos más pequeños de la señal como datos de entrada, se facilita la identificación de patrones y características relevantes, lo que potencia la capacidad de la red LSTM para realizar una segmentación más precisa de las fases de la marcha y por tanto, obtener una mayor *Acc*.

En conclusión, estos resultados confirman que los modelos RF y *XGBoost* continúan siendo los que mayores métricas reportan en términos de rendimiento. Además, se observa una mejoría en sus métricas en comparación con los experimentos anteriores.

5.4. Desempeño en nuevos individuos

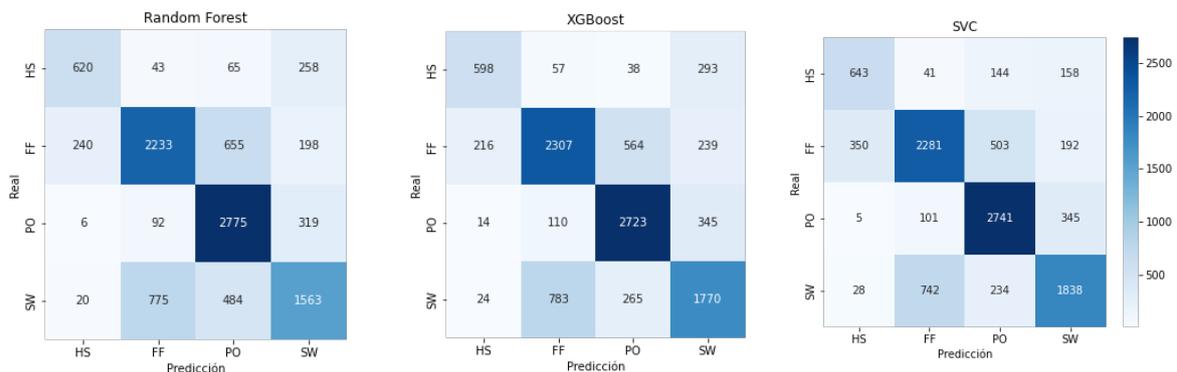
El experimento NI analiza la segmentación de las fases de la marcha en un voluntario independiente a con los que se entrenan los clasificadores. Para ello, se realiza una validación cruzada iterativa en la que se entrenan los modelos con cuatro voluntarios, y se evalúan con los datos del voluntario restante. Tras realizar las cinco iteraciones y evaluar los resultados en los cinco voluntarios independientes, se analizan las medias de las métricas de clasificación obtenidas en cada evaluación. Las medias de los resultados obtenidos en las iteraciones para cada uno de los métodos evaluados se muestran en la Tabla 5.11.

Tabla 5.11: Resultados del Experimento NI.

Modelos	<i>Acc</i>	<i>F1</i>	<i>Rec</i>	<i>Prec</i>
Regresión logística	0.528	0.524	0.528	0.532
KNN	0.670	0.666	0.670	0.678
Árbol de decisión	0.579	0.571	0.579	0.585
RF	0.693	0.684	0.693	0.717
XGBoost	0.700	0.694	0.700	0.713
AdaBoost	0.585	0.581	0.585	0.589
MLP	0.640	0.635	0.640	0.656
LSTM	0.657	0.652	0.657	0.673
SVC	0.713	0.707	0.713	0.734

En la Tabla 5.11 se observa que el modelo SVC reporta las mayores métricas, con un Acc media de 0.713. Le sigue el modelo $XGBoost$, que alcanza un 0.7 en la misma métrica, y RF con un 0.693. Por su parte, los resultados menos favorables se obtienen con los modelos regresión logística, árbol de decisión y $AdaBoost$, todos con métricas de Acc , $F1$, Rec y $Prec$ por debajo de 0.6.

Para mostrar los resultados obtenidos con los modelos más destacados de forma más detallada, en la Figura 5.3, se muestra la matriz de confusión de los modelos de RF, $XGBoost$ y SVC de la primera iteración. En esta iteración se entrenan los modelos con los datos de los voluntarios 3, 4, 5 y 8 y se evalúan con los datos del voluntario 1.



(a) Matriz de confusión RF. (b) Matriz de confusión XGB. (c) Matriz de confusión SVC.

Figura 5.3: Matrices de confusión obtenidas con RF, $XGBoost$ y SVC en el Experimento NI.

Analizando la Figura 5.3, destaca la dificultad de los modelos en la tarea de predicción de la fase FF, a pesar de ser la fase con menor movimiento. *A priori* se esperaría que la menor actividad recogida por los sensores inerciales condujera a resultados más favorables en la segmentación de esta fase en comparación con el resto. La dificultad en la predicción de esta fase se atribuye al ruido presente en los sensores inerciales, el cual afecta a la precisión de las mediciones y, en consecuencia, dificulta su correcta clasificación. Por otro lado, también se evidencia la dificultad de los algoritmos para la segmentación la fase HS. En concreto se observa que un alto número de ventanas pertenecientes a esta fase son clasificadas erróneamente como fase SW. Esta confusión se debe a la secuencialidad de ambas fases en el ciclo de la marcha, ya que se encuentran consecutivas en orden. Esta proximidad en el orden de las fases contribuye a la confusión entre las fases HS y SW durante el proceso de segmentación.

Como se puede observar, los resultados presentan un menor rendimiento que en experimentos anteriores. Al comparar este experimento con el experimento VI, cuyos resultados se muestran en la Tabla 5.9, destaca que el modelo con mayores métricas en la segmentación de la marcha no es el mismo. En el experimento VI, las mayores métricas se alcanzaban con el modelo $XGBoost$, que lograba un 0.912 de Acc de media. En ese

experimento el modelo SVC no presentaba las métricas más altas. Sin embargo, mostraba un rendimiento elevado con Acc superiores a 0.87, y por tanto cercano a los modelos que mayor rendimiento reportaban. Sin embargo, para este experimento las mayores métricas son las logradas por el modelo SVC, que alcanza hasta un 0.713 de Acc . Esto se debe a que los datos recogidos por el sensor inercial pueden variar ligeramente según la morfología y la forma de caminar de cada persona. Al entrenar los modelos con un conjunto de datos distinto al empleado para la validación, estas diferencias se hacen más notables. Con este experimento se determina que el modelo SVC tiene una mayor capacidad de generalización para segmentar patrones de marcha diferentes a los utilizados en el entrenamiento.

En resumen, los resultados obtenidos se consideran aceptables, ya que en más de la mitad de los modelos se supera el 65 % de ventanas predichas correctamente. Esto demuestra la capacidad de los modelos para generalizar entre diferentes individuos, considerando las variaciones individuales ya comentadas.

5.5. Predicción de eventos de cambio de fase

Este experimento, PE, tiene como objetivo la predicción del tiempo restante hasta cada una de las cuatro fases de la marcha, HS, FF, PO y SW. A partir de las estimaciones de los tiempos hasta los cambios de fase, se realiza una clasificación de las fases de la marcha siguiendo el sistema de reglas descrito en la Sección 4.4.5.

5.5.1. Predicción del tiempo restante hasta HS

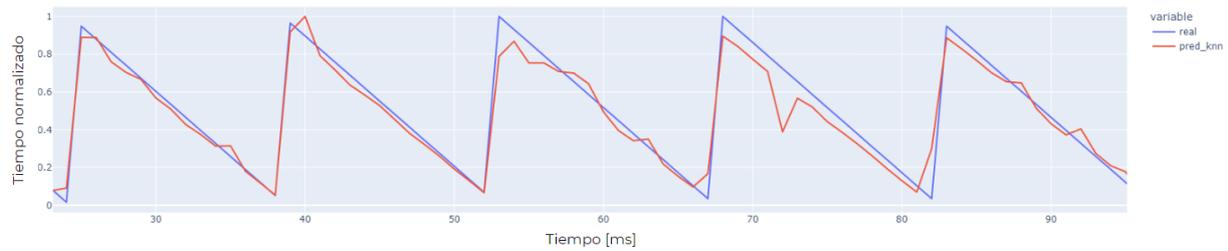
En primer lugar, se predice el tiempo restante hasta la fase de contacto inicial, HS. En la Tabla 5.12 se recogen las métricas obtenidas del entrenamiento de los diferentes modelos de regresión. Los datos de las métricas de los tiempos se reportan en milisegundos (ms). Además del MSE, RMSE, MAE y R^2 , descritos en la Sección 4.5.2, se ha incluido una columna con el RMSE relativo, siendo 1983 ms el mayor valor del conjunto de test.

Tabla 5.12: Resultados del Experimento R para predecir el tiempo restante hasta HS.

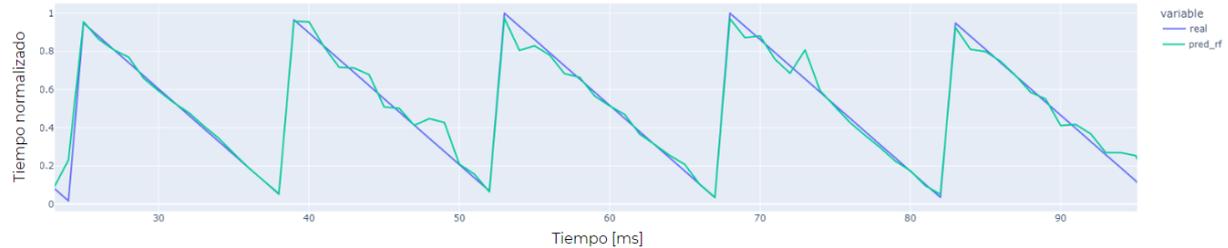
Modelos	MSE [ms]	RMSE [ms]	MAE [ms]	R^2	RMSE/ $y_{\text{test}}(\text{max})$ [%]
Regresión lineal	80322.16	283.41	233.73	0.178	14.29
KNN	32074.94	179.09	72.66	0.672	9.03
Árbol de decisión	54959.79	234.43	137.37	0.438	14.29
RF	34112.79	184.69	86.74	0.651	9.31
XGBoost	38978.83	197.43	105.26	0.601	9.95
AdaBoost	127566.82	357.16	309.25	-0.304	18.01
MLP	39597.88	198.99	118.13	0.595	10.03
LSTM	43563.15	208.71	123.73	0.554	10.52
SVR	30041.46	173.32	70.64	0.692	8.74

El modelo que muestra un desempeño superior es el modelo SVR, que obtiene un RMSE de 173.32 ms y un MAE de 70.64 ms, lo que indica que las predicciones al modelo son cercanas a los valores reales. Además, el coeficiente R^2 es 0.692, lo cual sugiere que este modelo muestra una tendencia a cierta correlación entre las entradas y las salidas. También cabe destacar que el coeficiente entre RMSE y el mayor valor del conjunto de test es de 8.74 %, lo que indica que el porcentaje de error cometido es considerablemente reducido.

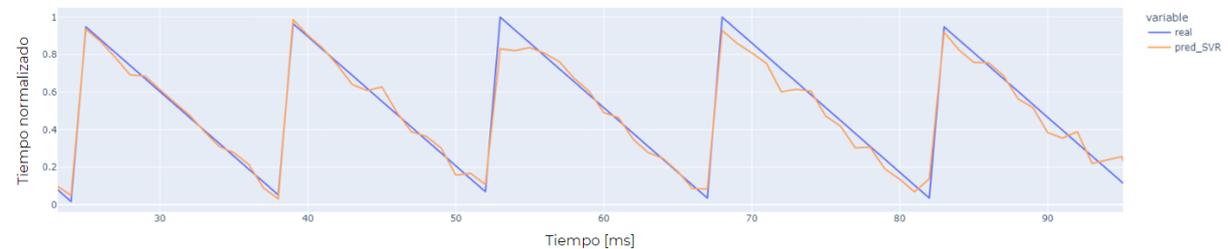
Por otro lado, los modelos KNN y RF también demuestran un destacado rendimiento. Al observar el coeficiente de determinación, se obtiene un valor de 0.672 para KNN y 0.651 para RF. Estos resultados no varían mucho de los obtenidos con el modelo SVR, lo que indica que tanto el modelo SVR, como los modelos KNN y RF pueden ser empleados para resolver este problema. La Figura 5.4 muestra un ejemplo de las predicciones de estos tres modelos.



(a) Resultado de la predicción del tiempo restante hasta HS con KNN.



(b) Resultado de la predicción del tiempo restante hasta HS con RF.



(c) Resultado de la predicción del tiempo restante hasta HS con SVR.

Figura 5.4: Resultados de la predicción del tiempo restante hasta HS.

Observando la Figura 5.4, donde se muestra en azul el valor real y en rojo, verde y naranja los valores predichos por los modelos KNN, RF y SVR respectivamente; destaca el ajuste de las predicciones a la tendencia de las curvas. En especial, destaca el rendimiento del modelo SVR, cuyas predicciones son muy similares a los valores reales.

En cuanto a los resultados menos favorables, en la Tabla 5.12 se pone de manifiesto que estos se obtienen con el modelo *AdaBoost*, que presenta un coeficiente de determinación de -0.304. Por lo tanto, el modelo no se ajusta adecuadamente a los datos de entrada correspondientes a las señales del acelerómetro y el giróscopo. Esto se debe a que este modelo se basa en la combinación de modelos de regresión débiles para construir un modelo final. Si estos modelos no se ajustan bien a los datos, puede afectar al rendimiento general del algoritmo.

Por su parte, la regresión lineal tampoco logra resultados óptimos, con un coeficiente de determinación de 0.178. Este limitado rendimiento se debe a que la regresión lineal asume una relación lineal simple entre las variables y esto no es suficiente para describir la complejidad de los datos recogidos por el sensor inercial.

En resumen, para la predicción del tiempo restante hasta la fase HS, los modelos KNN, RF y SVR logran resultados superiores. Por el contrario, los modelos *AdaBoost*

y regresión lineal presentan un desempeño inferior para la predicción de tiempos de cambio de fase. En estos últimos destaca también el porcentaje de error cometido, que en ambos casos es de un 14.29%.

5.5.2. Predicción del tiempo restante hasta FF

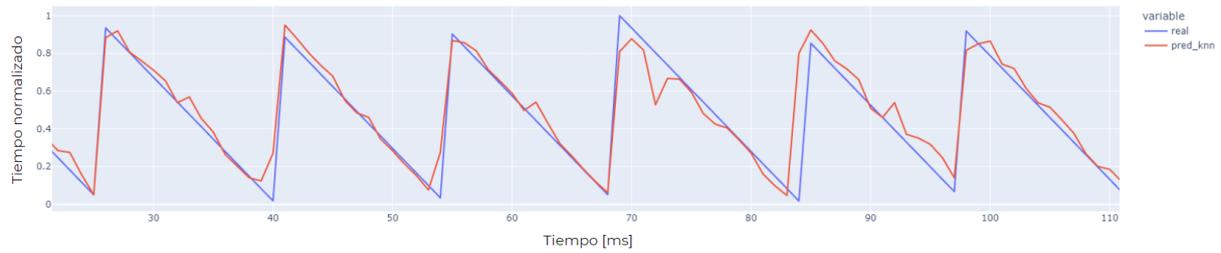
En segundo lugar, se predice el tiempo restante hasta la fase FF. Los resultados obtenidos se muestran en la Tabla 5.13. En este caso, el valor máximo del conjunto de test es 2083 ms.

Tabla 5.13: Resultados del Experimento R para predecir el tiempo restante hasta FF.

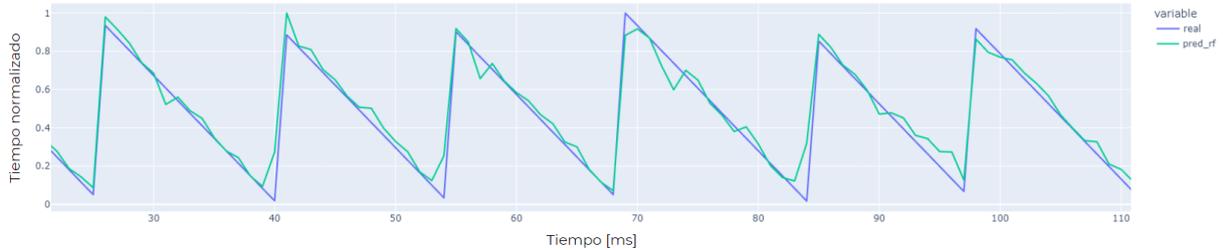
Modelos	MSE [ms]	RMSE [ms]	MAE [ms]	R^2	RMSE/ $y_{\text{test}}(\text{max})$ [%]
Regresión lineal	88966.14	298.27	232.31	0.179	14.31
KNN	43023.09	207.42	94.33	0.603	9.95
Árbol de decisión	64669.43	254.30	150.44	0.403	14.31
RF	44034.90	209.84	105.96	0.594	10.07
XGBoost	50262.96	224.19	122.89	0.536	10.76
AdaBoost	91000.13	301.66	238.53	0.161	14.48
MLP	50888.03	225.58	134.01	0.530	10.82
LSTM	53073.47	230.37	137.81	0.511	11.05
SVR	39948.81	199.87	82.03	0.631	9.59

Las conclusiones que se pueden extraer de esta tarea de regresión son similares a las obtenidas en el caso de la predicción del tiempo restante hasta HS (ver Tabla 5.12). De nuevo, el resultado más favorable se logra con el modelo SVR. Este modelo obtiene un RMSE de 199.87 ms, un MAE de 82.03 ms y un coeficiente de determinación de 0.631, lo cual indica que se encuentra una tendencia a cierta correlación entre las variables de entrada y salida, aunque tampoco puede considerarse significativa. Además el porcentaje de error con respecto al máximo valor de la variable a predecir es de 9.59%, lo que se considera un resultado satisfactorio.

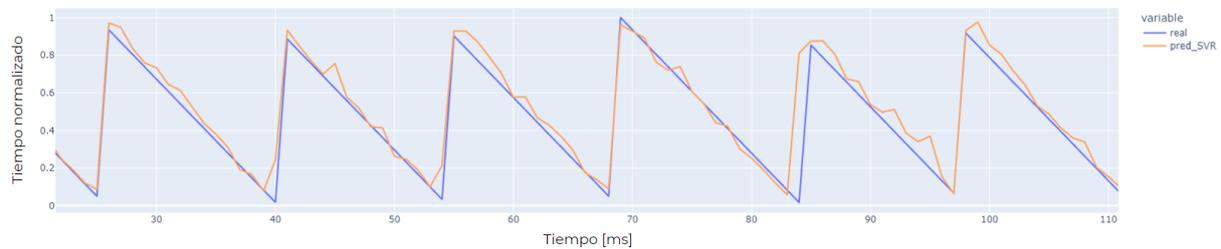
Otros modelos que también obtienen resultados destacables son KNN y RF. Estos modelos logran valores de RMSE muy similares, de 207.42 ms y 209.84 ms respectivamente. En cuanto a la métrica MAE, el modelo KNN obtiene un valor de 94.33 ms y RF de 105.96 ms. Para la métrica R^2 , los resultados son muy similares para KNN y RF, obteniendo 0.603 y 0.594 respectivamente. En la Figura 5.5, se muestra un ejemplo de los resultados obtenidos con los modelos KNN, RF y SVR.



(a) Resultado de la predicción del tiempo restante hasta FF con KNN.



(b) Resultado de la predicción del tiempo restante hasta FF con RF.



(c) Resultado de la predicción del tiempo restante hasta FF con SVR.

Figura 5.5: Resultados de la predicción del tiempo restante hasta FF.

Al igual que para la predicción del tiempo restante hasta la fase HS, el modelo que presenta un ajuste más preciso a los valores reales es SVR. No obstante, cabe destacar que en general los errores cometidos por la curva de regresión son mayores en comparación con los obtenidos en la Sección 5.5.1. Esto indica que el ajuste entre los valores reales y los predichos es menor en esta tarea.

De forma similar, los modelos que logran predicciones menos precisas son *AdaBoost* y regresión lineal. Estos obtienen un RMSE de 301.66 ms y 298.27 ms respectivamente, y un MAE de 238.53 ms y 232.31 ms respectivamente. Además, sus valores de R^2 son 0.161 y 0.179 respectivamente, lo cual indica que no son capaces de capturar adecuadamente las relaciones entre las variables de entrada y salida. También destacan sus porcentajes de error respecto al máximo valor del conjunto de test, que son de un 14.48 % para *AdaBoost* y 14.31 % para regresión lineal.

En consecuencia, las conclusiones extraídas de la predicción del tiempo restante hasta FF son similares a las de la Sección 5.5.1. De nuevo, los modelos más adecuados para la regresión de los tiempos de cambios de fase son KNN, RF y SVR y los peores *AdaBoost* y regresión lineal.

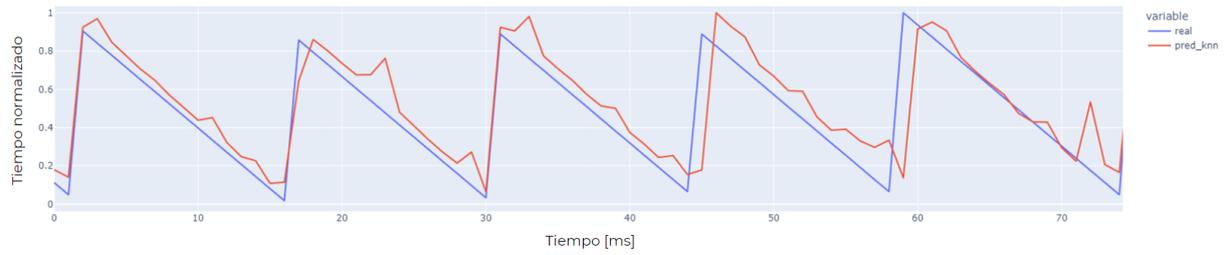
5.5.3. Predicción del tiempo restante hasta PO

A continuación, se predice el tiempo restante hasta la fase PO. Para ello, el procedimiento es el mismo que en el caso de la predicción del tiempo restante hasta HS o FF. En la Tabla 5.14 se recogen los resultados de esta predicción. En este caso, el mayor valor del conjunto de test es 1933 ms.

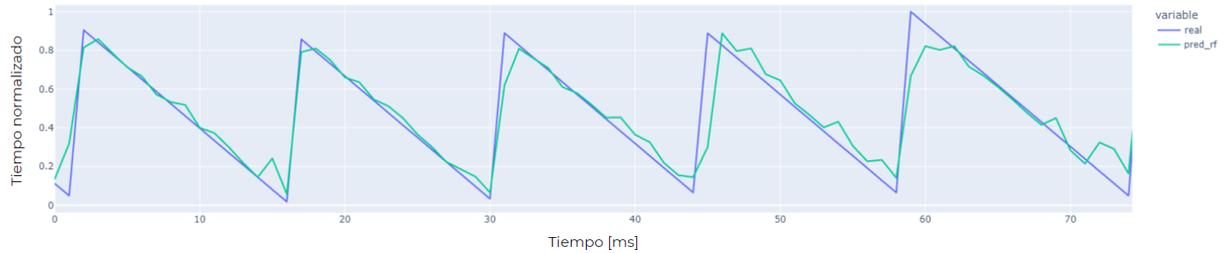
Tabla 5.14: Resultados del Experimento R para predecir el tiempo restante hasta PO.

Modelos	MSE [ms]	RMSE [ms]	MAE [ms]	R^2	RMSE/ $y_{\text{test}}(\text{max})$ [%]
Regresion lineal	85217.85	291.92	246.01	0.142	15.11
KNN	34517.13	185.78	83.98	0.652	9.61
Árbol de regresión	55614.37	235.82	134.37	0.441	12.19
RF	37816.84	194.46	101.09	0.619	10.06
XGBoost	42448.60	206.03	118.64	0.572	10.65
AdaBoost	76174.84	275.99	215.54	0.233	14.27
MLP	47466.31	217.86	137.52	0.522	11.27
LSTM	39085.98	197.70	104.36	0.606	10.22
SVR	33736.06	183.67	83.73	0.660	9.50

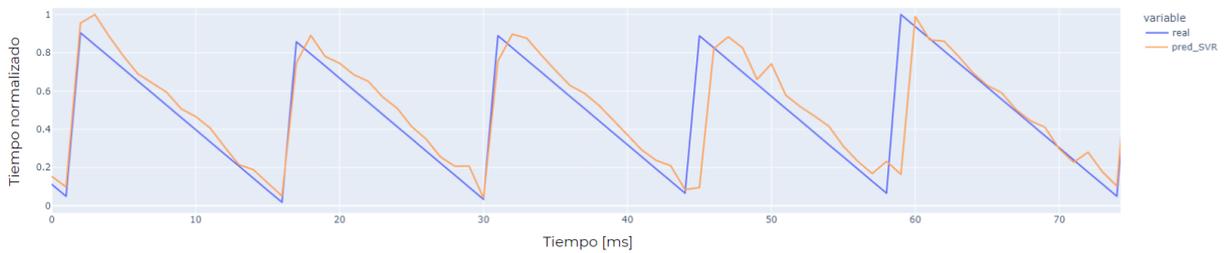
Acorde a las métricas obtenidas, los modelos que presentan un rendimiento superior para la predicción del tiempo restante hasta PO son los mismos que en las Secciones 5.5.1 y 5.5.2: KNN, RF y SVR. En concreto, el modelo que presenta los resultados más destacados es nuevamente SVR con métricas de RMSE, MAE y R^2 de 183.67 ms, 83.73 ms y 0.66, respectivamente. También obtiene un porcentaje de error de 9.5 %, el cual es bastante similar al obtenido en la predicción del tiempo restante hasta FF (ver Tabla 5.13). Los modelos KNN y RF una vez más logran resultados óptimos, obteniendo valores de RMSE de 185.78 ms y 94.46 ms, respectivamente; MAE de 83.98 ms y 101.09 ms, respectivamente y R^2 de 0.652 y 0.619, respectivamente. Un ejemplo de los resultados obtenidos con los modelos KNN, RF y SVR se muestra en la Figura 5.6.



(a) Resultado de la predicción del tiempo restante hasta PO con KNN.



(b) Resultado de la predicción del tiempo restante hasta PO con RF.



(c) Resultado de la predicción del tiempo restante hasta PO con SVR.

Figura 5.6: Resultados de la predicción del tiempo restante hasta PO.

En la Figura 5.6 se observa la discrepancia existente entre los valores reales y los valores predichos. En concreto, destaca que en la mayoría de muestras, el tiempo estimado es mayor al tiempo real. Además, se evidencia una mayor dificultad para la predicción del tiempo restante hasta PO que para la predicción del tiempo restante hasta HS o FF. Un ejemplo de esto es el modelo de KNN, que a pesar de ser uno de los modelos que menor error reporta, en la Figura 5.6a, destaca el desajuste entre los valores reales y predichos.

Por otra parte, en la Tabla 5.14 se evidencia que los modelos que muestran un desempeño inferior son los modelos *AdaBoost* y regresión lineal. Estos obtienen unos resultados de RMSE de 275.99 ms y 291.92 ms respectivamente; de MAE de 215.54 ms y 246.01 ms respectivamente, y coeficientes de determinación de 0.233 y 0.142. Además, destaca el porcentaje de error respecto al máximo valor del conjunto de test que es de 14.27 % para el modelo *AdaBoost* y 15.11 % para la regresión lineal, siendo este último el mayor obtenido hasta el momento.

5.5.4. Predicción del tiempo restante hasta SW

Por último, se predice el tiempo restante hasta la fase SW. La Tabla 5.15 resume los resultados obtenidos para este problema. En este problema de regresión el mayor valor del conjunto test era 1400 ms.

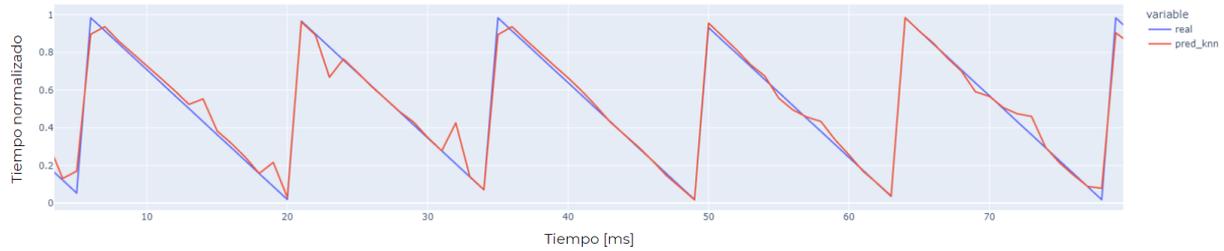
Tabla 5.15: Resultados del Experimento R para predecir el tiempo restante hasta SW.

Modelos	MSE [ms]	RMSE [ms]	MAE [ms]	R ²	RMSE/ $y_{\text{test}}(\text{max})$ [%]
Regresión lineal	75380.75	274.55	242.36	0.108	19.61
KNN	19006.18	137.86	49.89	0.775	9.84
Árbol de decisión	33031.68	181.74	90.28	0.609	19.61
RF	20304.43	142.49	69.67	0.759	10.17
XGBoost	23443.86	153.11	83.04	0.722	10.93
AdaBoost	65665.58	256.25	226.91	0.223	18.30
MLP	25976.65	161.17	98.55	0.692	11.51
LSTM	21568.83	146.86	78.43	0.745	10.49
SVR	18495.11	135.99	57.56	0.78	9.71

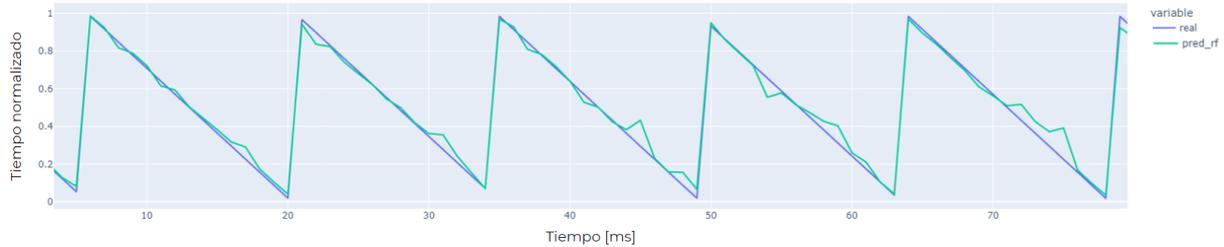
En la Tabla 5.15, destacan los modelos KNN, RF y SVR, que logran las mayores métricas. El modelo que presenta un rendimiento superior en comparación con los otros dos modelos es nuevamente SVR. Este logra un valor de RMSE de 135.99 ms, un valor de MAE de 57.56 ms y un coeficiente de determinación de 0.78, siendo este el más elevado de los obtenidos en las cuatro tareas de regresión.

Los modelos KNN y RF también alcanzan valores óptimos en las métricas estudiadas. En la métrica de RMSE obtienen 137.86 ms y 142.49 ms respectivamente, en la métrica de MAE 49.89 ms y 69.67 ms respectivamente, y en el coeficiente de determinación 0.775 y 0.759 respectivamente. También es destacable sus porcentajes de error respecto al valor máximo del conjunto de test, que son de un 9.84% y 10.17% respectivamente.

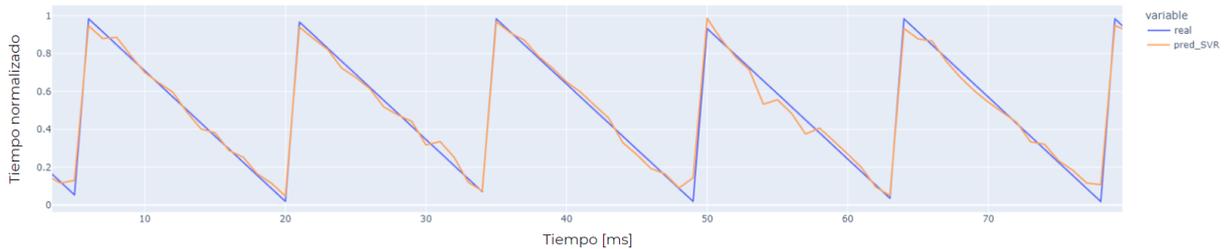
En la Figura 5.7 se observa un ejemplo de la predicción realizada en contraste con el valor real de los modelos KNN, RF y SVR.



(a) Resultado de la predicción del tiempo restante hasta SW con KNN.



(b) Resultado de la predicción del tiempo restante hasta SW con RF.



(c) Resultado de la predicción del tiempo restante hasta SW con SVR.

Figura 5.7: Resultados de la predicción del tiempo restante hasta SW.

Al graficar los valores reales frente a los estimados con estos modelos, se pone de manifiesto los resultados favorables. Como se puede observar, esta tarea de regresión es la que presenta menor error, haciendo que se obtengan resultados altamente satisfactorios; y una alta concordancia entre los valores predichos y los valores reales. Por otra parte, los modelos de regresión lineal y *AdaBoost* son los que exhiben un rendimiento inferior. En concreto, destacan por su alto porcentaje de error respecto del mayor valor de test, que para ambos modelos es de un 19.61 %.

Con el objetivo de comparar de forma clara los resultados obtenidos en las diferentes fases, en la Figura 5.8 se muestran los gráficos de barras de las métricas RMSE, MAE y R^2 para cada uno de los modelos.

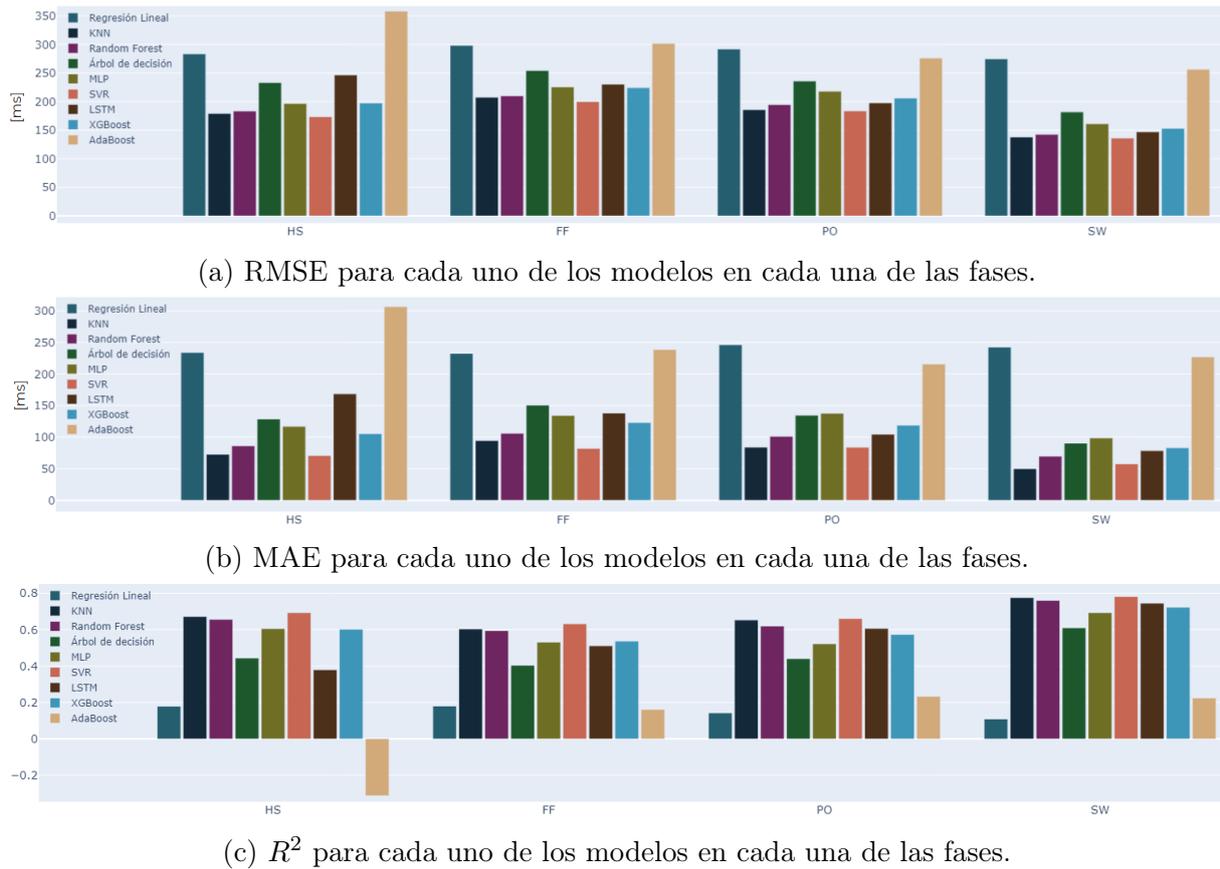


Figura 5.8: Representación de las métricas de regresión para cada uno de los modelos en cada una de las fases.

Analizando la Figura 5.8, se aprecia de manera más evidente los modelos que destacan. Los modelos que exhiben mayor desempeño son KNN, RF y SVR. Por el contrario, los modelos de regresión lineal y *AdaBoost* muestran predicciones menos precisas, siendo especialmente desfavorables para este último. Esto se refleja principalmente en el coeficiente R^2 , que en la predicción del tiempo restante hasta HS con *AdaBoost* obtiene un valor negativo (ver Tabla 5.12), lo cual significa que la regresión es menos precisa que simplemente considerar la media de los datos. La causa de este bajo rendimiento es su diseño, ya que se basa en la combinación de modelos simples que no capturan la complejidad de los datos procedentes de los sensores inerciales.

Finalmente, en términos generales los mejores resultados se obtienen al predecir el tiempo restante hasta la fase SW (ver Tabla 5.15). Esto se debe a que esta fase se inicia con el despegue del pie del suelo, lo cual implica un cambio significativo en la señal recogida. Gracias a este cambio brusco, las predicciones de los modelos muestran una mayor precisión. Por otro lado, en relación al rendimiento en la predicción del tiempo hasta las demás fases, no se observan diferencias notables.

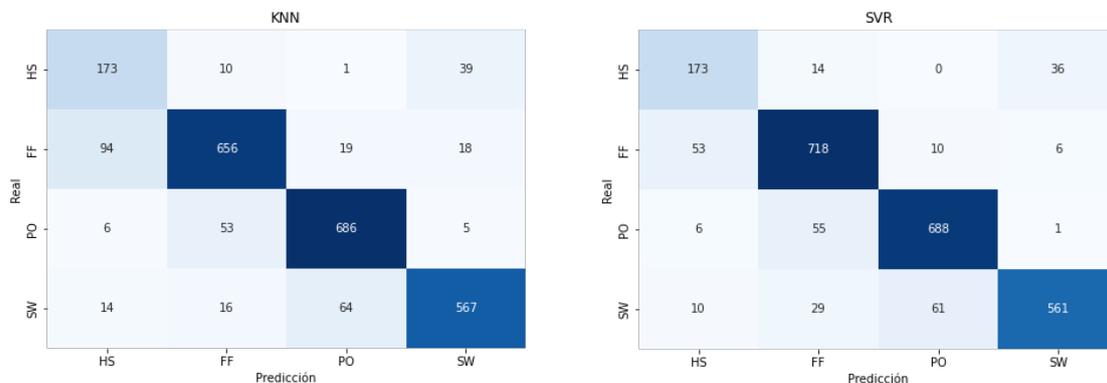
5.5.5. Clasificación a partir de los resultados obtenidos en regresión

Por último, se utilizan los resultados obtenidos de la predicción del tiempo restante hasta cada una de las fases para realizar la segmentación de las fases de la marcha. Para ello, se emplea el sistema de reglas descrito en la Sección 4.4.5, utilizando los tiempos predichos mediante los modelos KNN y SVR. Estos modelos se seleccionan debido a que han obtenido menores errores y mayores valores de R^2 en la tarea de regresión. Los resultados obtenidos para la segmentación de las fases basados en la predicción del tiempo restante hasta los principales eventos de la marcha se muestran en la Tabla 5.16.

Tabla 5.16: Resultados del Experimento PE para clasificar las fases de la marcha.

Modelos	<i>Acc</i>	<i>F1</i>	<i>Rec</i>	<i>Prec</i>	T-HS	T-FF	T-PO	T-SW
KNN	0.859	0.862	0.859	0.867	173	656	686	567
SVR	0.883	0.884	0.883	0.886	173	718	688	561
Total	-	-	-	-	223	787	750	661

Las mayores métricas se obtienen con el modelo SVR, aunque KNN también obtiene resultados por encima de 0.85. Destaca la *Acc* obtenida a partir de las predicciones del modelo SVR, siendo esta de 0.883. Por su parte, el modelo KNN obtiene un valor de *Acc* ligeramente inferior, en concreto de 0.859. En la Figura 5.9 se muestran las matrices de confusión de los dos modelos.



(a) Matriz de confusión KNN.

(b) Matriz de confusión SVR.

Figura 5.9: Matriz de confusión del experimento R.

Analizando la Figura 5.9, destaca el número de observaciones correctamente clasificadas, especialmente para el modelo SVR, aunque ambos modelos presentan un alto rendimiento. Por su parte, el modelo KNN, clasifica correctamente 173 de 223 observaciones de la clase HS; 656 de 787 observaciones de la clase FF, 686 de 750 observaciones de la clase PO y 567 de 661 observaciones de la clase SW. Para el modelo SVR, se clasifican correctamente 173 de 223 observaciones de la clase HS, 718 de 787 observaciones

de la clase FF, 688 de 750 observaciones de la clase PO y 561 de 661 observaciones de la clase SW.

Observando estos resultados, se pone de manifiesto su relación con los obtenidos en la predicción de los tiempos restantes hasta cada una de las fases. En la tarea de predicción del tiempo restante hasta cada una de las fases, se observa que el tiempo que se estima con mayor precisión es el tiempo restante hasta la fase SW. Esto tiene relación con los resultados obtenidos en la tarea de clasificación, donde los modelos KNN y SVR obtienen la mayor *Prec* en la fase SW, con unos resultados de 0.901 y 0.928, respectivamente.

Para el resto de fases, la tarea de predicción del tiempo restante obtiene unos resultados similares. Sin embargo, en la tarea de clasificación destacan los resultados desfavorables en la segmentación de la fase HS. Esto se debe a la corta duración de esta fase, lo que provoca que se encuentre subrepresentada en el conjunto de clases, afectando al proceso de aprendizaje de los modelos.

5.6. Comparativa entre los experimentos realizados

Al comparar los resultados obtenidos en los experimentos realizados con los datos del primer voluntario, se extraen las siguientes conclusiones. El experimento que reporta métricas más elevadas es el experimento DS, logrando un *Acc* de 0.942 utilizando el modelo RF. Estos resultados se deben a la creación de datos sintéticos para lograr un balanceo entre las clases, y a la estandarización de datos. Por su parte, el experimento HV, obtiene métricas ligeramente inferiores de hasta 0.936 al emplear el conjunto de datos en crudo y el modelo RF. Sin embargo, al realizar la clasificación a partir de la predicción de los tiempos restantes hasta cada fase, el modelo SVR obtiene las métricas más elevadas, aunque estas son inferiores a 0.89. A pesar del empeoramiento del rendimiento en comparación con los experimentos anteriores, el estudio del tiempo restante hasta los principales eventos de la marcha proporciona información adicional que no proporciona el resto de experimentos. En este contexto sería interesante evaluar el rendimiento de este método con un mayor número de sujetos.

En segundo lugar, al realizar una comparativa entre los experimentos que utilizan datos de cinco voluntarios, se evidencia que el mejor rendimiento se obtiene al entrenar los modelos con los datos del individuo que se desea evaluar. La mayor *Acc* de la media del experimento VI se logra con el modelo *XGBoost* y es 0.912 (ver Tabla 5.9); mientras que la mayor media del experimento NI se alcanza con SVC con un valor de 0.713 (ver Tabla 5.11). Estos resultados demuestran que el rendimiento de los modelos mejora al contar con información previa del voluntario para realizar el entrenamiento.

En relación al estado del arte estudiado en el Capítulo 2, este TFG aporta una com-

parativa entre diversos métodos y algoritmos de aprendizaje automático utilizados en el análisis de la marcha humana. De esta forma, se determina que los algoritmos más adecuados son RF, *XGBoost* y SVM, sirviendo de base para futuros estudios de segmentación o análisis de la marcha basados en una segmentación inicial de la misma. Además, una característica destacada de este estudio es el hecho de que se trabaja con un único sensor inercial colocado en el pie, a diferencia de estudios previos en del el estado del arte, que emplean múltiples sensores. Esto conlleva un aumento significativo en los costos asociados al análisis de la marcha.

Capítulo 6

Conclusiones y Líneas Futuras

En este capítulo se comentan las principales conclusiones extraídas del estudio, así como las limitaciones, líneas futuras e impacto social. En la Sección 6.1 se detallan los principales hallazgos de la investigación. En la Sección 6.2 las limitaciones y futuras líneas de investigación. La Sección 6.3 resume las consecuencias sociales, medioambientales y económicas de este TFG. Por último, en la Sección 6.4 se describen las lecciones aprendidas en el desarrollo de este TFG.

6.1. Conclusiones

Durante la realización de este TFG se realiza un estudio de la marcha humana enfocada en su segmentación. La segmentación de las fases de la marcha y la predicción del tiempo restante hasta el comienzo de estas fases pueden ser de gran utilidad en diversas áreas tales como la evaluación clínica, la rehabilitación o el diseño de dispositivos de ayuda a la marcha. De esta forma, se ayudaría a mejorar la calidad de vida de las personas que sufren traumatismos o patologías que afectan a la marcha.

Los resultados obtenidos demuestran que el uso de técnicas de aprendizaje automático es efectivo para la segmentación de la marcha humana y sugiere que estas técnicas tienen un gran potencial para ser utilizadas en el análisis de este movimiento. Además, el uso de señales obtenidas de sensores inerciales en entornos no controlados permite que los resultados obtenidos sean útiles en aplicaciones como monitorización de la marcha en situaciones cotidianas fuera del laboratorio.

Se estudian tres conjuntos de datos como entradas en los clasificadores de las fases de marcha: datos en crudo, características estadísticas y características polinomiales. Se observa que el uso de los datos en crudo proporciona mejores resultados en la segmentación. Aunque la extracción de características estadísticas obtiene resultados favorables, no logra superar los obtenidos con la utilización de datos en crudo. En concreto, la mayor tasa de

acierto, o *Acc*, obtenida con el entrenamiento de los datos en crudo es 0.936, mientras que en el caso de extracción de características estadísticas es 0.932. En relación a los resultados obtenidos con la extracción de características polinomiales, se denota un empeoramiento de los resultados. En concreto, las mayores métricas de *Acc* se encuentran por debajo de 0.88.

En relación con los modelos utilizados para realizar el entrenamiento de la tarea de clasificación, se observa que algunos modelos como KNN, RF, *XGBoost* y SVC muestran un rendimiento destacado. Además, se demuestra que el uso de técnicas como validación cruzada para determinación de hiperparámetros, estandarización de datos y creación de datos sintéticos es muy útil para mejorar el rendimiento de los modelos y, por tanto, mejorar los resultados obtenidos. Como muestra de esto, en la tarea de segmentación de las fases de la marcha el mejor resultado se obtiene en el experimento DS, en el que se aplican técnicas de estandarización y balanceo de datos. En dicho experimento se logran tasas de acierto de hasta un 94.2%. Por otra parte, los algoritmos que reportan métricas menos favorables en la tarea de clasificación son los modelos *AdaBoost* y regresión logística, además de la red LSTM cuando no se aplican los métodos de estandarización y balanceo de datos. Por lo tanto, no se recomienda utilizar estos modelos para la tarea de segmentación de las fases de la marcha.

En relación a la tarea de predicción de los tiempos restantes hasta cada una de las cuatro fases, los modelos KNN, RF y SVR presentan un rendimiento destacado. En concreto, se alcanzan valores de R^2 de hasta 0.75 en la predicción del tiempo restante hasta la fase SW. Los modelos que presentan mayor error son *AdaBoost*, regresión lineal y árbol de decisión, alcanzando RMSE relativos superiores al 14%. Por tanto, ninguno de estos modelos se considera adecuado para abordar la tarea de regresión. Con respecto a la segmentación de las fases de la marcha basada en la predicción de los tiempos restantes hasta cada una de las fases, aunque esta alcanza buenos resultados, no logra superar los resultados obtenidos en otros experimentos. Por ejemplo, el *Acc* obtenido en la segmentación de las fases de la marcha a partir de la predicción de los tiempos restantes con el modelo SVR es de 0.883. Sin embargo, el *Acc* obtenido por el experimento DS con el modelo SVC es de 0.934.

Además, cabe destacar que el entrenamiento de los modelos utilizando datos del voluntario que se desea evaluar proporciona mejores resultados en comparación con la segmentación de las fases de la marcha de un individuo cuyos datos no forman parte del conjunto de entrenamiento. En concreto, se logran métricas de *Acc* de más de 0.91 al entrenar los modelos con datos del voluntario que se evalúa, mientras que al entrenar los modelos con un conjunto de datos distinto al empleado para la evaluación, las métricas reportadas son inferiores a 0.75. Esto resalta la importancia de la adaptación de los modelos a las características individuales de cada individuo. Al utilizar datos del individuo que se desea

evaluar para entrenar, se consigue capturar las particularidades de su patrón de marcha, mejorando así el rendimiento de los modelos.

En resumen, los resultados obtenidos en este TFG muestran que es posible realizar segmentación de las fases de la marcha y predicción de eventos con técnicas de aprendizaje automático. Además se demuestra que estas técnicas son efectivas y permiten obtener resultados precisos. Esto es de gran utilidad en diversas aplicaciones, desde la evaluación clínica hasta el diseño de dispositivos de ayuda a la marcha.

6.2. Limitaciones y líneas futuras

En este estudio se han encontrado algunas limitaciones que deben tenerse en cuenta al analizar los resultados alcanzados. Estas limitaciones tienen implicaciones en la generalización de las conclusiones obtenidas.

En primer lugar, se ha utilizado una muestra reducida de la base de datos original. Los datos empleados para esta investigación han sido solo de 5 voluntarios, lo cual no permite extraer una conclusión global que sea aplicable a cualquier individuo. Además, todos estos voluntarios presentaban un buen estado de salud. Para solucionar estas limitaciones, habría que utilizar los datos de más voluntarios, incluyendo también pacientes con patologías relacionadas con la marcha.

Otra de las limitaciones relacionada con los datos empleados tiene que ver con el hecho de que se han utilizado únicamente las señales en las que el voluntario estaba caminando en entornos llanos. Para hacer frente a esto, sería interesante aplicar los algoritmos a entornos con rampas, escaleras o en los que el voluntario está corriendo.

Además, para mejorar las métricas de clasificación de los modelos de aprendizaje automático, sería importante considerar que la marcha depende de factores más allá de las señales recogidas, como la edad, sexo, morfología y estado de salud del voluntario. Por lo tanto, sería interesante para futuros estudios, incluir en el entrenamiento de los modelos de aprendizaje automático datos tabulares adicionales que proporcionen información sobre estos factores y su impacto en la marcha.

Otro aspecto a mejorar tiene relación con las configuraciones utilizadas para los experimentos realizados. En primer lugar, en relación al conjunto de datos empleado, sería beneficioso ampliar el grado del polinomio en la extracción de características polinomiales para así capturar de manera más precisa la complejidad de las relaciones entre los datos. Además, en cuanto al entrenamiento de los modelos, hay múltiples algoritmos en los que no se ha llevado a cabo una validación cruzada para determinar los hiperparámetros. Una futura línea de investigación consistiría en realizar una selección de hiperparámetros más exhaustiva, empleando validación cruzada *k-fold* para todos los modelos. En el caso del

modelo LSTM, también sería interesante ampliar la estructura de la red, incluyendo más capas y evaluando si esto mejora las métricas reportadas.

También es importante destacar que para afrontar el problema de regresión se ha trabajado con los datos de un único voluntario. Sería conveniente realizar pruebas con otros voluntarios y llevar a cabo una validación cruzada en la que se entrene con un conjunto de voluntarios y se valide con un voluntario externo; tal y como se ha realizado para el problema de clasificación. Esto permitiría ampliar la aplicabilidad de los resultados obtenidos, ya que, por ejemplo, en dispositivos de ayuda a la rehabilitación, las necesidades y patologías de cada paciente pueden variar. En el caso de patologías como el accidente cerebrovascular, se podrían utilizar las señales del lado no afectado para entrenar los modelos, mientras que para patologías que afectan a ambos miembros inferiores, como el Parkinson, será necesario utilizar datos de otros pacientes con una marcha normal. En definitiva, la ampliación del conjunto de datos utilizados para entrenar y validar los modelos podría mejorar la robustez y fiabilidad de los resultados obtenidos. Esto, a su vez, permitiría obtener conclusiones más sólidas y generalizables.

6.3. Impacto

A continuación se describe el impacto que tiene este TFG en el ámbito social, medioambiental y económico.

6.3.1. Impacto Social

El análisis de la marcha es de gran importancia en diversos ámbitos, debido a su relación con enfermedades motoras y cognitivas. Este análisis normalmente depende de la detección de las fases de la marcha, las cuales se analizan de manera individual para extraer sus características relevantes [15, 24, 31].

Con el objetivo de segmentar y predecir los principales eventos que componen el ciclo de la marcha, se ha propuesto el uso de algoritmos de aprendizaje automático que emplean datos de IMUs recogidos en entornos no controlados. Esto permitiría la monitorización de los individuos, contribuyendo así a mejorar tanto el diagnóstico como el tratamiento asociados a estas patologías, y en última instancia, la calidad de vida de las personas afectadas.

6.3.2. Impacto Medioambiental

Para llevar a cabo este estudio, se han utilizado señales recogidas a partir de FSRs e IMUs. El impacto medioambiental asociado a esto tiene relación con los recursos natu-

rales y energía que requiere su producción. Además, la eliminación de estos puede generar residuos electrónicos y químicos que pueden ser perjudiciales para el medio ambiente. Sin embargo, es importante destacar que este TFG contribuye a mitigar este problema al reducir la cantidad de sensores necesarios para realizar un análisis de la marcha.

Sumado a ello, debido a que este TFG involucra el empleo de algoritmos de aprendizaje automático, se requiere el uso de recursos informáticos, como ordenadores, que consumen gran cantidad de energía eléctrica durante la fase de entrenamiento y evaluación de los modelos. La generación de electricidad implica la quema de combustibles fósiles como el carbón, petróleo o gas natural. Estos combustibles liberan grandes cantidades de dióxido de carbono (CO_2) y otros gases de efecto invernadero a la atmósfera, lo que contribuye al calentamiento global y al cambio climático [76].

6.3.3. Impacto Económico

Por último, el impacto económico radica en el bajo coste que presentan los sensores inerciales. Esto permite que se realicen estudios a un menor costo, mejorando así el acceso a dichos estudios por parte de poblaciones desfavorecidas. Además, es importante destacar que se están comenzando a desarrollar prendas de ropa que contienen estos sensores [77], lo que permitirá realizar un análisis del movimiento de manera más sencilla en el futuro. Los algoritmos desarrollados en este TFG se adelantan a esta tendencia, proponiendo la aplicación de la IA para llevar a cabo estudios relacionados con el movimiento.

En relación con el impacto social, medioambiental y económico, es importante destacar como este TFG se enmarca dentro de los Objetivos de Desarrollo Sostenible (ODS). Los ODS son una serie de 17 objetivos globales establecidos en 2015 por la Organización de las Naciones Unidas (ONU) como parte de su agenda 2030 para el desarrollo sostenible [78]. Estos objetivos tienen como finalidad abordar los desafíos mundiales más urgentes tratando de promover un mundo más justo, equitativo y sostenible para todos. En la Tabla 6.1 se recogen aquellos objetivos y metas a los que aporta un beneficio el estudio llevado a cabo en esta memoria.

Tabla 6.1: Metas específicas de los ODS que se adecuan a este TFG.

Objetivo	Metas	Descripción
ODS 1: Fin de la pobreza.	ODS 1.4	Para 2030, garantizar que todos los hombres y mujeres, en particular los pobres y los más vulnerables, tengan los mismos derechos a los recursos económicos, así como acceso a los servicios básicos, la propiedad y el control de las tierras y otros bienes, la herencia, los recursos naturales, las nuevas tecnologías y los servicios económicos, incluida la microfinanciación.
ODS 3: Salud y bienestar	ODS 3.8	Lograr la cobertura sanitaria universal, en particular la protección contra los riesgos financieros, el acceso a servicios de salud esenciales de calidad y el acceso a medicamentos y vacunas seguros, eficaces, asequibles y de calidad para todos.
	ODS 3.d	Reforzar la capacidad de todos los países, en particular los países en desarrollo, en materia de alerta temprana, reducción de riesgos y gestión de los riesgos para la salud nacional y mundial.
ODS 7: Energía asequible y no contaminante.	ODS 7.a	De aquí a 2030, aumentar la cooperación internacional para facilitar el acceso a la investigación y la tecnología relativas a la energía limpia, incluidas las fuentes renovables, la eficiencia energética y las tecnologías avanzadas y menos contaminantes de combustibles fósiles, y promover la inversión en infraestructura energética y tecnologías limpias.
ODS 9: Industria, innovación e infraestructura.	ODS 9.5	Aumentar la investigación científica y mejorar la capacidad tecnológica de los sectores industriales de todos los países, en particular los países en desarrollo, entre otras cosas fomentando la innovación y aumentando considerablemente, de aquí a 2030, el número de personas que trabajan en investigación y desarrollo por millón de habitantes y los gastos de los sectores público y privado en investigación y desarrollo.
	ODS 9.b	Apoyar el desarrollo de tecnologías, la investigación y la innovación nacionales en los países en desarrollo, incluso garantizando un entorno normativo propicio a la diversificación industrial y la adición de valor a los productos básicos, entre otras cosas.

6.4. Lecciones aprendidas

Durante el desarrollo de este TFG se han ampliado los conocimientos del Grado en Ingeniería Biomédica en diversas áreas. En primer lugar, se ha profundizado en el estudio de la marcha humana y su análisis. Mediante la utilización de IMUs, se han explorado las fases y eventos que componen el ciclo de la marcha, lo que ha permitido comprender en mayor profundidad esta actividad tan importante. Además, se han aplicado las nociones de programación adquiridas a lo largo de la carrera, en el desarrollo de algoritmos de ML y DL. También se han adquirido conocimientos sobre diversas técnicas como el balanceo de datos o validación cruzada, que han sido fundamentales para mejorar el rendimiento de los modelos utilizados. Por último, se ha aprendido a redactar documentos científicos empleando $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, comprendiendo la estructura y la organización propias de este tipo de documentos.

Bibliografía

- [1] Adalbert I Kapandji et al. *Fisiología articular*, volume 1. Médica Panamericana, 1998.
- [2] Noriko Hatanaka, Kota Sato, Nozomi Hishikawa, Mami Takemoto, Yasuyuki Ohta, Toru Yamashita, and Koji Abe. Comparative gait analysis in progressive supranuclear palsy and parkinson’s disease. *European neurology*, 75(5-6):282–289, 2016.
- [3] Christian Fisahn, Mirko Aach, Oliver Jansen, Marc Moisi, Angeli Mayadev, Krystle T Pagarigan, Joseph R Dettori, and Thomas A Schildhauer. The effectiveness and safety of exoskeletons as assistive and rehabilitation devices in the treatment of neurologic gait disorders in patients with spinal cord injury: a systematic review. *Global spine journal*, 6(8):822–841, 2016.
- [4] Birol Balaban and Fatih Tok. Gait disturbances in patients with stroke. *Pm&R*, 6(7):635–642, 2014.
- [5] Maria Constantinou, Rod Barrett, Mark Brown, and Peter Mills. Spatial-temporal gait characteristics in individuals with hip osteoarthritis: a systematic literature review and meta-analysis. *journal of orthopaedic & sports physical therapy*, 44(4):291–B7, 2014.
- [6] Bastiaan R Bloem, Michael S Okun, and Christine Klein. Parkinson’s disease. *The Lancet*, 397(10291):2284–2303, 2021.
- [7] Bradley Hobbs and Panagiotis Artemiadis. A review of robot-assisted lower-limb stroke therapy: unexplored paths and future directions in gait rehabilitation. *Frontiers in neurorobotics*, 14:19, 2020.
- [8] Du Hyun Ro, Joonhee Lee, Jangyun Lee, Jae-Young Park, Hyuk-Soo Han, and Myung Chul Lee. Effects of knee osteoarthritis on hip and ankle gait mechanics. *Advances in orthopedics*, 2019, 2019.
- [9] Freshta Mohammadzada, Carl Moritz Zipser, Chris A Easthope, David M Halliday, Bernard A Conway, Armin Curt, and Martin Schubert. Mind your step: Target wal-

- king task reveals gait disturbance in individuals with incomplete spinal cord injury. *Journal of NeuroEngineering and Rehabilitation*, 19(1):36, 2022.
- [10] Casey K Kerrigan. *Gait analysis in the science of rehabilitation*, volume 2. Diane Publishing, 2000.
- [11] Anika Steinert, Igor Sattler, Karen Otte, Hanna Röhling, Sebastian Mansow-Model, and Ursula Müller-Werdan. Using new camera-based technologies for gait analysis in older adults in comparison to the established gaitrite system. *Sensors*, 20(1):125, 2019.
- [12] Reed D Gurchiek, Cole P Garabed, and Ryan S McGinnis. Gait event detection using a thigh-worn accelerometer. *Gait & posture*, 80:214–216, 2020.
- [13] Francesco Di Nardo, Christian Morbidoni, Alessandro Cucchiarelli, and Sandro Fioretti. Recognition of gait phases with a single knee electrogoniometer: A deep learning approach. *Electronics*, 9(2):355, 2020.
- [14] Yunus Celik, Sam Stuart, Wai Lok Woo, and Alan Godfrey. Gait analysis in neurological populations: Progression in the use of wearables. *Medical Engineering & Physics*, 87:9–29, 2021.
- [15] Prithvi Patil, K Shusheel Kumar, Neha Gaud, and Vijay Bhaskar Semwal. Clinical human gait classification: extreme learning machine approach. In *2019 1st international conference on advances in science, engineering and robotics technology (ICASERT)*, pages 1–6. IEEE, 2019.
- [16] José Henry Osorio and Mauricio Hernando Valencia. Bases para el entendimiento del proceso de la marcha humana. *Archivos de Medicina (Col)*, 13(1):88–96, 2013.
- [17] Taimoor Afzal, Shih-Chiao Tseng, John A Lincoln, Marcie Kern, Gerard E Francisco, and Shuo-Hsiu Chang. Exoskeleton-assisted gait training in persons with multiple sclerosis: a single-group pilot study. *Archives of physical medicine and rehabilitation*, 101(4):599–606, 2020.
- [18] Francisco Anaya, Pavithra Thangavel, and Haoyong Yu. Hybrid fes–robotic gait rehabilitation technologies: a review on mechanical design, actuation, and control strategies. *International journal of intelligent robotics and applications*, 2:1–28, 2018.
- [19] Aodhán Hickey, Silvia Del Din, Lynn Rochester, and Alan Godfrey. Detecting free-living steps and walking bouts: validating an algorithm for macro gait analysis. *Physiological measurement*, 38(1):N1, 2016.

- [20] Sampath Dakshina Murthy Achanta, T Karthikeyan, et al. A wireless iot system towards gait detection technique using fsr sensor and wearable iot devices. *International journal of intelligent unmanned systems*, 8(1):43–54, 2019.
- [21] Trupti Gujarathi and Kalyani Bhole. Gait analysis using imu sensor. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–5. IEEE, 2019.
- [22] Soumya K Manna, MA Hannan Bin Azhar, and Ann Greace. Optimal locations and computational frameworks of fsr and imu sensors for measuring gait abnormalities. *Heliyon*, 2023.
- [23] Christian Morbidoni, Alessandro Cucchiarelli, Sandro Fioretti, and Francesco Di Nardo. A deep learning approach to emg-based classification of gait phases during level ground walking. *Electronics*, 8(8):894, 2019.
- [24] Johnny D Farah, Natalie Baddour, and Edward D Lemaire. Gait phase detection from thigh kinematics using machine learning techniques. In *2017 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pages 263–268. IEEE, 2017.
- [25] Johnny D Farah, Natalie Baddour, and Edward D Lemaire. Design, development, and evaluation of a local sensor-based gait phase recognition system using a logistic model decision tree for orthosis-control. *Journal of neuroengineering and rehabilitation*, 16(1):1–11, 2019.
- [26] Florian Weigand, Andreas Höhl, Julian Zeiss, Ulrich Konigorski, and Martin Grimmer. Continuous locomotion mode recognition and gait phase estimation based on a shank-mounted imu with artificial neural networks. *arXiv preprint arXiv:2208.00861*, 2022.
- [27] Preeti Khara and Neelesh Kumar. Role of machine learning in gait analysis: a review. *Journal of Medical Engineering & Technology*, 44(8):441–467, 2020.
- [28] Viktor Losing and Martina Hasenjäger. A multi-modal gait database of natural everyday-walk in an urban environment. *Scientific Data*, 9(1):1–12, 2022.
- [29] Wei Li, Yuanbo Chai, Fazlullah Khan, Syed Rooh Ullah Jan, Sahil Verma, Varun G Menon, and Xingwang Li. A comprehensive survey on machine learning-based big data analytics for iot-enabled smart healthcare system. *Mobile networks and applications*, 26:234–252, 2021.

- [30] Arif Reza Anwary, Hongnian Yu, and Michael Vassallo. An automatic gait feature extraction method for identifying gait asymmetry using wearable sensors. *Sensors*, 18(2):676, 2018.
- [31] Yi Chiew Han, Kiing Ing Wong, and Iain Murray. Gait phase detection for normal and abnormal gaits using imu. *IEEE Sensors Journal*, 19(9):3439–3448, 2019.
- [32] Chrsitine F Martindale, Vincent Christlein, Philipp Klumpp, and Bjoern M Eskofier. Wearables-based multi-task gait and activity segmentation using recurrent neural networks. *Neurocomputing*, 432:250–261, 2021.
- [33] MD Akhtaruzzaman, Amir Akramin Shafie, and Md Raisuddin Khan. Gait analysis: Systems, technologies, and importance. *Journal of Mechanics in Medicine and Biology*, 16(07):1630003, 2016.
- [34] Jianfu Yang, Tzu-Hao Huang, Shuangyue Yu, Xiaolong Yang, Hao Su, Ann M Spungen, and Chung-Ying Tsai. Machine learning based adaptive gait phase estimation using inertial measurement sensors. In *Frontiers in Biomedical Devices*, volume 41037, page V001T09A010. American Society of Mechanical Engineers, 2019.
- [35] Xinyu Wu, Ye Yuan, Xikun Zhang, Can Wang, Tiantian Xu, and Dacheng Tao. Gait phase classification for a lower limb exoskeleton system based on a graph convolutional network model. *IEEE Transactions on Industrial Electronics*, 69(5):4999–5008, 2021.
- [36] Fangzheng Wang, Lei Yan, and Jiang Xiao. Recognition of the gait phase based on new deep learning algorithm using multisensor information fusion. *Sensors Mater*, 3110:3041, 2019.
- [37] Lei Yan, Tao Zhen, Jian-Lei Kong, Lian-Ming Wang, and Xiao-Lei Zhou. Walking gait phase detection based on acceleration signals using voting-weighted integrated neural network. *Complexity*, 2020, 2020.
- [38] Francisco A. Garcia, Juan C. Pérez-Ibarra, Marco H. Terra, and Adriano A. G. Siqueira. Adaptive algorithm for gait segmentation using a single imu in the thigh pocket. *IEEE Sensors Journal*, 22(13):13251–13261, 2022.
- [39] Chao Huang, Fuping Zhang, Zhengyi Xu, and Jianming Wei. The diverse gait dataset: Gait segmentation using inertial sensors for pedestrian localization with different genders, heights and walking speeds. *Sensors*, 22(4):1678, 2022.
- [40] Unang Sunarya, Yuli Sun Hariyani, Taeheum Cho, Jongryun Roh, Joonho Hyeong, Illsoo Sohn, Sayup Kim, and Cheolsoo Park. Feature analysis of smart shoe sensors for classification of gait patterns. *Sensors*, 20(21):6253, 2020.

- [41] Christopher L Vaughan, Brian L Davis, Jeremy C O’connor, et al. *Dynamics of human gait*. Human Kinetics Publishers, 1992.
- [42] Tino Stöckel, Robert Jacksteit, Martin Behrens, Ralf Skripitz, Rainer Bader, and Anett Mau-Moeller. The mental representation of the human gait in young and older adults. *Frontiers in psychology*, 6:943, 2015.
- [43] Blanca De La Cruz. *Biomecánica de la marcha y la carrera*, 2009.
- [44] Gianna Arencibia Castellanos, Fidel Ernesto Hernández Montero, Jaime Menéndez Álvarez, Juan Raúl, Rodríguez Suárez, and Ailén Pérez Molinet. Estimación de orientación, basada en filtro de kalman, usando unidad de medición inercial sin magnetómetro. *Revista Investigación Operacional*, 41(3):369–378, 2020.
- [45] Home: Technaid - leading motion - human rehabilitation technologies, Oct 2022.
- [46] Chris Perabo. How do force sensitive resistor (fsr sensor) work?, Dec 2016.
- [47] T Lee, Y Kwon, and H Kim. Smart location tracking system using fsr (force sensing resistor). In *International Conference on Artificial Reality and Telexistence*, 2004.
- [48] Isabel Morales, Rafael González-Landaeta, and Franco Simini. Pressure sensors used as bioimpedance plantar electrodes: A feasibility study. In *2021 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pages 1–6. IEEE, 2021.
- [49] Yetao Yang, Ke Wu, Yi Wang, Tao Chen, and Xiang Wang. Two-layered graph-cuts-based classification of lidar data in urban areas. *Sensors*, 19(21):4685, 2019.
- [50] Mostafa Haghi, Akhmadbek Asadov, Andrei Boiko, Juan Antonio Ortega, Natividad Martínez, and Ralf Seepold. Validating force sensitive resistor strip sensors for cardiorespiratory measurement during sleep: A preliminary study. Apr 2023.
- [51] Paola Catalfamo, David Moser, Salim Ghousayni, and David Ewins. Detection of gait events using an f-scan in-shoe pressure measurement system. *Gait & posture*, 28(3):420–426, 2008.
- [52] Rodrigo Alonso. Ia, machine learning y deep learning, ¿cuál es la diferencia?, Feb 2023.
- [53] Amrita Pathak. Principales modelos de aprendizaje automático explicados, Jan 2023.
- [54] Ligdi González. ¿qué es el perceptrón? perceptrón simple y multicapa, Sep 2022.
- [55] Miguel Sotaquirá. ¿qué es la regresión multiclase?, Aug 2018.

- [56] Luis Torres. Regresión logística ¿cómo funciona? ¿qué secretos esconde?, Jan 2021.
- [57] Joaquín Amat Rodrigo. Máquinas de vector soporte (support vector machines, svms), Apr 2017.
- [58] IBM. ¿qué es el algoritmo de k vecinos más cercanos?, 2022.
- [59] Miryan Estela Narváez, Raquel Johanna Moyano Ariasy Moyano Arias, Diego Bernardo Palacios Campana, and Geovanny Augusto Izurieta Guamán. Predicción de clientes potenciales utilizando k vecino más cercano en el área de negocios de la cooperativa riobamba. *Revista Perspectivas*, 4(1):21–26, 2022.
- [60] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.
- [61] Rosana Ferrero and Nacho García. Qué son los árboles de decisión y para qué sirven, May 2020.
- [62] Joaquín Amat Rodrigo. Árboles de decisión, random forest, gradient boosting y c5.0, 2017.
- [63] Scott Fortmann-Roe. Bias and variance, Jun 2012.
- [64] Trevor Hastie, Saharon Rosset, Ji Zhu, and Hui Zou. Multi-class adaboost. *Statistics and Its Interface*, 2(3):349–360, 2009.
- [65] Cómo funciona el algoritmo xgboost-arccgis pro | documentación.
- [66] Ligdi González. ¿qué es el perceptrón? perceptrón simple y multicapa, Sep 2022.
- [67] Luis Castillo and Diego Maldonado. Redes neuronales, Apr 2020.
- [68] Data Science Team. Introducción al concepto de lstm - aprendizaje automático, Nov 2020.
- [69] Christopher Olah. Understanding lstm networks, Aug 2015.
- [70] Shivanshu Aggarwal. The ultimate guide to building your own lstm models, 2023.
- [71] Swastik Satpathy. Overcoming class imbalance using smote techniques, Jan 2021.
- [72] Arvind Padmanabhan. Cross-validation. *Algorithms*, 7:15, 2022.
- [73] Nagesh Singh Chauhan. Métricas de evaluación de modelos en el aprendizaje automático, Sep 2020.

-
- [74] Anders Norén. Aprendizaje automático: Como seleccionar métricas de regresión, Dec 2019.
- [75] Esteban Madrigal. Conoce las métricas de precisión más comunes para modelos de regresión, Nov 2022.
- [76] David Edgar Araiz Huarte and Pública de Navarra España. La inteligencia artificial como agente contaminante: concepto jurídico, impacto ambiental y futura regulación. *Actualidad Jurídica Ambiental*, (130):51–105, 2023.
- [77] Sung-Won Kang, Hyeob Choi, Hyung-Il Park, Byoung-Gun Choi, Hyobin Im, Dongjun Shin, Young-Giu Jung, Jun-Young Lee, Hong-Won Park, Sukyung Park, et al. The development of an imu integrated clothes for postural monitoring using conductive yarn and interconnecting technology. *Sensors*, 17(11):2560, 2017.
- [78] Carlos Gómez Gil. Objetivos de desarrollo sostenible (ods): una revisión crítica. *Papeles de relaciones ecosociales y cambio global*, 140(1):107–118, 2018.