



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

**Grado en Ingeniería Informática
(Mención Computación)**

**Captura de datos de ponibles
(wearables) y su utilización para la
clasificación del tipo de actividad
realizada por una persona**

Autor:

D. Mario Villacorta García

Tutores:

Dña. María Aránzazu Simón Hurtado

D. Carlos Enrique Vivaracho Pascual

Resumen

El reconocimiento de actividades humanas (HAR) constituye un campo de relevancia y amplio desarrollo en el dominio de la inteligencia artificial y la obtención de conocimiento. Tiene aplicaciones en muy diversas disciplinas, desde la medicina hasta la robótica. Mas, a pesar del interés que suscita en tantas áreas del ámbito científico, no existe una aplicación clara del HAR para el reconocimiento biométrico de personas según su forma de actuar.

El presente trabajo constituye una continuación de varios trabajos previos que abordan el reconocimiento biométrico a partir de la forma de caminar, detectada mediante el uso de dispositivos ponibles de carácter comercial. De manera prospectiva, se pretende sentar las bases de conocimiento necesario con el fin de lograr una generalización de los mencionados trabajos mediante el reconocimiento de un conjunto definido de actividades cotidianas en lugar de una sola. Para ello, se lleva a cabo el desarrollo de un sistema de reconocimiento de actividades humanas a partir de datos recogidos de dispositivos ponibles de uso comercial en entornos sin supervisión, mediante técnicas del proceso habitual de minería de datos y aprendizaje artificial, con el fin de que las conclusiones extraídas del sistema HAR desarrollado puedan ser empleadas con posterioridad para llegar a una solución de reconocimiento biométrico conductual de personas a partir de actividades cotidianas.

Abstract

The recognition of human activities (HAR) constitutes a field of relevance and wide development in the domain of artificial intelligence and obtaining knowledge. It has several applications in very different disciplines, from healthcare to robotics. Nevertheless, despite the interest of the matter in so many areas of science, it does not exist a clear HAR approach to the biometric recognition of people based on their way of acting.

This working paper makes a continuation of some previous works that address the matter of biometrics based on the gait, detected through the use of commercial wearable devices. In a prospective manner, it is pretended to lay the groundwork of necessary knowledge in order to achieve a generalization of the aforementioned works through the recognition of a set of daily activities instead of just one. To do this, it will be covered the developing of a human activity recognition system based on data collected from commercial wearable devices in non supervised environments, using techniques of the usual data mining and machine learning process, in order to get the conclusions of the development to be used subsequently to achieve a biometrical conductual recognition solution based on daily activities.

Índice general

1. Introducción	6
1.1. Breve introducción a la biometría	6
1.2. Biometría conductual y reconocimiento de actividades humanas	8
1.3. Objetivos	8
1.3.1. Objetivo general	8
1.3.2. Objetivos específicos	9
1.4. Estructura de la obra	9
2. Planificación y seguimiento	10
2.1. Categorización del proyecto	10
2.2. Plan de acción	10
2.3. Gestión de riesgos	12
2.3.1. Análisis y evaluación de riesgos	12
2.3.2. Plan de contingencia	13
2.4. Seguimiento	13
3. Estado del arte	16
3.1. Introducción al Reconocimiento de Actividades Humanas (HAR)	16
3.2. Aproximaciones al HAR	17
3.2.1. HAR mediante dispositivos externos	18
3.2.2. HAR mediante dispositivos ponibles	18
3.3. Diseño de un sistema HAR	19
3.4. Factores determinantes en la minería de datos para HAR	20
3.4.1. Protocolo de recogida	20
3.4.2. Definición de actividades	20
4. Recogida de datos	22
4.1. Protocolo de recogida	22
4.2. Datos recogidos	23
5. Análisis y procesamiento de datos	26
5.1. Inspección visual de datos	26
5.2. Preprocesamiento	27
5.2.1. Eliminación de regiones de nulo muestreo	27
5.2.2. Interpolación a frecuencia de muestreo fija	29
5.3. Extracción de características	31
5.4. Procesamiento ad hoc	32
5.4.1. Selección y transformación de atributos	32

5.4.2.	Normalización de los datos	35
5.4.3.	Reducción de los datos	36
6.	Minería de datos y evaluación de resultados	37
6.1.	Descripción del dataset procesado	37
6.2.	Metodología de evaluación	38
6.3.	Configuración experimental	39
6.3.1.	Parámetros de experimentación	39
6.3.2.	Algoritmos de aprendizaje	40
6.3.2.1.	Aprendizaje supervisado	40
6.3.2.2.	Aprendizaje no supervisado	43
6.4.	Evaluación de resultados	45
6.4.1.	Rendimiento global	45
6.4.2.	Evaluación individual de clasificadores	48
6.4.2.1.	Árboles de decisión	48
6.4.2.2.	k-NN	50
6.4.2.3.	Naive Bayes	52
6.4.2.4.	SVM	54
6.4.2.5.	MLP	55
6.4.2.6.	Clustering	55
6.4.3.	Comparación de clasificadores	57
7.	Conclusiones y trabajo futuro	62
7.1.	Conclusiones	62
7.2.	Líneas de trabajo futuras	63

Índice de figuras

2.1.	Precedencia de las actividades principales del plan de acción	11
2.2.	Seguimiento de las actividades realizadas durante el desarrollo del trabajo	14
3.1.	Ejemplo del error inducido a través de la relajación del HARP[1]	18
4.1.	Distribución de las mediciones capturadas por sensor en la sesión 1	24
4.2.	Distribución de las mediciones capturadas por usuario en la sesión 1	25
4.3.	Distribución de las mediciones capturadas por actividad en la sesión 1	25
5.1.	Ejemplo de señal cruda correspondiente a la muestra 14 de la sesión 1 del usuario <i>velazquez</i>	27
5.2.	Ejemplo de zonas sin muestreo en señal cruda correspondiente a la componente X de la muestra 14 de la sesión 1 del usuario <i>velazquez</i>	28
5.3.	Ejemplo de zonas de amplitud constante en señal cruda correspondiente a la componente X de la muestra 9 de la sesión 1 del usuario <i>sorolla</i>	28
5.4.	Distribución de los períodos de muestreo en ms antes y después de limpiar las señales de valores atípicos	29
5.5.	Ejemplo de señal limpia de outliers correspondiente a la muestra 14 de la sesión 1 del usuario <i>velazquez</i>	29
5.6.	Ejemplo de zonas sin muestreo en señal limpia correspondiente a la componente X de la muestra 14 de la sesión 1 del usuario <i>velazquez</i>	30
5.7.	Correlación de los atributos de clasificación para el dataset limpio interpolado a 12 y 24 Hz según tamaño de ventana para el sensor acelerómetro	34
5.8.	Correlación de los atributos de clasificación para el dataset limpio interpolado a 12 y 24 Hz según tamaño de ventana para el sensor giroscopio	34
5.9.	Desempeño de la selección de atributos mediante árbol de decisión para el dataset limpio interpolado a 12 y 24 Hz según tamaño de ventana para el sensor acelerómetro	35
5.10.	Desempeño de la selección de atributos mediante árbol de decisión para el dataset limpio interpolado a 12 y 24 Hz según tamaño de ventana para el sensor giroscopio	35
6.1.	Distribución de clases de los datasets procesados según sensor y frecuencia de muestreo	37
6.2.	Visualización del tamaño de los datasets procesados por sensor	38
6.3.	Algoritmos de aprendizaje más populares según el paradigma de aprendizaje automático	40
6.4.	Ejemplo de árbol de decisión para la concesión de préstamos bancarios	41
6.5.	Efecto de la aplicación de funciones kernel en SVM	43
6.6.	Esquema de funcionamiento de un perceptrón simple (neurona) de cinco entradas. Alejandro Cartas, CC BY-SA 4.0, via Wikimedia Commons	44
6.7.	Esquema de un perceptrón multicapa de 2 capas con tres neuronas ocultas para la clasificación binaria. Chrislb, CC BY-SA 3.0, via Wikimedia Commons	44
6.8.	Desempeño según algoritmo de clasificación supervisada para el sensor de acelerómetro	45

6.9. Desempeño según algoritmo de clasificación supervisada para el sensor de giroscopio 46

6.10. Efecto del tamaño de ventana sobre el desempeño de los clasificadores para el sensor de acelerómetro 47

6.11. Efecto del tamaño de ventana sobre el desempeño de los clasificadores para el sensor de giroscopio 47

6.12. Mediana de las tasas de acierto de la clasificación de los datasets para los sensores de acelerómetro y giroscopio 48

6.13. Efecto del sobreajuste en la tasa de error del modelo según su complejidad 49

6.14. Evolución de la tasa de acierto según la profundidad de inducción del árbol de decisión de 1 a 15 nodos de profundidad 49

6.15. Precisión del árbol de decisión para cada clase del sensor de acelerómetro 50

6.16. Precisión del árbol de decisión para cada clase del sensor de giroscopio 51

6.17. Evolución de la tasa de acierto de k-NN según el valor de k para acelerómetro y giroscopio . . 51

6.18. Evolución de la precisión de clase de k-NN según el valor de k sin selección de atributos para acelerómetro y giroscopio 52

6.19. Precisión de Naive Bayes para cada clase del sensor de acelerómetro 53

6.20. Precisión de Naive Bayes para cada clase del sensor de giroscopio 54

6.21. Evolución de la tasa de acierto de SVM según la función kernel para acelerómetro y giroscopio 54

6.22. Evolución de la precisión de clase de SVM según la función kernel sin selección de atributos para acelerómetro y giroscopio 55

6.23. Precisión de MLP para cada clase del sensor de acelerómetro 56

6.24. Precisión de MLP para cada clase del sensor de giroscopio 57

6.25. Evolución de la tasa de acierto de k-medias según el valor de k para acelerómetro y giroscopio 57

6.26. Evolución de la precisión de clase de k-means según el valor de k con PCA para acelerómetro y sin selección de atributos para giroscopio 58

6.27. Evolución de la tasa de acierto de clustering aglomerativo según el número de centros para acelerómetro y giroscopio 58

6.28. Evolución de la precisión de clase de clustering aglomerativo según el número de clusters con PCA para acelerómetro y sin selección de atributos para giroscopio 59

Índice de tablas

2.1. Evaluación de riesgos técnicos	12
2.2. Evaluación de riesgos infraestructurales	12
2.3. Evaluación de riesgos de información	12
2.4. Plan de contingencia de riesgos	13
2.5. Aproximación del tiempo empleado en cada actividad particular del proyecto medido en jornadas de trabajo	15
4.1. Número de mediciones capturadas por sensor y usuario en la sesión 1	24
5.1. Selección de atributos basada en correlación	33
5.2. Selección de atributos basada en árboles de decisión	36
5.3. Selección de atributos mediante ReliefF	36
6.1. Tamaño de los datasets procesados	38
6.2. Resultados de la clasificación mediante árbol de inducción de 8 nodos de profundidad	49
6.3. Resultados de la clasificación mediante k-NN	53
6.4. Resultados de la clasificación mediante Naive Bayes	53
6.5. Resultados de la clasificación mediante SVM con kernel polinómico	55
6.6. Resultados de la clasificación mediante MLP	56
6.7. Resultados de la clasificación mediante k-medias	58
6.8. Resultados de la clasificación mediante clustering aglomerativo	59
6.9. Comparación de la diferencia significativa en el desempeño de la inducción de árbol de decision frente al resto de clasificadores	60
6.10. Comparación de la diferencia significativa en el desempeño de 1-NN frente al resto de clasificadores	61
6.11. Comparación de la diferencia significativa en el desempeño de 8-NN frente al resto de clasificadores	61

Capítulo 1

Introducción

El mundo virtual está cada vez más acoplado al mundo físico. Esta realidad no solo es consecuencia de la digitalización de procesos que tradicionalmente se solían llevar a cabo de forma analógica, sino también de los nuevos servicios que surgen constantemente de forma nativa en el entorno informático. Enviamos correos electrónicos y mensajes de texto, subimos nuestras fotos a la nube y redes sociales, gestionamos nuestras cuentas de banco mediante el móvil e incluso alquilamos servicios de coche con conductor a través de Internet. En nuestro día a día dependemos cada vez más de la informática, lo que conlleva un aumento evidente tanto en la complejidad como en la sensibilidad de la información que manejamos. Los datos a los que estamos asociados son cada vez más y representan cada vez más información. Como consecuencia, demostrar la autenticidad de nuestra identidad se vuelve la piedra angular de cualquier sistema informático moderno.

Raro es el servicio del que hacemos uso de forma impersonal, y esta tendencia se mantendrá al alza casi inexorablemente. Sin embargo, a pesar de este aumento exponencial de los servicios digitales que requieren de nuestra identificación para poder funcionar, los mecanismos que empleamos para ello se han mantenido prácticamente constantes desde que surgieron, hace casi medio siglo. Nada tiene que ver el mundo de ahora con el de los años 80, y pese a todo la forma que tenemos de demostrar nuestra identidad sigue siendo esencialmente la misma, mediante un nombre de usuario y una contraseña. Como consecuencia, se hace evidente la necesidad de un replanteamiento de la autenticación de identidad acorde a las exigencias de la presente sociedad de la información.

1.1. Breve introducción a la biometría

La **autenticación** es el proceso por el cual alguien demuestra que es quien dice ser[2]. A pesar de lo que pueda parecer, este no es un concepto que se circunscriba tan solo a la informática, sino que está presente en prácticamente cualquier ámbito de la vida en sociedad. La cuestión es que lo tenemos tan interiorizado que no nos percatamos de la relevancia y la complejidad que conlleva. Por ejemplo, cuando queremos entrar en casa de algún amigo o familiar y llamamos al timbre para que nos abra la puerta, nuestro huésped tan solo necesita observar por la mirilla para ser capaz de determinar si debe o no dejarnos entrar. En este contexto, nuestra apariencia física autentica nuestra identidad personal[3]. En la vida cotidiana, autenticamos identidades casi sin pensar. Sin embargo, cuando se trata de la autenticación en servicios informáticos, la cuestión no es tan trivial.

Una primera aproximación a este ideal de la autenticación mediante la mirilla de una puerta puede ser la posesión de un **token de identificación único**. Si aseguramos que cada individuo a autenticar en nuestro

sistema posee un elemento único completamente distinguible del que poseen todos los demás, podemos asociar su identidad a la respuesta de la pregunta: '*¿qué posee?*'[3]. Por ejemplo, cuando tenemos un registro horario en el trabajo y utilizamos nuestro carnet de empresa para indicar a qué hora entramos a trabajar, estamos autenticando nuestra identidad a través de la posesión de un token, el carnet. Es una solución simple y efectiva, pero presenta una clara vulnerabilidad: el token empleado constituye **un punto de fallo único**. En el momento en que extraviemos nuestra tarjeta, no podremos fichar en el trabajo. En el momento en que alguien la encuentre, podrá hacerse pasar por nosotros.

Para contrarrestar este inconveniente, surge una segunda aproximación, la autenticación basada en un elemento que nadie pueda perder, un **secreto**. Si cada persona que deseamos autenticar conoce una pieza de información única asociada a su identidad, la autenticación en el sistema se basará en la respuesta a la pregunta: '*¿qué conoce?*'[3], de forma que el proceso será inherentemente más seguro tan solo por una cuestión de cambio en el enfoque paradigmático. Este es el caso del uso de PIN y contraseña o, más actualmente, patrón de desbloqueo.

A pesar de la ventaja comparativa que supone el uso de sistemas de autenticación basados en algo conocido con respecto al de sistemas de autenticación basados en algo poseído[2], no cabe duda de que el secreto aún puede ser robado, si se almacena mal; o adivinado, si es demasiado simple. Según un estudio de DataGenetics, los ladrones son capaces de adivinar los PINs de banco el 25 % de las veces[4]. Además, cabe recordar que hace menos de un año un grupo de hackers reunidos a través de un foro han sido capaces de filtrar más de 8 mil millones de contraseñas de cuentas de Internet[5]. Los secretos no son infalibles. Es este el motivo por el cual el uso conjunto de **múltiples factores de autenticación** puede mejorar la seguridad de la autenticación en el sistema al incrementar la complejidad de la misma.

Para que una autenticación multifactor sea relativamente beneficiosa con respecto a una más simple de un solo factor, conviene que incluya mecanismos correspondientes a distintos paradigmas de autenticación[6]. La implementación de autenticación multifactor interfiere negativamente en la usabilidad de los sistemas, por lo que se suele encontrar este mecanismo únicamente en los procedimientos más sensibles, como en el acceso a la banca online. Normalmente, para acceder a una cuenta de banco a través de Internet es necesario introducir un documento de identidad y PIN y, para realizar cualquier acción, un código SMS enviado al teléfono móvil del usuario. Tanto el DNI como el PIN son parte de la autenticación basada en algo conocido, y el código SMS representa una autenticación basada en algo poseído, el móvil.

Aun con todo, en cualquiera de los casos previamente descritos, si un usuario lleva a cabo una acción malintencionada y esta queda asociada a su identidad basada en autenticación mediante conocimiento de secreto o posesión de token, puede sencillamente **negar plausiblemente** su acción, aduciendo un robo de su identidad y eximirse así de la responsabilidad de sus actos[3]. A priori, nadie podría demostrar que el malhechor miente: tanto el secreto como el token son elementos externos a él. Por lo tanto, toda autenticación basada en posesión o conocimiento, que realmente es posesión de un secreto, presenta vulnerabilidades que, si bien se pueden paliar parcialmente a través del aumento de la complejidad de los mecanismos de autenticación, son inherentes a la propia concepción de los sistemas y en última instancia insalvables.

Consecuentemente, se debe replantear el paradigma de autenticación, es decir, la pregunta a la que debemos contestar para demostrar ser quienes somos. Claramente, el ideal de autenticación debe estar basado en la respuesta a: '*¿quién es?*'[3]. La **biometría** es precisamente la disciplina científica encargada de desarrollar y medir los atributos y características de una persona que pueden ser usados para identificarla de forma única[3]. Tal y como está formulada, la pregunta a la que responde la biometría es demasiado general, por lo que la dividiremos en dos más simples que la sintetizan: '*¿cuáles son sus características físicas?*' y '*¿cómo se comporta?*'. En función de estas últimas, distinguimos dos tipos de modalidades biométricas: la **biometría fisiológica** y la **biometría del comportamiento o conductual**[3]. Esto no nos debe resultar

algo desconocido, al menos no del todo. Cada vez se hace más común el uso de la biometría fisiológica para la autenticación de individuos. Por ejemplo, la mayor parte de los nuevos teléfonos inteligentes incorporan por defecto el desbloqueo a través de huella digital o rostro. Sin embargo, el uso de la biometría conductual es mucho más inusual, dada la alta dificultad de implementación que puede llegar a representar. Esta modalidad constituye un interesante campo de estudio cuya explotación puede dar lugar a sistemas de autenticación de mayor complejidad y seguridad, dado que el comportamiento es, por lo general, más difícil de suplantar que los rasgos fisiológicos de una persona[7].

1.2. Biometría conductual y reconocimiento de actividades humanas

Debido a su alta complejidad y todas las posibilidades que abarca, la forma habitual de implementar la biometría conductual se basa en la monitorización de **la interacción de los usuarios con sus dispositivos electrónicos**. Medir la velocidad de tecleo, cómo se hace scroll en una pantalla o la forma de arrastrar iconos son ejemplos de los parámetros que a día de hoy se intentan explotar con el fin principal de mejorar la experiencia de usuario en la utilización de servicios digitales[8]. Se trata de procedimientos que no comprometen la eficacia ni la eficiencia de los sistemas y pueden ser de gran utilidad y aportar valor a los servicios. Por ejemplo, el aprovechamiento de la forma de interactuar con dispositivos resulta de gran utilidad para la seguridad de las empresas frente a las ciberamenazas. A través de la identificación de los patrones conductuales en las páginas de banca online se puede llegar a evitar el fraude, detectando cuándo se trata de una interacción habitual o de una conducta anómala[9]. Al automatizar el rechazo de esta última se incrementa la seguridad sin tener que comprometer la experiencia de usuario con verificaciones de identidad tediosas como la autenticación multifactor que hemos relatado anteriormente.

Sin embargo, a pesar del desarrollo que está teniendo el empleo de la forma de interactuar con los dispositivos para el reconocimiento biométrico, el uso de **la forma en la que las personas realizan actividades físicas** es una aproximación de gran interés a la biometría conductual que no se está explotando de forma amplia. A día de hoy es bien sabido que la forma de caminar puede ser empleada para el reconocimiento biométrico de personas[10]. En concreto, sobre este tema cabe destacar los tres trabajos de fin de grado que preceden al mío y que tratan esta cuestión de forma amplia, llegando a un sistema de autenticación basado en datos proporcionados por dispositivos ponibles comerciales [11, 12, 13]. Sin embargo, existen pocas referencias sobre el empleo de otro tipo de actividades físicas cotidianas para el reconocimiento biométrico.

La idea del presente proyecto es sentar las bases para la futura extensión del reconocimiento biométrico conductual de personas mediante actividades más allá de caminar. Para ello, llevaremos a cabo un sistema de reconocimiento de actividades humanas a partir de datos recogidos a través de dispositivos ponibles comerciales, de forma que el reconocimiento de actividades pueda ser el punto de partida para la extracción de patrones rutinarios del comportamiento de una persona o la forma concreta que tenga de realizar las actividades reconocidas, con el fin de identificarla de forma única en trabajos posteriores.

1.3. Objetivos

1.3.1. Objetivo general

El objetivo principal del trabajo es el uso de dispositivos ponibles comerciales para la captura de datos y su explotación para el reconocimiento de actividades humanas mediante técnicas de minería de datos y aprendizaje automático.

1.3.2. Objetivos específicos

Para acometer el propósito principal del proyecto, este ha sido subdividido en los siguientes objetivos específicos:

- Recoger datos conductuales de voluntarios para obtener una base de datos que poder explotar tanto en el presente trabajo como en futuros trabajos sobre reconocimiento de actividades.
- Analizar y procesar los datos recogidos de tal manera que sean útiles para la explotación mediante aprendizaje automático.
- Desarrollar un sistema de reconocimiento de actividades humanas a partir de técnicas de aprendizaje automático y evaluarlo en función de distintos parámetros con el fin de determinar la configuración que produce los mejores resultados.

1.4. Estructura de la obra

La presente obra constituye la memoria de un Trabajo de Fin de Grado de Ingeniería Informática por la Universidad de Valladolid.

Siendo el actual el capítulo introductorio, el trabajo consta de los siguientes capítulos:

Capítulo 2. Planificación y seguimiento. En este capítulo se presenta el plan de acción considerado para la realización del trabajo, la gestión de riesgos previa a su desarrollo y el seguimiento del mismo.

Capítulo 3. Estado del arte. En este capítulo se estudian los fundamentos del reconocimiento de actividades humanas, así como los resultados experimentales de trabajos previos considerados de utilidad para el presente desarrollo.

Capítulo 4. Recogida de datos. En este capítulo se explica el protocolo de recogida de datos para la base de actividades y los datos obtenidos para la explotación en el reconocimiento.

Capítulo 5. Análisis y procesamiento de datos. En este capítulo se estudia la naturaleza y características de los datos capturados y se procesan con el fin de su preparación para el reconocimiento de actividades.

Capítulo 6. Minería de datos y evaluación de resultados. En este capítulo se implementan los sistemas de reconocimiento de actividades humanas efectivos a partir de las técnicas más habituales de aprendizaje automático, con el fin de determinar las de mejor comportamiento.

Capítulo 7. Conclusiones y trabajo futuro. En este capítulo se expondrán las conclusiones obtenidas tras el desarrollo del trabajo y se propondrán las posibles líneas de investigación futuras relativas al reconocimiento de actividades humanas.

Capítulo 2

Planificación y seguimiento

Con el fin de llevar a cabo la planificación y el seguimiento del proyecto, se ha optado por implementar una aproximación del proceso de planificación enfocada a la investigación científica[14]. Por otro lado, el modelo de proceso que ha pautado el marco de trabajo está basado en el desarrollo ágil de software aplicado ad hoc al proceso de minería de datos y descubrimiento de conocimiento[15].

2.1. Categorización del proyecto

Los proyectos de investigación se clasifican en función de la naturaleza de las actividades que circunscriben. Para poder categorizar las de la investigación abordada, empleamos la referencia del *Excellence in Research for Australia*[16]. Así, el presente trabajo está enmarcado dentro del campo de **información y ciencias de la computación**, en la disciplina de **inteligencia artificial** para el reconocimiento de actividades humanas (HAR). A su vez, se trata de una **investigación aplicada**, enfocada a la **obtención de resultados** y desarrollada de forma individual[14].

2.2. Plan de acción

Las actividades principales del plan de acción a seguir se basan en el proceso habitual para la minería de datos y descubrimiento de conocimiento[15]. A su vez, debemos tener en cuenta que estamos realizando un trabajo de fin de grado que deberemos documentar y que estamos empleando una aplicación de elaboración propia para la recogida de datos, sujeta a evaluación y mejora continua. Para llevar a cabo nuestro plan de acción, disponemos de un mínimo de 300 horas correspondientes a los 6 créditos ECTS que representa el trabajo de fin de grado, ampliables si la situación así lo requiere. Definimos una **jornada de trabajo** como un periodo de 8 horas de trabajo, en uno o varios días. Se prevé que el desarrollo completo del trabajo de fin de grado tome un periodo total aproximado de 37.5 jornadas de trabajo. Trabajando de lunes a viernes, con opción a recuperar horas durante fines de semana, y dando comienzo el proyecto a fecha 21 de abril de 2021, establecemos como **fecha final de entrega** el día 11 de junio de 2021. Con todo, las principales actividades del proyecto son las siguientes:

- **Mantenimiento de aplicación móvil:** se parte de una aplicación móvil desarrollada previamente durante el periodo de prácticas en empresa para la recogida de datos a usar en el trabajo de fin de grado. Sin embargo, el uso de la aplicación inevitablemente llevará a la detección de errores que deberán ser subsanados a lo largo de todo el proyecto. Puede llegar a representar un cuello de botella, aunque

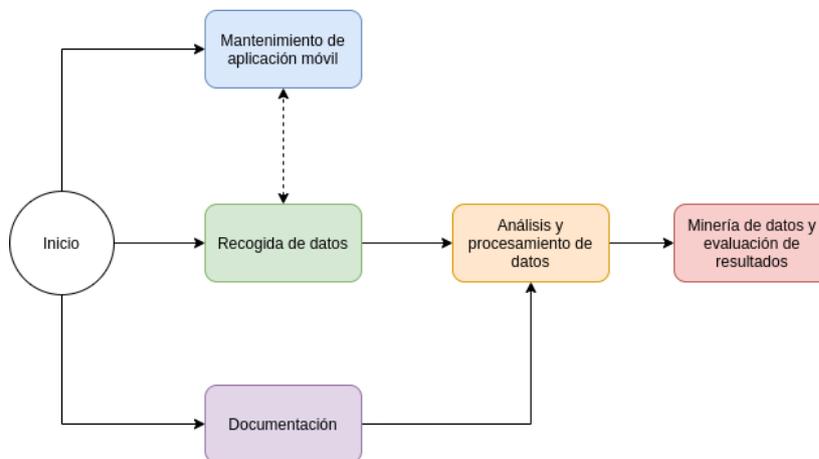


Figura 2.1: Precedencia de las actividades principales del plan de acción

suponemos que no limitará demasiado el proyecto gracias al testeado previo al trabajo de fin de grado. Se estima que esta fase tome no más de 7 jornadas de trabajo.

- **Recogida de datos:** representa el punto de partida del trabajo propiamente dicho. Los datos serán recogidos por voluntarios de forma autónoma e individual con un seguimiento continuo e indicaciones sobre la idoneidad de los datos que se van recogiendo, de tal manera que a pesar de ser un claro punto problemático, se prevé que la obtención gradual de datos permita el desarrollo de fases posteriores de manera simultánea a la obtención del dataset final completo. Esta fase influye decisivamente en el mantenimiento de la aplicación móvil, puesto que es en la que, de existir errores, surgiran los fallos. Se estima que esta fase tome de en torno a las 5 jornadas de trabajo.
- **Análisis y procesamiento de datos:** incluye las operaciones de integración, limpieza, transformación y reducción. Se trata de una actividad crítica para la obtención de resultados de calidad. A esta tarea la precede tanto la lectura de trabajos previos y el análisis de un estado del arte, como la recogida de datos de voluntarios, si bien no completa. Se estima la duración de esta fase en unas 12 jornadas de trabajo.
- **Minería de datos y evaluación de resultados:** son las tareas finales del proceso de reconocimiento de actividades. Consisten en la construcción de clasificadores para la obtención de resultados relativos al reconocimiento de actividades humanas. El desempeño de los clasificadores puede llegar a suponer una vuelta al análisis y procesamiento de datos. Se estima la duración de esta fase en unas 5 jornadas.
- **Documentación:** incluye tanto la lectura de trabajos previos como la documentación del proyecto en desarrollo. Si bien la documentación del proyecto se llevará a cabo de forma paralela al desarrollo del mismo, la lectura de trabajos previos supone un punto de partida esencial para poder comenzar el desarrollo del trabajo, por lo que, con la recogida de datos, representa un cuello de botella en el proceso total. Se estima la duración de esta fase en a unas 9 jornadas.

Encontramos representadas las relaciones de precedencia de las actividades principales en la figura 2.1. Por otro lado, la pormenorización de las tareas relativas a cada actividad principal del plan de acción se llevará a cabo en **reuniones semanales** con los tutores del trabajo, en función del avance del mismo.

2.3. Gestión de riesgos

2.3.1. Análisis y evaluación de riesgos

Para realizar la evaluación de riesgos del proyecto, trataremos de identificar las potenciales amenazas categorizándolas según los tipos principales asociados a proyectos de investigación. Los riesgos considerados son los siguientes[14]:

- **Riesgos técnicos:** son los relativos a la factibilidad del enfoque tomado.
- **Riesgos infraestructurales:** son los relativos a las dependencias entre los activos del proyecto.
- **Riesgos de información:** son los relativos a la disponibilidad de la información crítica necesaria para llevar a cabo las tareas.

A su vez, para poder determinar la evaluación de cada amenaza, empleamos las siguientes escalas cualitativas crecientes de probabilidad, impacto y nivel de riesgo de las amenazas[17]:

- **Probabilidad:** muy improbable, improbable, posible, muy probable, casi seguro.
- **Impacto:** insignificante, menor, serio, desastroso, catastrófico.
- **Nivel de riesgo:** asumible, bajo, apreciable, importante, crítico.

Los niveles de riesgo están definidos según la estimación del producto *probabilidad x impacto* de la guía de evaluación de riesgos del INCIBE[17]. En las tablas 2.1, 2.2 y 2.3 se muestran los riesgos identificados en el presente proyecto.

ID	Descripción	Probabilidad	Impacto	Nivel de riesgo
Tec.1	Imposibilidad de reconocimiento de actividades humanas a partir de wearables	Muy improbable	Catastrófico	Bajo
Tec.2	Tiempo establecido para el trabajo insuficiente	Muy probable	Desastroso	Importante
Tec.3	Voluntarios no saben utilizar aplicación	Muy probable	Catastrófico	Crítico
Tec.4	Desconocimiento de tecnologías en uso	Posible	Catastrófico	Importante

Tabla 2.1: Evaluación de riesgos técnicos

ID	Descripción	Probabilidad	Impacto	Nivel de riesgo
Infra.1	Aplicación de recogida de datos falla	Muy probable	Catastrófico	Crítico
Infra.2	Reloj inteligente falla	Posible	Desastroso	Apreciable
Infra.3	Servidor falla	Muy improbable	Desastroso	Asumible
Infra.4	Frecuencia de muestreo insuficiente	Muy improbable	Desastroso	Asumible
Infra.5	Dirección IP pública cambia y servidor back-end no permite conexión	Casi seguro	Insignificante	Asumible
Infra.6	Voluntario rehusa continuar participando en el proyecto	Posible	Serio	Bajo

Tabla 2.2: Evaluación de riesgos infraestructurales

ID	Descripción	Probabilidad	Impacto	Nivel de riesgo
Info.1	Datos recogidos no son suficientes	Muy probable	Desastroso	Importante
Info.2	Datos recogidos no corresponden a actividad indicada	Posible	Catastrófico	Importante
Info.3	Datos presentan ruido asistemático	Muy probable	Serio	Apreciable

Tabla 2.3: Evaluación de riesgos de información

Como podemos apreciar, los riesgos de mayor relevancia son aquellos relativos a **la recogida de datos y la calidad de los mismos**, constituyendo esta actividad no solo un cuello de botella sino una actividad crítica del proyecto. No obstante, nos encontramos con el compromiso de obtener unos datos de calidad aceptable en un proceso de recogida no supervisada, por lo que todas las medidas del plan de contingencia deberán tener esta idea presente y estar enfocadas a mitigar los riesgos mencionados.

2.3.2. Plan de contingencia

Estableceremos planes de contingencia para todas las amenazas cuyo nivel de riesgo sea superior a apreciable. Se presentan en la tabla 2.4.

ID Riesgo	Salvaguarda	Probabilidad residual	Impacto residual	Nivel de riesgo residual
Tec.2	Trabajar los fines de semana y/o ampliar las 300 horas	Improbable	Desastroso	Bajo
Tec.3	Elaboración de un manual de usuario. Contacto continuo.	Muy improbable	Catastrófico	Bajo
Tec.4	Documentación continua	Improbable	Catastrófico	Apreciable
Infra.1	Desarrollo continuo. Contacto continuo.	Muy probable	Menor	Bajo
Info.1	Búsqueda de nuevos voluntarios. Recogida de datos propia.	Posible	Desastroso	Apreciable
Info.2	Contacto continuo	Posible	Catastrófico	Importante

Tabla 2.4: Plan de contingencia de riesgos

Tras establecer el plan de contingencia con los nuevos niveles de riesgo residual de las amenazas más relevantes, nos damos cuenta de que inevitablemente deberemos asumir un nivel de riesgo importante en cuanto a la recogida de datos se refiere. Esto es algo evidente si tenemos en cuenta que es la única tarea que depende de terceros y, por tanto, sobre la que no tenemos un nivel de control relevante. La calidad de los datos base será determinante en el proceso de minería para el reconocimiento de actividades humanas en particular y, consecuentemente, para el buen término del proyecto en general.

2.4. Seguimiento

Como se ha mencionado, la planificación detallada de las actividades concretas del plan de acción ha sido realizada en reuniones aproximadamente semanales, con ayuda de los tutores del trabajo. El tiempo empleado para cada tarea en particular viene pormenorizado en la tabla 2.5. Con respecto a las previsiones efectuadas en la sección 2.2, hemos infraestimado **la recogida de datos**, la **minería** y la **documentación**; y hemos sobreestimado la **aplicación móvil** y el **análisis y procesamiento de datos**. A su vez, nos hemos excedido unas 8 jornadas en el desarrollo y hemos dispersado el tiempo de exceso, finalizando el día 21 de junio de 2021 en lugar del día 11 de junio de 2021. Las relaciones de dependencia y la calendarización de las actividades viene reflejada en la figura 2.2.

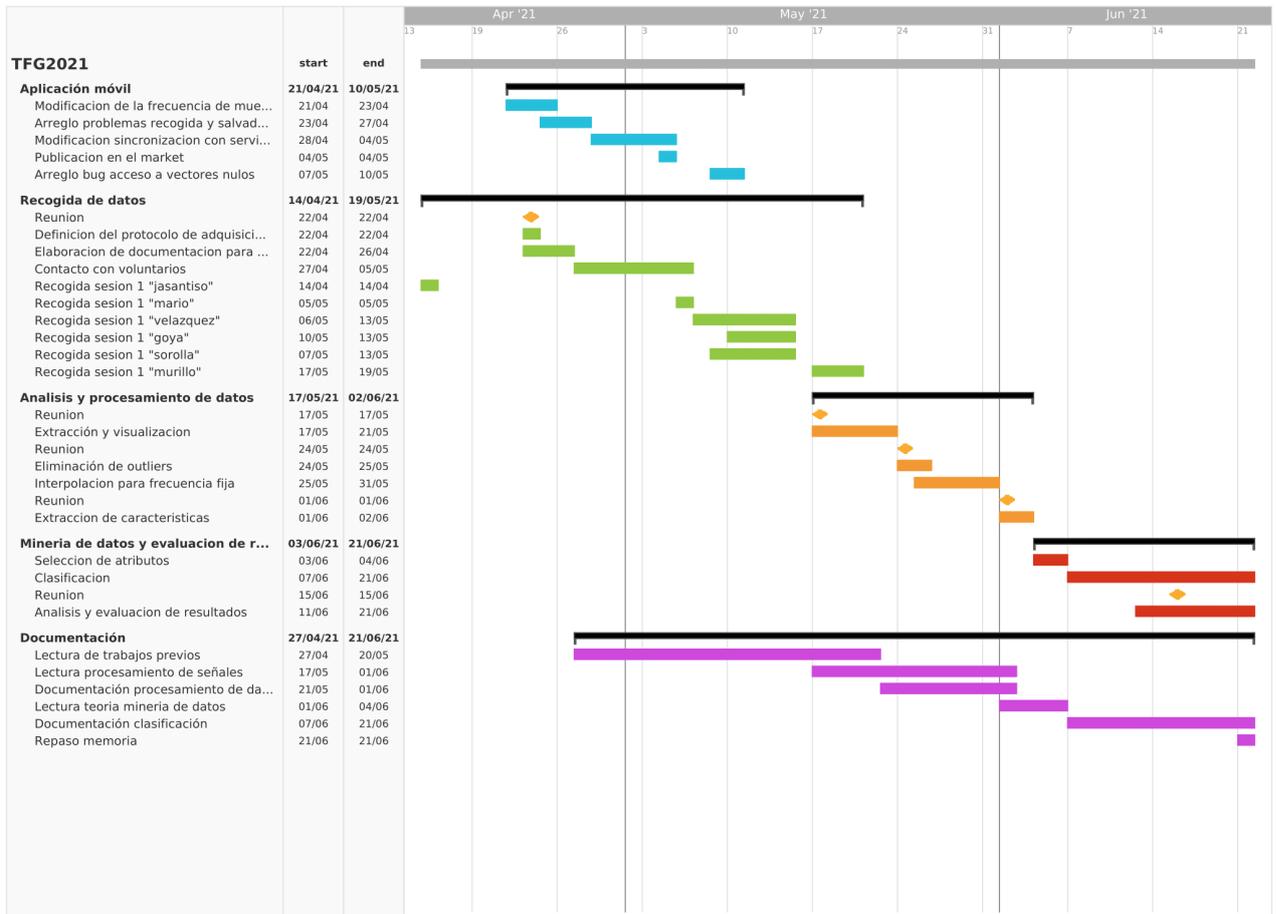


Figura 2.2: Seguimiento de las actividades realizadas durante el desarrollo del trabajo

Actividad principal	Actividad particular	Tiempo empleado	Tiempo total
Aplicación móvil	Modificación de la frecuencia de muestreo	2	6.25
	Arreglo problemas recogida y salvado en memoria	4	
	Modificación sincronización con servidor	0.0625	
	Publicación en el market	0.125	
	Arreglo bug acceso a vectores nulos	0.0625	
Recogida de datos	Definición del protocolo de adquisición	0.25	5.375
	Elaboración de documentación para voluntarios	0.125	
	Contacto con voluntarios	5	
Análisis y procesamiento de datos	Extracción y visualización	2	9
	Eliminación de outliers	1	
	Interpolación para frecuencia fija	5	
	Extracción de características	1	
Minería de datos y evaluación de resultados	Selección de atributos	0.25	15.25
	Clasificación	9	
	Análisis y evaluación de resultados	6	
Documentación	Lectura de trabajos previos	2	9.5
	Lectura procesamiento de señales	2	
	Documentación procesamiento de datos	2	
	Lectura teoría minería de datos	0.5	
	Documentación clasificación	2	
	Repaso memoria	1	
Total			45.375

Tabla 2.5: Aproximación del tiempo empleado en cada actividad particular del proyecto medido en jornadas de trabajo

Capítulo 3

Estado del arte

A pesar de su escaso desarrollo con objeto de la biometría conductual, el uso de modelos de aprendizaje automático para el reconocimiento de actividades humanas es una realidad hoy en día y tiene una gran proyección de futuro. Esta disciplina de la inteligencia artificial tiene innumerables aplicaciones en campos tan dispares como la medicina, la seguridad, el entretenimiento o, tal como nuestro trabajo augura de forma prospectiva, la biometría. Tanto la forma de realizar actividades de una persona como su rutina de actividades cotidianas representan una valiosa información que creemos que se puede explotar con el fin de usarse para una autenticación de personas de alta fiabilidad. Dada la novedad que representa la aplicación planteada, el primer paso a abordar y, por tanto, el que se llevará a cabo en este trabajo, consiste en el reconocimiento de actividades mediante el uso de sensores de tipo comercial, accesibles para el común de la población. Es por ello que nos disponemos a dar unas pinceladas sobre los fundamentos del reconocimiento de actividades humanas como disciplina de la computación y la inteligencia artificial y su estado de desarrollo en la actualidad.

3.1. Introducción al Reconocimiento de Actividades Humanas (HAR)

Para poder dar una definición acertada del problema del reconocimiento de actividades humanas, es necesario primeramente determinar qué es lo que entendemos por actividad humana. Según Chen y Nugent[18], una **acción** es un comportamiento simple ambulatorio. Desde mover ligeramente la mano hasta abrir una puerta o realizar un trazo en un papel, cualquier conducta que responda a esta definición es susceptible de ser considerada acción. Cabe destacar que la caracterización de la acción humana que hemos descrito responde a un criterio fisiológico, no praxeológico, por lo que atiende tanto a una acción deliberada como un acto reflejo (reacción animal)[19], en tanto en cuanto estos sean objetivables a través de alguna medición.

Así, una **actividad** se define como un comportamiento complejo consecuencia de la secuenciación, entrelazada o no, de diversas acciones. Nos damos cuenta de que la distinción entre acción y actividad que estamos manejando no responde a ningún criterio objetivo, sino más bien al nivel de abstracción que deseemos considerar en cada situación particular de estudio. Una actividad puede ser una acción en un nivel de abstracción superior y al revés, dependiendo de la forma de abordar el dominio de interés. Complementariamente, Reyes Ortiz [20] distingue tres niveles de abstracción en las actividades realizadas por un ser humano:

- **Gestos corporales y transiciones.** Consisten en movimientos visibles usados en comunicación no verbal y cambios entre las actividades en realización, respectivamente.
- **Actividades básicas.** Caracterizadas por acciones continuas o cíclicas de tipo estático o dinámico.
- **Actividades complejas.** Consisten en combinaciones de acciones simples (caso multi-actividad) o

combinaciones de acciones de distintos usuario (caso multi-usuario).

Eligiendo el nivel de abstracción más apropiado, seremos capaces de definir el conjunto de actividades óptimo para cada problema.

Con todo, el **reconocimiento de actividades humanas (HAR)** es el proceso por el cual el comportamiento de una persona y su entorno son monitorizados y analizados para inferir la actividad en desarrollo[18]. Formalmente, este proceso se puede caracterizar como problema de la siguiente manera[21]:

Definición 1 Sea $S = \{S_0, \dots, S_{k-1}\}$ un conjunto de k series temporales correspondientes cada una a un atributo de medida y definidas en el intervalo $I = [t_\alpha, t_\omega]$. El **problema del reconocimiento de actividades humanas (HARP)** consiste en encontrar una partición temporal $\langle I_0, \dots, I_{r-1} \rangle$ de I basada en S y un conjunto de etiquetas de actividad para cada I_j de la partición tal que $\bigcup_{j=0}^{r-1} I_j = I$ siendo los intervalos I_j consecutivos, no vacíos y no solapados y las actividades no simultáneas.

Tal y como se ha presentado el HARP en la definición 1, la obtención de la partición temporal apropiada se habría de llevar a cabo a través del análisis combinatorio de todas las particiones temporales posibles de I , lo que constituye un problema no resoluble en tiempo determinista, es decir, NP. Por lo tanto, con vistas a la aplicación práctica de un método resolutivo a partir de la minería de datos, se define la **versión relajada del HARP** como sigue[21]:

Definición 2 Sea $W = \{W_0, \dots, W_{m-1}\}$ un conjunto de m ventanas de igual tamaño total o parcialmente etiquetadas tal que cada ventana W_i contiene un conjunto de series temporales $S_i = \{S_{i,0}, \dots, S_{i,k-1}\}$ correspondiente a k atributos medidos, y un conjunto $A = \{a_0, \dots, a_{n-1}\}$ de etiquetas de actividad. El **problema del reconocimiento de actividades humanas relajado (relaxed HARP)** consiste en obtener una función $f : S_i \rightarrow A$ que pueda ser evaluada para cada S_i tal que $f(S_i)$ sea tan similar como sea posible a la actividad realizada durante W_i .

La relajación introducida en la definición 2 a través del ventaneo de S mediante W permite la caracterización del HARP como un **problema de aprendizaje supervisado de clasificación**, dado que se basa en la inducción de una función que relaciona valores de entrada, las características de cada ventana, con etiquetas de clase, las actividades asociadas. Sin embargo, esta simplificación induce **un error en el modelo** durante las ventanas que almacenan transiciones entre series contiguas que representan dos actividades distintas, debido a que se está estableciendo una única etiqueta de actividad donde realmente debiera haber más de una (figura 3.1). Como consecuencia, la magnitud del error cometido viene determinada por el número de series contenidas en cada ventana. Cuantas menos series abarque una ventana, más se aproximará el concepto de ventana a una serie temporal y menor será el error cometido, pero mayor será la complejidad del modelo. Habremos de tener este compromiso en consideración a la hora de elegir el tamaño de ventana m , dado que en nuestro caso implementaremos una solución para el problema relajado, como viene siendo habitual.

3.2. Aproximaciones al HAR

Como hemos comentado, el reconocimiento de actividades humanas viene siempre precedido por la monitorización del comportamiento personal y del entorno del individuo de estudio con el fin de objetivizar las acciones realizadas y reconocer las actividades en desarrollo. Esto no quiere decir que toda investigación sobre HAR parta desde la creación de un dataset de series temporales de atributos de medida. De hecho, gran parte de los trabajos de este campo emplean fuentes bien conocidas como *UCI Machine Learning Repository*,

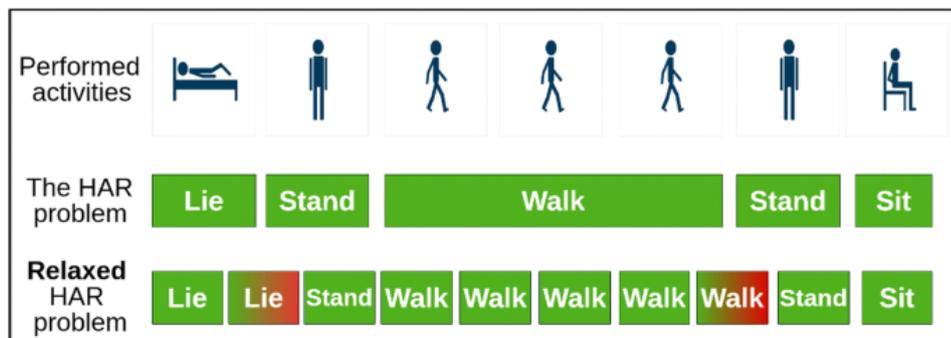


Figura 3.1: Ejemplo del error inducido a través de la relajación del HARP[1]

limitando así su estudio a la generación de un modelo de reconocimiento a partir de datos preexistentes. Sin embargo, en cualquier caso, la forma de abordar el proceso viene fuertemente determinada por el dispositivo de recogida empleado a la hora de la captura de datos durante la realización de las actividades, dado que estos limitan el nivel de detalle de la objetivización de las mismas a través de mediciones. Es por ello que podemos distinguir dos tipos de aproximaciones al reconocimiento de actividades humanas, dependiendo de la posición del dispositivo con respecto al usuario: a través del uso de **dispositivos externos** (fuera del individuo) y a través del uso de **dispositivos ponibles** (portados por el usuario)[21].

3.2.1. HAR mediante dispositivos externos

El uso de dispositivos externos en la recogida de datos suele ser útil cuando esta se desea abordar en un entorno supervisado. El paradigma principal al que responde es el **reconocimiento basado en visión**, por el cual se emplean cámaras para la obtención de la información de partida del proceso de obtención de conocimiento. Algunas de las aplicaciones más interesantes de esta aproximación son la vigilancia, el aprendizaje de robots, la medicina o la seguridad. Por otro lado, los dispositivos externos también pueden ser útiles para el **reconocimiento basado en objetos**, que complementa en gran medida el uso de sensores al permitir la caracterización de actividades complejas según los objetos manipulados durante las mismas[18].

El uso de dispositivos externos es prácticamente una condición *sine qua non* para la monitorización del entorno del usuario, por lo que, en general, si se prescinde de ellos se tiene que asumir la monitorización únicamente del individuo, simplificando el modelo pero perdiendo información contextual que puede ser relevante. Por ejemplo, a priori parece que un sistema HAR tendrá mucho más sencilla la tarea de distinguir cuándo una persona está durmiendo de cuando está despierta si para hacerlo le es posible tener en cuenta la luminosidad de la habitación en la que se encuentra y no solo un registro, por ejemplo, del movimiento de su brazo.

Los principales inconvenientes que presenta el uso de dispositivos externos son, por un lado, los altos costes de mantenimiento e instalación de los dispositivos dedicados y, por otro, la privacidad, permisividad y escalabilidad en el uso de los datos capturados para el reconocimiento de actividades[21].

3.2.2. HAR mediante dispositivos ponibles

El uso de dispositivos ponibles en la recogida de datos viene a solucionar en gran medida los principales problemas derivados del uso de dispositivos externos para el reconocimiento de actividades humanas. Los dispositivos ponibles son baratos, dado su carácter comercial, y no son invasivos, dado que en muchas ocasiones vienen en forma de reloj o pulsera y, por tanto, tienen una utilidad práctica para el usuario en monitorización. Además, permiten un tratamiento relativamente eficiente de la información capturada. Como

contrapartida, la autonomía de los mismos limita su uso en aplicaciones prácticas y se deben complementar a la hora de capturar actividades que empleen movimientos complejos[18].

El uso de ponibles para HAR responde principalmente al paradigma de **reconocimiento basado en sensores**[18], que son esencialmente de tres tipos[21]:

- **Sensores de movimiento.** Típicamente miden la aceleración y el ángulo del movimiento en realización. Son baratos y suelen dar un buen desempeño en el uso para el reconocimiento de actividades que se diferencian según el movimiento de la parte del cuerpo donde se porten. Son principalmente el acelerómetro y el giroscopio. Según diversos trabajos [21, 22, 23], la frecuencia de muestreo de estos sensores influye en el desempeño de los modelos de reconocimiento, siendo en general suficiente una frecuencia de muestreo de 10 a 20 Hz.
- **Sensores de variables de entorno.** Pueden medir tanto atributos de ambiente como la humedad, la intensidad lumínica o la temperatura, como la localización del individuo. En este último caso se debe tener en consideración la privacidad de los usuarios, anonimizando los datos si fuera preciso. La mayor parte de los sensores de variables de entorno no se suelen encontrar en ponibles comerciales, sino en dispositivos externos.
- **Sensores de señales fisiológicas.** La medida más clara de este tipo es el ritmo cardíaco. No suele ser útil para el reconocimiento de actividades humanas por sí solo, dado que, por ejemplo, es posible tener un pulso alto correspondiente a una actividad física intensa incluso estando sentado.

Los ponibles abren el uso de los modelos de reconocimiento de actividades al público general, permitiendo la escalabilidad de los sistemas a múltiples ámbitos como, en nuestro caso de forma prospectiva, la biometría. Además, habilitan la recogida de datos en entornos no supervisados, de forma que permiten una representación muy fidedigna de la forma natural en la que las personas realizan actividades.

3.3. Diseño de un sistema HAR

Antes de comenzar con el diseño de un sistema para el reconocimiento de actividades humanas, es necesario comprender los dos paradigmas que se pueden seguir en la construcción de los mismos[18]:

- **Reconocimiento dirigido por conocimiento** (*knowledge-driven*). Se parte de un conocimiento a priori del dominio para el modelado y representación de las actividades y los datos de sensores, empleando el razonamiento lógico para llevar a cabo el HAR. Es necesario el uso de formalización logico-matemática.
- **Reconocimiento dirigido por datos** (*data-driven*). Se basa en la búsqueda y reconocimiento de patrones a través de la minería de datos y el aprendizaje automático. Emplea un razonamiento probabilístico y estadístico a partir de conjunto de datos capturados. El punto débil de esta aproximación es precisamente el tamaño de necesario de los conjuntos de datos a usar en el proceso, que es determinante junto con la técnica de minería empleada para el desempeño de los modelos de HAR.

En general, cualquier proceso de obtención de datos para el que se parta de conocimiento a priori del dominio tenderá a dar como resultado modelos de mejor rendimiento que si se parte del desconocimiento. No es algo exclusivo de los sistemas HAR, sino, en general, de cualquier aplicación del proceso de minería de datos. En el caso de [22], los autores inducen un modelo híbrido para HAR que incorpora conocimiento

a priori en su estructura y es comparado con algoritmos de clasificación inducidos de forma automática. El resultado es que el modelo híbrido HAR brinda una tasa de acierto en el reconocimiento notablemente superior a la del resto de clasificadores en general.

A pesar de esto, dado que nuestro trabajo es de carácter prospectivo, llevaremos a cabo un diseño dirigido por datos sin uso de conocimiento adicional. Implementaremos el proceso típico de minería de datos descrito previamente en la sección 2.2.

3.4. Factores determinantes en la minería de datos para HAR

3.4.1. Protocolo de recogida

Como hemos anticipado en la sección 3.2, no existe una forma única de recoger datos cuando de HAR se trata. En la medida en la que nuestros dispositivos lo permitan, la obtención de datos para el reconocimiento de actividades puede llevarse a cabo en un **entorno controlado** o en un **entorno natural**[21]. La diferencia radica en el protocolo a seguir por los participantes en la investigación, que viene a su vez determinado por los objetivos finales de la misma. Si se desea obtener un modelo de reconocimiento muy certero para unas actividades muy concretas a partir de unos datos de mucha calidad, quizá sea mejor optar por una aproximación supervisada por los investigadores. Por el contrario, si lo que se desea obtener es un modelo fiel al comportamiento real y cotidiano de los individuos con vistas a una aplicación práctica, sin duda la aproximación no supervisada es del todo adecuada.

Históricamente, la elección de la supervisión o no en la recogida de datos de las investigaciones de reconocimiento de actividades humanas a través de aprendizaje automático ha sido determinante a la hora de limitar el desempeño de los clasificadores, dado que suelen establecer una cota superior a la tasa de acierto difícil de superar. Es el caso de [22], donde la tasa de acierto total de los clasificadores disminuye de un 90 % a un 72 % únicamente por dar el paso de una recogida controlada a una sin supervisión. Dado que nuestro enfoque es la futura aplicación al reconocimiento biométrico en entornos realistas, pretendemos recoger información fidedigna de los usuarios en su día a día. Adoptaremos así como aproximación una recogida en entorno natural, sin control, asumiendo el riesgo sobre la calidad de los datos inherente a la misma.

Por otra parte, en el protocolo de recogida también se determina la **posición de los sensores**, en nuestro caso ponibles, con respecto al usuario en la monitorización de actividades. El uso de uno o varios sensores y las posiciones de los mismos influyen, si bien no tan determinadamente como la supervisión, de forma notable en el desempeño de los modelos de clasificación. En [23], los investigadores comprueban el impacto de la posición de los sensores en la tasa de acierto y el $f1$ de los clasificadores, concluyendo que la posición óptima de los mismos consiste en una combinación de varios sensores distribuidos en la parte superior e inferior del cuerpo. En este estudio, una combinación subóptima en la posición de los sensores lleva a una disminución de hasta un 10 % en la tasa de acierto en la peor de las combinaciones posibles. Puesto que nuestra investigación tiene como fin una prospección para una futura aplicación práctica en el terreno de la biometría, y dado que para ello empleamos solo sensores ponibles de uso comercial, relojes inteligentes, nuestro margen de actuación se ve limitado a escoger una de las dos muñecas en las que llevar el reloj.

3.4.2. Definición de actividades

La elección del conjunto de actividades a considerar está íntimamente relacionado con el objetivo final de cualquier modelo de reconocimiento de actividades humanas. Un sistema HAR enfocado a la medicina no suele definir las mismas actividades que uno dedicado al entretenimiento y los videojuegos. Decidir tanto el

número de actividades como el nivel de abstracción de las mismas es fundamental, porque una mala elección puede conllevar la imposibilidad del reconocimiento.

En [24], Paraschiakos construye una ontología de actividades cotidianas con tres niveles de abstracción y comprueba su efecto en el desempeño de la clasificación final mediante la minería de datos recogidos de ponibles. Llega a la conclusión de que el nivel óptimo de granularidad es un conjunto de 7 actividades de nivel de descripción medio, de las que hemos categorizado en la sección 3.1 como básicas. Las actividades que él determina óptimas según el nivel de intensidad física que representan son las siguientes:

- **Actividades de baja intensidad:** *estar acostado, estar sentado.*
- **Actividades de media intensidad:** *estar de pie, tareas del hogar.*
- **Actividades de alta intensidad:** *saltar, caminar y montar en bicicleta.*

Por otro lado, en [25], los autores obtienen los mejores rendimientos de clasificación para los dos datasets de ponibles, *UCI-HAD* y *USC-HAD*, con representación de actividades que comportan movimientos distinguibles entre sí. Nos resultan de especial interés las actividades contenidas en el primero, a saber: *caminar, subir escaleras, bajar escaleras, estar sentado, estar de pie y estar acostado*. En gran medida, este conjunto coincide con las actividades que concluye Paraschiakos como óptimas.

Las actividades que hemos definido para nuestro proyecto también se pueden considerar como un subconjunto de las de Paraschiakos, dado que las que no coinciden son asimilables a otras de intensidad similar. La elección es apropiada para el posterior reconocimiento biométrico de personas debido a que las actividades englobadas son realizables por la mayor parte de la población y su nivel de detalle es suficiente para el HAR.

Capítulo 4

Recogida de datos

Como hemos analizado tras la gestión de riesgos (sección 2.3), la recogida de datos representa una fase crucial del proyecto, no solo por constituir el punto de partida imprescindible para nuestro propósito final, sino también por las implicaciones que supone la obtención de una nueva base de datos conductual para investigaciones futuras. Anteriormente ya se había llevado a cabo la recogida de una base de datos dedicada al reconocimiento biométrico mediante ponibles[12]. Sin embargo, para nuestro trabajo no ha sido conveniente el uso de estos datos debido, por un lado, a que la recogida de datos se efectuó de forma supervisada, y por otro, a que las mediciones de los sensores responden únicamente a la actividad de *caminar*. Por lo tanto, nos hemos visto en la obligación de solicitar la ayuda de voluntarios para confeccionar una nueva base de datos con mediciones de los sensores de los ponibles correspondientes a un conjunto más amplio de actividades.

4.1. Protocolo de recogida

Cualquier voluntario cuyos datos estén presentes en la base de datos habrá llevado la recogida de forma autónoma y **no supervisada**. La recogida de datos se ha efectuado a través de una aplicación móvil desarrollada durante el periodo de prácticas en empresa, de tal forma que cada usuario participante ha sido libre de utilizar un reloj inteligente que le hemos facilitado o el suyo propio para la captura de datos. De esta manera, los datos recogidos representan la actividad realizada de una forma más fidedigna y menos impostada que en una recogida condicionada a la observación.

Se ha decidido llevar a cabo **tres sesiones de recogida**, de tal manera que entre la primera y la segunda haya unas tres semanas de diferencia; y entre la segunda y la tercera unos tres meses de diferencia. A su vez, en la semana primera y tercera el dispositivo ponible se llevará en la muñeca de la **mano izquierda**, mientras que en la segunda el dispositivo se cambiará de muñeca a la de la **mano derecha**. De esta forma, se podrá estudiar la influencia tanto de la posición del ponible como del paso del tiempo en la conducta durante la realización de actividades.

A cada participante se le ha hecho entrega de un documento explicativo que incluye las líneas generales de la actuación durante una recogida. Además, se ha puesto a disposición de todos un teléfono de contacto para cualquier duda tanto en el uso de la aplicación como en la forma de actuar. Sin embargo, cabe destacar que el carácter autónomo en la obtención de muestras obliga a que la aplicación de este protocolo no sea de carácter estricto. Las indicaciones dadas son las siguientes:

- Se deberá anotar cualquier incidencia durante la recogida de alguna muestra.
- Se deberán tomar muestras tanto de la realización de actividades aisladas como de la realización de

actividades encadenadas. El usuario debe indicar en la aplicación la actividad inicial antes de comenzar la recogida de cualquier muestra y, en caso de tratarse de una muestra de actividades encadenadas, el resto de las actividades según vayan transcurriendo durante la misma.

- Las actividades sedentarias se adquirirán durante un tiempo máximo de 3 minutos o hasta que se haya realizado todo el abanico de movimientos significativos de la actividad en cuestión.
- Las actividades dinámicas se adquirirán durante un tiempo no inferior a 5 minutos.
- Se estima el tiempo total de recogida de entre 1 a 2 horas, con la mayor variedad posible de actividades.

4.2. Datos recogidos

Dada la limitación temporal a la que nos enfrentamos en el desarrollo del trabajo de fin de grado, nos hemos visto obligados a emplear solo los datos capturados durante la **primera sesión** de recogida. No obstante, se mantiene la pretensión de completar la base de datos conductual.

Para poder obtener las muestras conductuales, hemos contactado con voluntarios que han accedido a participar, consintiendo explícitamente a través de su firma las siguientes condiciones:

- Los usuarios participantes se identifican en la base de datos a través de identificadores anonimizados. No se almacena ningún tipo de información personal en la misma.
- La correspondencia entre la identidad real de los participantes y su identificación en la base de datos se mantendrá en una tabla temporal a efectos prácticos para poder contactar con los participantes y realizar las tres capturas en el tiempo, pero será destruida al finalizar el proceso.
- Los usuarios participantes aceptan la monitorización de los datos requeridos durante cada recogida de muestras, con las garantías previstas por la LOPDGDD (Ley Orgánica de Protección de Datos y Garantía de Derechos Digitales).

Durante la sesión 1 han sido recogidas un total de **56 muestras** de **6 usuarios distintos**: *goya*, *jasantiso*, *mario*, *murillo*, *sorolla* y *velazquez*. Las recogidas han hecho uso de hasta 6 sensores tanto del ponible como del teléfono móvil:

- **Sensores del dispositivo ponible:** acelerómetro, giroscopio, PPG (fotopletismografía, bajo disponibilidad) y sensor cardíaco (bajo disponibilidad).
- **Sensores del teléfono móvil:** GPS y dirección IP pública.

El número de mediciones capturadas por cada sensor de cada usuario son las indicadas en la tabla 4.1. Según podemos apreciar en figura 4.1, los tres sensores de mayor captura de mediciones son el *acelerómetro*, el *giroscopio* y el sensor de *PPG*. Por otro lado, según la figura 4.2, las mediciones capturadas por usuario están más uniformemente distribuida, siendo los usuarios de mayores capturas *velazquez*, *sorolla* y *mario*.

Los datos capturados corresponden a la realización de seis actividades humanas: *caminar*, *correr*, subir y bajar *escaleras*, *trabajar*, *comer* y *conducir*. El *fallo* durante la recogida se captura como una actividad, mas no está presente en todos los sensores sino tan solo en los del teléfono móvil. Esto se debe a que los fallos corresponden a fallos de conexión entre el teléfono y el reloj, por lo que el teléfono continua capturando datos hasta que se finaliza la recogida. Las mediciones según actividad están distribuidas como indica la figura 4.3.

Las tres actividades para las que se dispone de un mayor número de mediciones son *conducir*, *caminar* y *trabajar*.

Durante la sesión 1, se han empleado **4 relojes inteligentes** Wear OS para la recogida de datos. Son los siguientes:

- *Fossil Carlyle HR 5th Gen*. Es el dispositivo ponible más empleado por ser el dispositivo de recogida que hemos facilitado a los voluntarios. Ha sido empleado por: *goya*, *mario* y *velazquez*.
- *TAG Heuer B0DF*. Ha sido empleado por *murillo*.
- *LG G Watch*. Ha sido empleado por *sorolla*.
- *Dispositivo desconocido*. Ha sido empleado por *jasantiso*.

Usuario	goya	jasantiso	mario	murillo	sorolla	velazquez	Total
Sensor							
Acelerómetro	15676	11917	20756	14343	47386	35933	146011
Giroscopio	14891	1012	20702	14373	47852	33427	132257
PPG	12533	0	17841	13299	0	27521	71194
Cardiaco	893	1084	1	902	0	2017	4897
GPS	8	7	10	10	35	28	98
IP	29	6	26	20	49	72	202
Total	44030	14026	59336	42947	95322	98998	354659

Tabla 4.1: Número de mediciones capturadas por sensor y usuario en la sesión 1

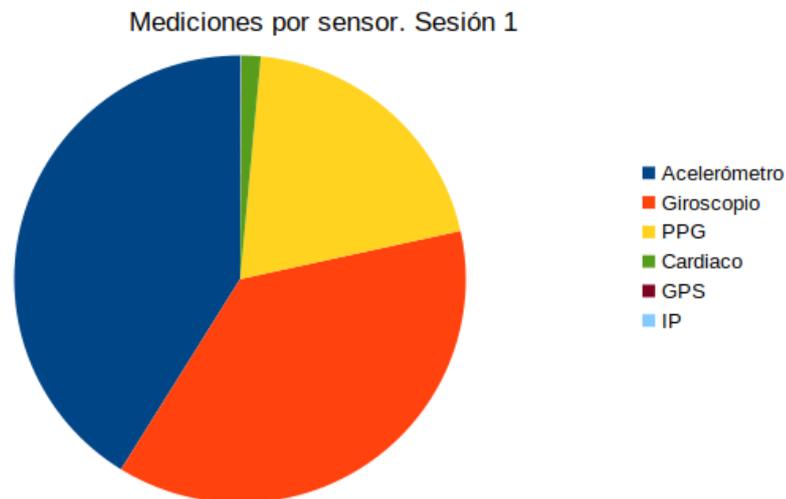


Figura 4.1: Distribución de las mediciones capturadas por sensor en la sesión 1

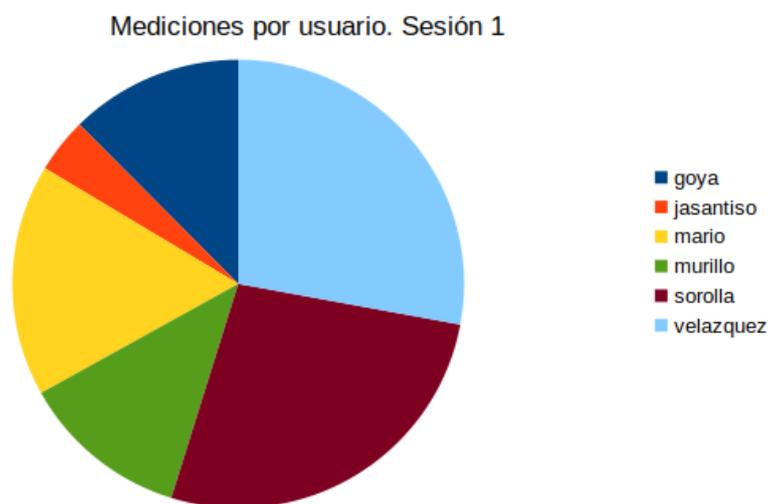


Figura 4.2: Distribución de las mediciones capturadas por usuario en la sesión 1

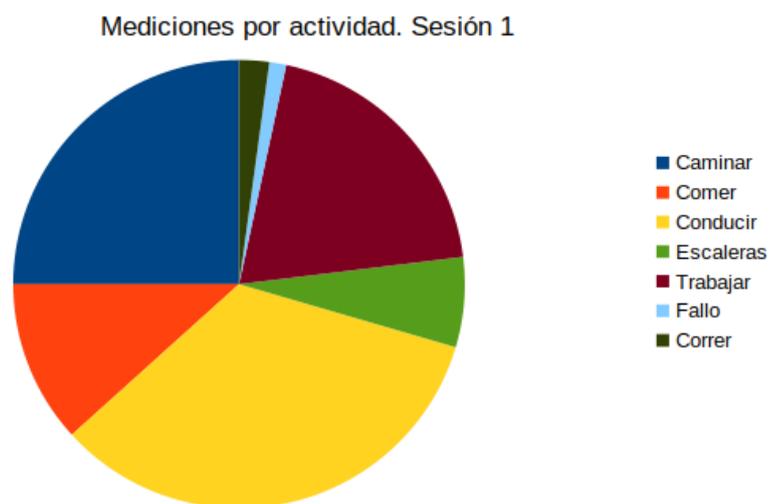


Figura 4.3: Distribución de las mediciones capturadas por actividad en la sesión 1

Capítulo 5

Análisis y procesamiento de datos

El paso previo a cualquier proceso de minería de datos y aprendizaje automático consiste en un tratamiento de los datos capturados con el objetivo de garantizar el buen desempeño de los clasificadores en la fase final de la misma. Los datos ante los que nos encontramos corresponden a series temporales, en concreto a señales digitales, por lo que emplearemos las técnicas más apropiadas de esta disciplina matemática.

5.1. Inspección visual de datos

Antes de comenzar con el procesamiento de las capturas, es necesario representar los datos de los sensores de interés para poder realizar un primer análisis visual de las señales y determinar su calidad a priori. Con el fin de limitar el alcance del proyecto dado el carácter prospectivo y, sobre todo, en base al desarrollo de los trabajos previos[13, 11, 12], hemos decidido que nuestros principales sensores de interés son tanto el **acelerómetro** como el **giroscopio**, por lo que hemos representado la aceleración registrada por el acelerómetro en las tres componentes X, Y y Z, medida en m/s^2 ; y la velocidad angular capturada por el giroscopio en las mismas componentes, medida en rad/s [26], ambos con respecto al tiempo, medido en ms. Analizaremos los aspectos de interés para el procesamiento sobre un ejemplo de señal cruda de la base de datos.

En una primera inspección de la figura 5.1, nos damos cuenta de que nos encontramos ante una **señal aperiódica** con registro de tres actividades: *correr*, *caminar* y *escaleras*. El orden de aparición de las actividades en la leyenda de la figura corresponde al orden de realización de las mismas durante la recogida. El valor de frecuencia indicado en el título de las primeras gráficas corresponde a la mediana de las frecuencias de muestreo del sensor correspondiente, de en torno a los 24 Hz. No obstante, este valor no es constante a lo largo de la señal. La variabilidad de la frecuencia de muestreo se debe, por un lado, a que la aplicación de recogida realiza un muestreo aleatorio de los datos capturados dado el ingente volumen de los mismos y, por otro, que los sensores no realizan un muestreo perfectamente periódico. Es algo evidente, dado que, como podemos observar más pormenorizadamente en un análisis de la componente X de los sensores en la figura 5.2, existen lapsos de tiempo en los que la señal es completamente **plana debido al nulo muestreo de los sensores**. Cabe diferenciar este tipo de zonas planas de aquellas en las que, a pesar de que el muestreo sigue ocurriendo, los valores de amplitud para cada componente son prácticamente constantes, como vemos en la figura 5.3. Este otro tipo de **zonas planas, de amplitud constante**, son de absoluto interés, y se caracterizan por no verse reflejadas de forma casi simétrica entre ambos sensores, acelerómetro y giroscopio, correspondiendo a actividades de poco movimiento del brazo, como es el caso de *trabajar* en la señal de ejemplo. En consecuencia, una de las tareas que deberemos llevar a cabo en el preprocesamiento de datos será la identificación de las **regiones de nulo muestreo** y su eliminación para la obtención de una señal limpia.

Usuario: velazquez, Sesión: 14

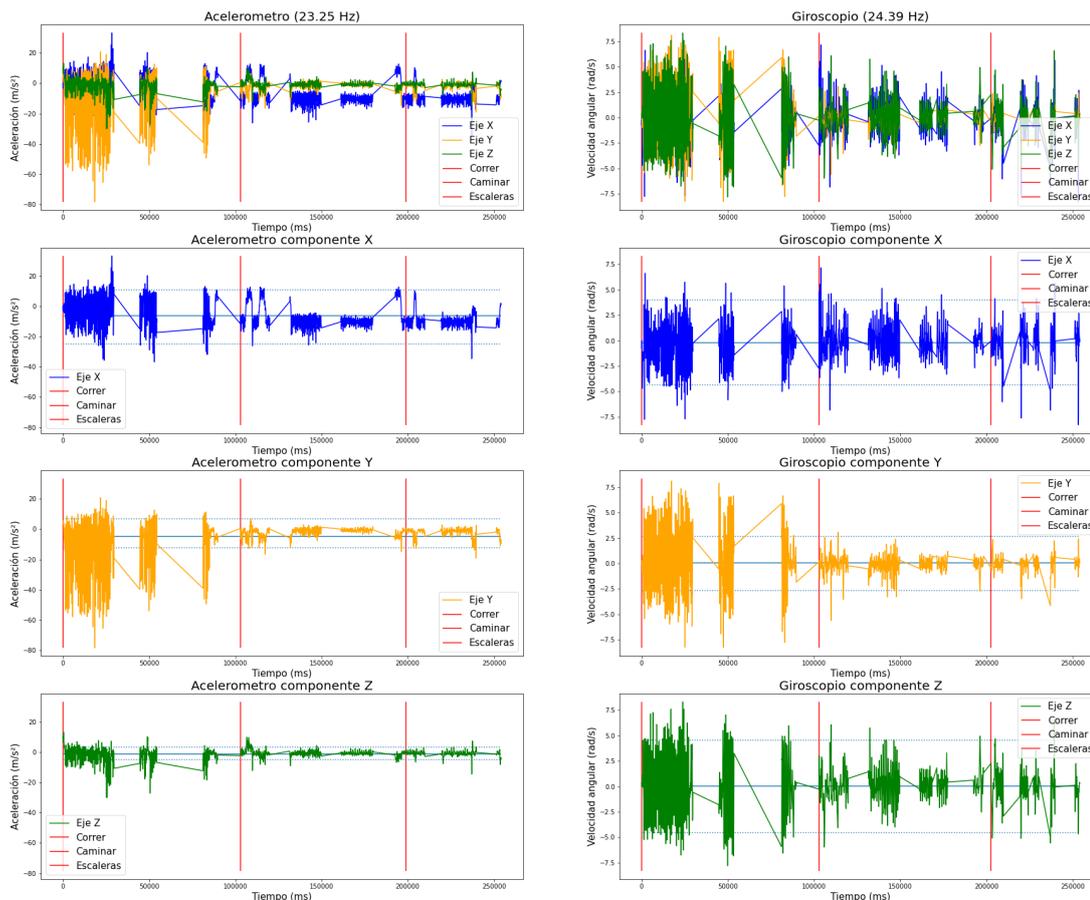


Figura 5.1: Ejemplo de señal cruda correspondiente a la muestra 14 de la sesión 1 del usuario *velazquez*

Por otro lado, si consideramos como ruido de la amplitud a los valores fuera del intervalo delimitado a partir de las expresiones 5.1 y 5.2, gran parte de la señal se encuentra fuera de los márgenes de outlying (líneas horizontales punteadas en las figuras 5.1, 5.2 y 5.3), por lo que a priori no podemos considerarlos como valores atípicos consecuencia de ruido asistémico (ruido del entorno) en las mediciones de los sensores, sino puntos de interés a la hora del reconocimiento y búsqueda de patrones, dado que existe la posibilidad de que correspondan a un movimiento brusco correspondiente a la actividad en desarrollo.

Como conclusión, a través de una inspección visual de las señales capturadas podemos determinar que la frecuencia de muestreo no es constante y que existen regiones sin muestreo de la señal que deberemos identificar y eliminar. Por otro lado, deberemos asumir que el posible ruido asistémico presente en la amplitud de las señales no es identificable a priori dada la naturaleza del dominio del problema.

5.2. Preprocesamiento

5.2.1. Eliminación de regiones de nulo muestreo

El proceso de identificación de las regiones de nulo muestreo en las señales recogidas está íntimamente ligado a la distribución de la diferencia temporal entre una medición de la señal y la siguiente a lo largo de toda

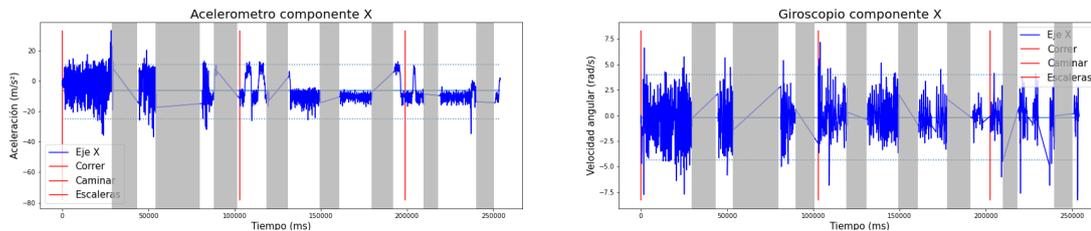


Figura 5.2: Ejemplo de zonas sin muestreo en señal cruda correspondiente a la componente X de la muestra 14 de la sesión 1 del usuario *velazquez*

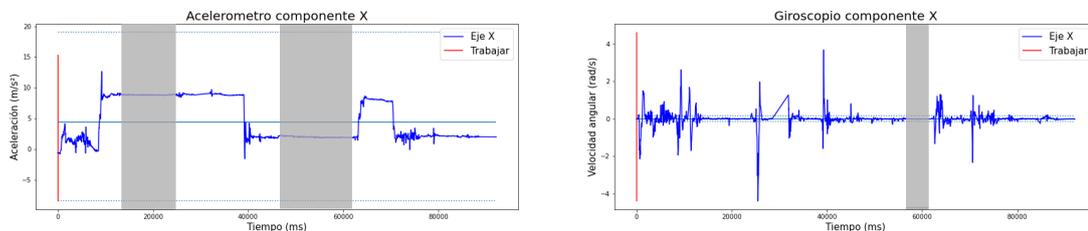


Figura 5.3: Ejemplo de zonas de amplitud constante en señal cruda correspondiente a la componente X de la muestra 9 de la sesión 1 del usuario *sorolla*

una señal, o sea, el **período de muestreo**. En primer lugar, obtenemos todos los períodos de muestreo de todas las señales disponibles para tratarlos como un conjunto único, sin agrupaciones por dispositivo posible de captura. La decisión se debe a que si bien no todas las muestras han sido tomadas por el mismo reloj, todas han sido procesadas por la misma aplicación wearable, con un período de muestreo de mediciones fijo en el dispositivo posible. Por tanto, las regiones planas de muestreo corresponden a ruido en el tratamiento de los datos realizado por la aplicación móvil a la que se conecta el reloj, por lo que tiene sentido la búsqueda de valores atípicos considerando todas las muestras disponibles.

Por el teorema del límite central, asumiremos la distribución normal de dichos valores. Así, podemos establecer los márgenes superior e inferior de valores atípicos mediante las siguientes expresiones[27]:

$$\text{Margen superior} = Q3 + 1,5 \cdot IQR \quad (5.1)$$

$$\text{Margen inferior} = Q1 - 1,5 \cdot IQR \quad (5.2)$$

Donde $Q1$ y $Q3$ son los valores correspondientes al primer y tercer cuartil del período de muestreo, respectivamente, e IQR es el rango intercuartílico de las observaciones.

Al realizar el cálculo de los márgenes superior e inferior de valores atípicos de periodo de muestreo e ignorar aquellos que no se encuentran dentro del conjunto de valores válido [Margen superior, Margen inferior], obtenemos las distribuciones de los periodos de muestreo representadas en la figura 5.4. La diferencia en la dispersión de los periodos de muestreo es muy notable: las mediciones pasan de una diferencia máxima de más de 32 horas a tan solo **168 milisegundos**. Determinamos que esta será la máxima diferencia entre una medición y la siguiente, de tal manera que todas las mediciones cuya diferencia temporal a la anterior sea superior a este límite serán borradas del conjunto de datos.

En el caso de la señal analizada en las figuras 5.1 y 5.2, la eliminación de las zonas planas sin muestreo da lugar a la señal limpia representada en las figuras 5.5 y 5.6. La mediana de las frecuencias de muestreo ha aumentado ligeramente en algo menos de 1 Hz hasta más de los 24 Hz en ambos sensores del ejemplo, llegando hasta los 25Hz en la mayoría de las señales del dataset completo. Esta variación nos permitirá incrementar la frecuencia de muestreo en estudio, con respecto al trabajo previo [13]. Por otro lado, las transiciones en las regiones eliminadas se encuentran dentro de los límites de outlying (expresiones 5.1 y 5.2) de la amplitud de las componentes, por lo que no representan un cambio demasiado brusco con respecto a la totalidad de

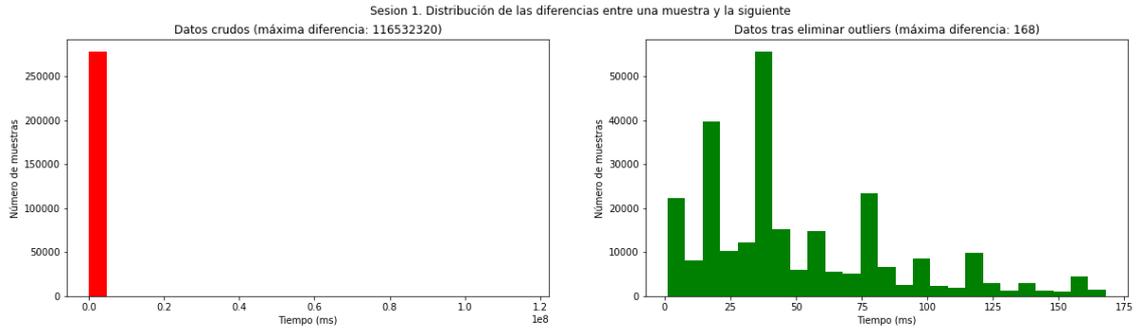


Figura 5.4: Distribución de los períodos de muestreo en ms antes y después de limpiar las señales de valores atípicos

la señal en ningún caso.

Usuario: velazquez, Sesión: 14

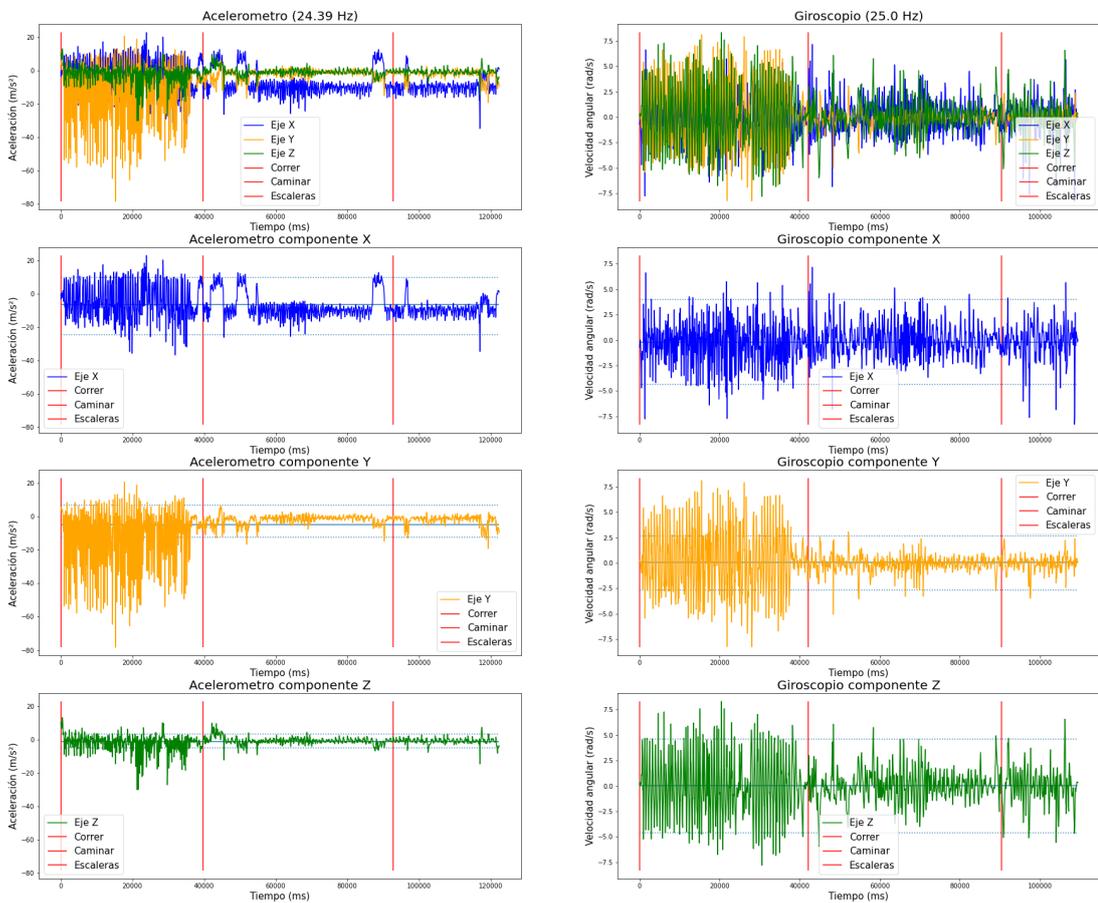


Figura 5.5: Ejemplo de señal limpia de outliers correspondiente a la muestra 14 de la sesión 1 del usuario *velazquez*

5.2.2. Interpolación a frecuencia de muestreo fija

Como venimos comentando desde el análisis de la señal de la figura 5.2, la frecuencia de muestreo de las señales no es constante. Con el fin de garantizar una frecuencia de muestreo fija a la hora de dividir las señales

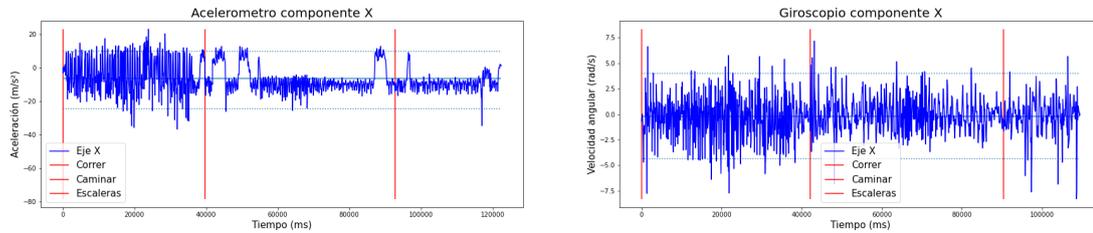


Figura 5.6: Ejemplo de zonas sin muestreo en señal limpia correspondiente a la componente X de la muestra 14 de la sesión 1 del usuario *velazquez*

en ventanas de igual tamaño, dada nuestra aproximación al HARP relajado (definición 2), interpolaremos la señal capturada a un período fijo desde el inicio de la señal original hasta el final de la misma para obtener una nueva señal interpolada de mediciones equiespaciadas. Además, este paso es estrictamente necesario en términos de señales digitales debido a que posteriormente nos interesará obtener las características de la señal en el dominio de la frecuencia a través de la aplicación de la transformada discreta de Fourier, y esta requiere de entrada una señal del tiempo de muestras equiespaciadas[28].

En el trabajo de fin de grado previo[13], se ha comprobado que una frecuencia de muestreo de 12 Hz en los sensores de movimiento es suficiente para el reconocimiento biométrico de personas a través de la forma de caminar, dado que la oscilación del brazo no es superior a 8.6 Hz. Por otro lado, los trabajos relativos a reconocimiento de actividades humanas[22, 21] llegan a la conclusión de que 20 Hz es una frecuencia suficiente para el reconocimiento de actividades humanas mediante el uso de los sensores previamente mencionados. En nuestro caso, comprobaremos el efecto de la frecuencia de muestreo en el desempeño de los clasificadores finales estableciendo **dos frecuencias de estudio: 12 Hz y 24 Hz**. Llevaremos a cabo una **interpolación lineal** cada $T = \lfloor \frac{1000}{f} \rfloor$ milisegundos, donde f es la frecuencia de muestreo que se pretende fijar. Así, los periodos de interpolación que emplearemos para fijar 12 y 24 Hz serán, respectivamente, 83 y 41 ms. El hecho de llevar a cabo una interpolación lineal y no de otro tipo se debe a que es suficiente para el problema a abordar y es relativamente más sencilla que el resto de posibilidades[13].

La señal interpolada obtenida a partir de la señal original es una función discreta del tiempo en tres componentes x , y y z , donde cada instante de interpolación es de la forma $t = k \cdot T$, $k \in \{0, 1, 2, \dots, \lfloor \frac{t_{max}}{T} \rfloor\}$ siendo t_{max} la duración de la señal original, en milisegundos. El valor de la amplitud para cada componente c en cada instante del tiempo de interpolación t se obtiene mediante la siguiente función:

$$c(t) = c_{prev} + \frac{c_{post} - c_{prev}}{t_{post} - t_{prev}} \cdot (t - t_{prev}) \quad (5.3)$$

Donde

- t_{prev} es el instante de muestreo de la señal original inmediatamente anterior a t .
- t_{post} es el instante de muestreo de la señal original inmediatamente posterior a t .
- c_{prev} es el valor de la aceleración de la componente de interés en la señal original en el instante t_{prev} .
- c_{post} es el valor de la aceleración de la componente de interés en la señal original en el instante t_{post} .

A través de la interpolación lineal, la forma de la señal no varía con respecto a la señal limpia de regiones de nulo muestreo.

5.3. Extracción de características

Una vez establecida la frecuencia de muestreo fija para las señales limpias de periodos de muestreo marginales, deberemos extraer las características representativas de las mismas. A esta operación la precede la **división de las señales en ventanas**. El proceso consiste en agrupar mediciones contiguas según un **tamaño de ventana** fijo, establecido como parámetro del procesamiento. A su vez, las ventanas adyacentes se solaparán en un **porcentaje de solapamiento**, también parámetro del procesamiento. De esta forma, dos ventanas contiguas compartirán una fracción de las mediciones contenidas. Variando estos dos parámetros, seremos capaces de controlar el error intrínseco al modelo presente en las ventanas de transición, descrito en la sección 3.1.

Al tratarse de señales que corresponden a la realización de actividades tanto periódicas como aperiódicas, no disponemos de un tiempo de referencia para establecer un tamaño de ventana en función del período de la señal, como en el caso del reconocimiento biométrico a través de la forma de caminar[13], donde el movimiento del brazo oscila en un movimiento aproximadamente periódico. A falta de una mejor referencia, estableceremos el tamaño de ventana mínimo en función del periodo aproximado del movimiento del brazo caminando, o sea, 1 segundo[29], de tal forma que la señal de 12 Hz se dividirá en ventanas de 12 muestras y la señal de 24 Hz en ventanas de 24. A partir de ahí, iremos probando con otros **tamaños de ventana mayores**: de **2 segundos** (24 y 48 muestras), **4 segundos** (48 y 96 muestras) y **8 segundos** (96 y 192 muestras) para analizar la dependencia del rendimiento del sistema con respecto a este parámetro. El **porcentaje de solapamiento** elegido es fijo, del **50 %**, dado el buen resultado del mismo en HAR[23] y ser el habitual de la bibliografía.

Para gestionar las ventanas de **transición entre actividades** de una forma apropiada, se ha tomado la decisión de asignar a cada ventana la **moda de las actividades** de las mediciones que abarca como actividad de clasificación.

Tras ventanear las señales, comenzamos con la extracción de las características de cada ventana. Una señal temporal tiene dos dominios de análisis: el dominio del tiempo y el dominio de la frecuencia. Para poder obtener las características de las señales en el dominio de la frecuencia, aplicaremos la **transformada discreta de Fourier** (DFT) mediante el algoritmo de la transformada rápida de Fourier (FFT). La DFT es una función que toma señales en el dominio del tiempo y las transforma al dominio de la frecuencia. La DFT de una señal discreta $x[n]$ definida en el rango $0 \leq n \leq N - 1$ se define como[30]:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j \frac{2\pi kn}{N}} \quad (5.4)$$

Donde $X[k]$ son los valores del análisis espectral de $x[n]$. Esta es una expresión compleja, por lo que los valores $X[k]$ tienen un módulo y un argumento. En la extracción de características prescindiremos del argumento y trabajaremos tan solo con el **módulo de las amplitudes** de la señal $X[k]$.

La FFT es el estándar de facto para el cálculo de la DFT debido a que es un algoritmo extremadamente eficiente, con una complejidad $O(n \cdot \log n)$. Emplearemos su implementación en el módulo `fft` de la librería `scipy` para Python[31]. Obtendremos así, de cada componente X,Y,Z en cada medición de cada ventana, el módulo de la amplitud de la transformada de Fourier para cada frecuencia.

Para la extracción de características, emplearemos las funciones incorporadas en la librería `pandas`, `numpy` y el módulo `stats` de `scipy`. De esta manera, las características extraídas para cada componente X,Y,Z por ventana y sensor son las siguientes:

■ Dominio del tiempo

- *Máximo.*
- *Mínimo.*
- *Rango.*
- *Media.*
- *Desviación estándar.*
- *Mediana.*
- *Primer cuartil.*
- *Tercer cuartil.*
- *Kurtosis.*
- *Skewness.* Computada como el coeficiente de asimetría de Fisher-Pearson[32].

■ Dominio de la frecuencia

- *Frecuencia dominante.*
- *Amplitud de la frecuencia dominante.*
- *Segunda frecuencia dominante.*
- *Amplitud de la segunda frecuencia dominante.*
- *Amplitud máxima.*
- *Amplitud mínima.*
- *Rango de la amplitud.*
- *Amplitud media.*
- *Desviación estándar de la amplitud.*
- *Amplitud mediana.*
- *Primer cuartil de la amplitud.*
- *Tercer cuartil de la amplitud.*
- *Kurtosis de la amplitud.*
- *Skewness de la amplitud.* Computada como el coeficiente de asimetría de Fisher-Pearson[32].

5.4. Procesamiento ad hoc

El procesamiento de datos comentado en la sección 5.2 es aplicable en general a toda señal del conjunto de datos y es el punto de partida para el proceso de minería y extracción de conocimiento. Sin embargo, dependiendo del clasificador que deseemos inducir, los datos requerirán algún otro tratamiento específico a mayores para garantizar el buen desempeño de los mismos.

5.4.1. Selección y transformación de atributos

La cantidad de características extraídas para las señales tanto en el dominio del tiempo como en el de la frecuencia hacen un total de **72 atributos** para cada instancia del dataset. Como consecuencia, es posible que se dé una de estas tres situaciones problemáticas a la hora de abordar el proceso de aprendizaje

automático[33]: la existencia de **atributos irrelevantes**, la existencia de **atributos redundantes** y la **mal-dicción de la dimensionalidad**. Cualquiera de estas tres circunstancias implica el riesgo de la disminución en el rendimiento de los clasificadores, a la par que un aumento del tiempo de cómputo necesario para su inducción.

Con el fin de evitar esto, llevaremos a cabo los siguientes procesamientos previos:

- **Selección de atributos.** El objetivo de la selección de atributos es obtener un subconjunto de los atributos originales evitando los atributos irrelevantes y redundantes. Aplicaremos las siguientes técnicas:

- Métodos de filtro
 - Selección basada en **correlación**.

A través del cálculo de los coeficientes de correlación de Pearson entre los valores de los atributos mediante la implementación de la librería `pandas`, obtenemos una matriz de correlación de los mismos, representada como un mapa de calor en las figuras 5.7 y 5.8. A priori podemos afirmar que los atributos con menor correlación entre sí brindarán mejor desempeño del clasificador, dado que serán los atributos con menor redundancia. En un análisis visual, nos damos cuenta de que los atributos de menor correlación con el resto parecen ser la *kurtosis* y *skewness* de las componentes en el dominio del tiempo, y las *frecuencias dominantes* en el dominio de la frecuencia. Consideramos así que los atributos menos correlacionados son aquellos cuyo coeficiente de Pearson se encuentre en el intervalo $(-0,25, 0,25)$. Obtenemos las selecciones de atributos indicadas en la tabla 5.1.

Sensor	Frecuencia interpolación	Tamaño ventana	Selección atributos	Total
Acelerómetro	12 Hz	1s	['maxX', 'maxY', 'kurtosisX', 'kurtosisY', 'skewX', 'skewY', 'skewZ', 'freqDom1X', 'freqDom1Y', 'freqDom1Z', 'freqDom2X', 'freqDom2Y', 'freqDom2Z']	12
		2s	['maxX', 'maxY', 'skewX', 'skewY', 'skewZ', 'freqDom1X', 'freqDom1Y', 'freqDom2X', 'freqDom2Y', 'freqDom2Z']	10
		4s	['maxX', 'skewX', 'skewY', 'skewZ', 'freqDom1X', 'freqDom1Y', 'freqDom1Z', 'freqDom2Z']	8
		8s	['maxX', 'skewX', 'skewY', 'freqDom1Y', 'freqDom1Z']	5
	24 Hz	1s	['maxX', 'maxY', 'kurtosisX', 'skewX', 'skewY', 'skewZ', 'freqDom1X', 'freqDom1Y', 'freqDom2X', 'freqDom2Y', 'freqDom2Z']	11
		2s	['maxX', 'maxY', 'kurtosisX', 'skewX', 'skewY', 'skewZ', 'freqDom1X', 'freqDom2X', 'freqDom2Y']	9
		4s	['maxX', 'kurtosisX', 'skewX', 'skewY', 'skewZ', 'freqDom1X', 'freqDom1Z']	7
		8s	['maxX', 'kurtosisX', 'skewX', 'skewY', 'skewZ', 'freqDom1X', 'freqDom1Y', 'freqDom1Z']	8
Giroscopio	12 Hz	1s	['maxX', 'kurtosisX', 'skewY', 'skewZ', 'freqDom1X', 'freqDom1Y', 'freqDom1Z']	7
		2s	['maxX', 'mediaZ', 'kurtosisX', 'skewY', 'skewZ', 'freqDom1X', 'freqDom1Y', 'freqDom1Z']	8
		4s	['maxX', 'mediaZ', 'skewY', 'freqDom1X']	4
		8s	['maxX', 'mediaX', 'mediaY', 'mediaZ', 'skewY', 'freqDom1X']	6
	24 Hz	1s	['maxX', 'kurtosisX', 'skewY', 'skewZ', 'freqDom1X', 'freqDom1Y']	6
		2s	['maxX', 'mediaZ', 'kurtosisX', 'skewY', 'skewZ', 'freqDom1X']	6
		4s	['maxX', 'mediaY', 'mediaZ', 'skewY', 'skewZ', 'freqDom1X']	6
		8s	['maxX', 'mediaX', 'mediaY', 'mediaZ', 'skewY', 'freqDom1X']	6

Tabla 5.1: Selección de atributos basada en correlación

- Métodos empotrados
 - Selección basada en **árboles de decisión**.

El proceso constará de dos partes. En primer lugar, induciremos un árbol de decisión a partir del dataset completo y almacenaremos los atributos por nivel de profundidad en el árbol, obteniendo así una lista de atributos ordenados de mayor a menor discriminancia. A continuación, iremos induciendo tantos árboles de decisión como atributos, de manera que en cada paso i se emplearán los atributos de la lista desde el 0 hasta el i . Evaluaremos a través de hold out estratificado $2/3 - 1/3$ la tasa de acierto del árbol inducido y nos quedaremos con el subconjunto de atributos mínimo que brinde una tasa de acierto de, al menos, 0.5.

Tras llevar a cabo la inducción de los árboles de decisión mediante la implementación de la clase `DecisionTreeClassifier` módulo `tree` de la librería `sklearn`, obtenemos las tasas de acierto en función de los atributos empleados indicadas en las figuras 5.9 y 5.10. La selección de atributos, por tanto, es la indicada en la tabla 5.2.

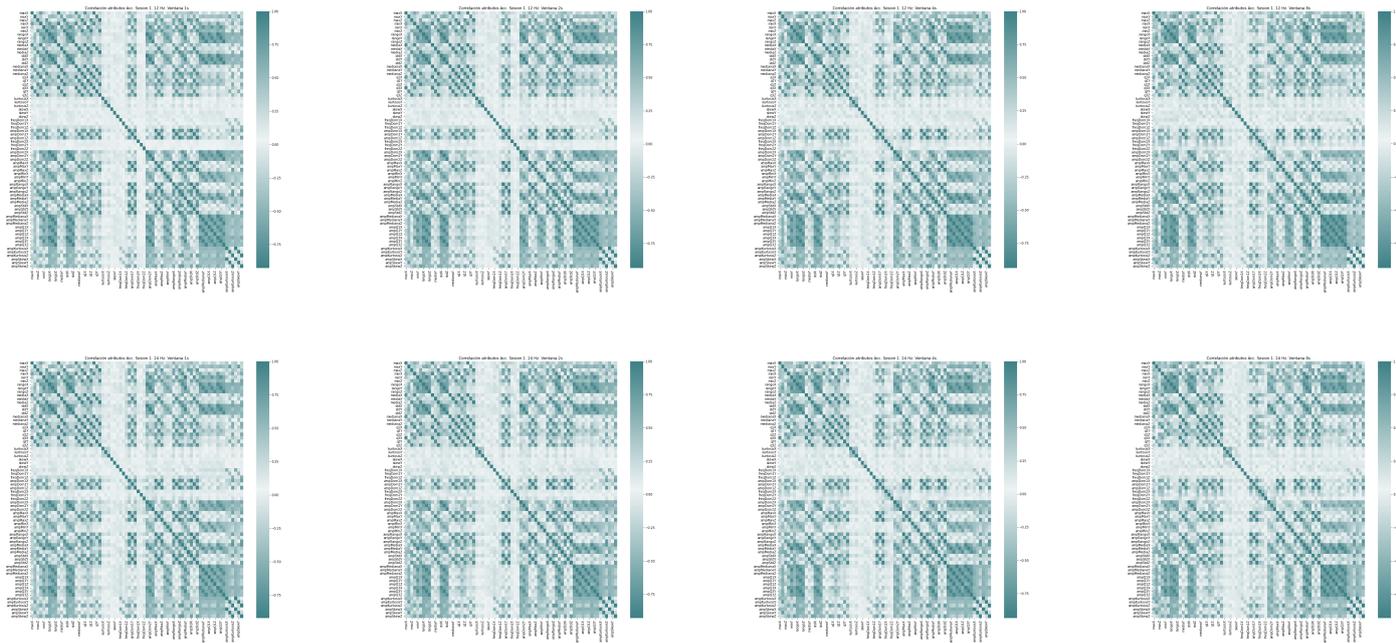


Figura 5.7: Correlación de los atributos de clasificación para el dataset limpio interpolado a 12 y 24 Hz según tamaño de ventana para el sensor acelerómetro

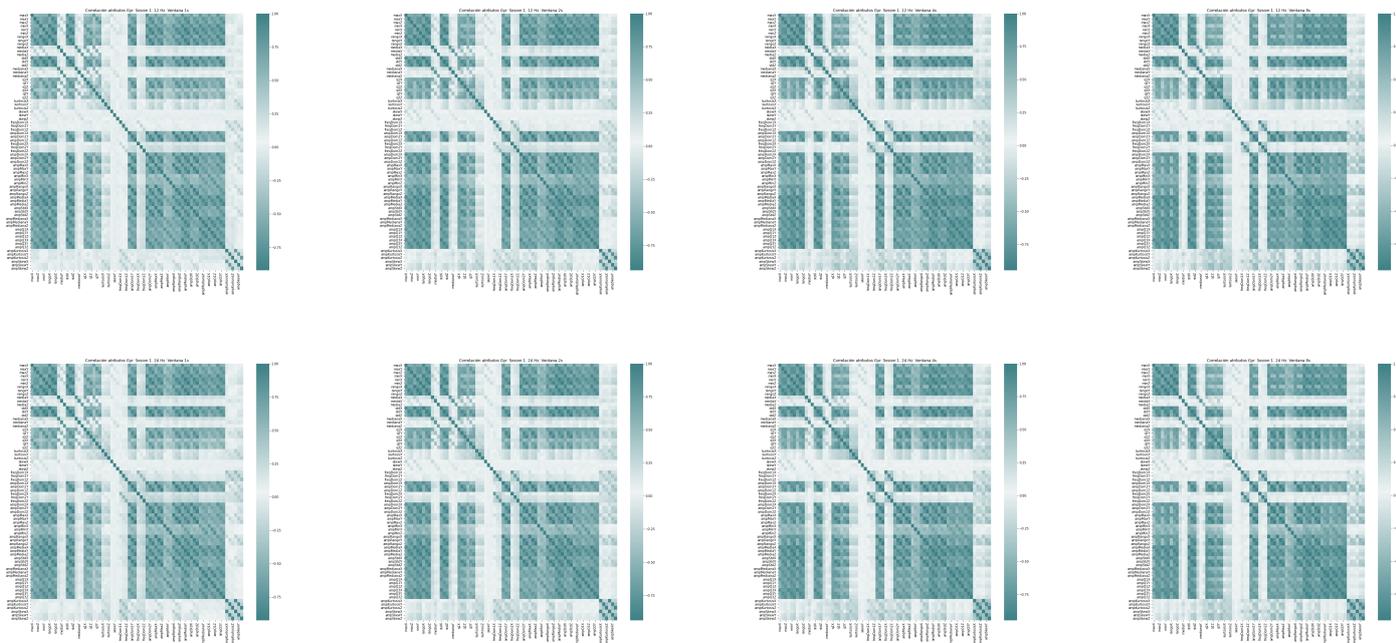


Figura 5.8: Correlación de los atributos de clasificación para el dataset limpio interpolado a 12 y 24 Hz según tamaño de ventana para el sensor giroscopio

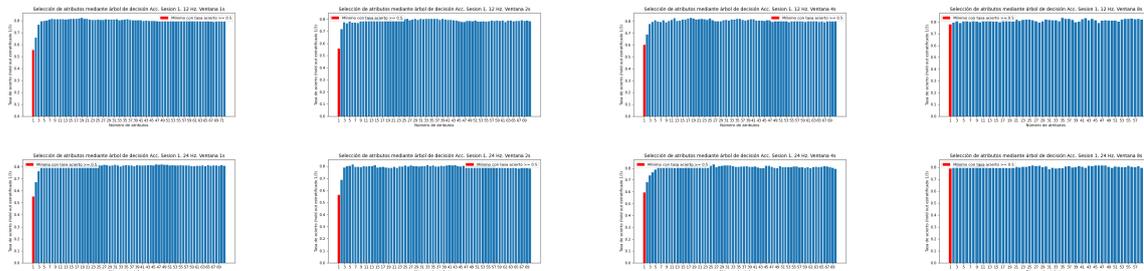


Figura 5.9: Desempeño de la selección de atributos mediante árbol de decisión para el dataset limpio interpolado a 12 y 24 Hz según tamaño de ventana para el sensor acelerómetro

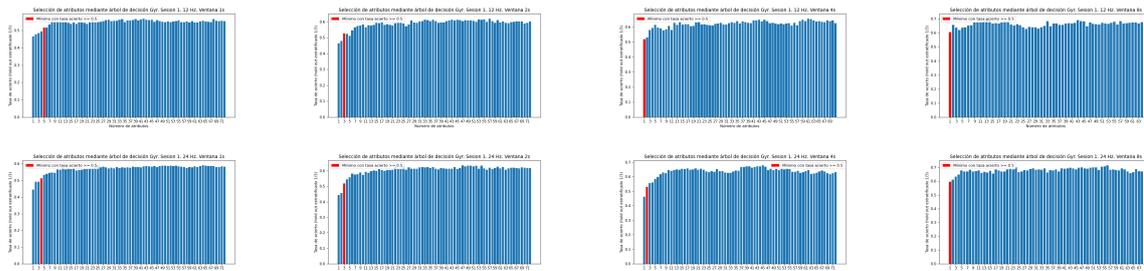


Figura 5.10: Desempeño de la selección de atributos mediante árbol de decisión para el dataset limpio interpolado a 12 y 24 Hz según tamaño de ventana para el sensor giroscopio

- o Selección basada en instancias: **Relieff**.

Relieff es una variante del algoritmo de selección de características Relief, que permite la aplicación del mismo en un problema de clasificación multinomial. Es un algoritmo bien conocido basado en la distancia entre las instancias. Su ventaja en comparación con otros métodos de selección de atributos es que no necesita suponer la independencia de los atributos[34]. Nosotros implementaremos de forma manual el algoritmo en Python.

Tras la aplicación, obtenemos la selección de atributos indicada en la tabla 5.3.

- **Transformación de atributos.** El objetivo de la transformación de atributos es evitar la maldición de la dimensionalidad. Para ello, generaremos atributos nuevos mediante el **análisis de componentes principales (PCA)**. El PCA es una técnica empleada para expresar conjuntos de datos en términos de nuevas variables independientes y no correlacionadas entre sí, denominadas componentes principales. Cada componente principal es el resultado de una combinación lineal de las variables originales de forma que se pierda la menor cantidad de información posible[35]. Emplearemos la implementación del PCA disponible en el módulo `decomposition` de la librería `sklearn`. Así, generamos tres componentes principales (PC1, PC2 y PC3) para cada dataset.

5.4.2. Normalización de los datos

El buen comportamiento de los métodos de aprendizaje basados en instancias es altamente dependiente de la definición de la distancia entre las instancias del conjunto de datos. Esta distancia puede verse distorsionada por la escala de los atributos, que ponderará a unos por encima de otros tan solo por una cuestión de diferencia en las unidades de medida, haciendo que la clasificación se base más en el azar que en la proximidad entre instancias.

Para contrarrestar la escala de los atributos, llevaremos los datos al intervalo $[0, 1]$ a través de la **normalización Min-Max**, de tal manera que de cada valor v de un atributo A en el intervalo original $[min_A, max_A]$

Sensor	Frecuencia interpolación	Tamaño ventana	Selección atributos	Total
Acelerómetro	12 Hz	1s	['q1X']	1
		2s	['q1X']	1
		4s	['q1X']	1
		8s	['q1X']	1
	24 Hz	1s	['q1X']	1
		2s	['q1X']	1
		4s	['q1X']	1
		8s	['q1X']	1
Giroscopio	12 Hz	1s	['stdZ', 'ampMedianaY', 'ampMedianaZ', 'stdY', 'ampMedianaX']	5
		2s	['q3Z', 'ampMedianaY', 'q1Z']	3
		4s	['q3Z']	1
		8s	['q3Y']	1
	24 Hz	1s	['stdZ', 'ampMedianaY', 'ampMedianaZ', 'stdY']	4
		2s	['stdZ', 'ampMedianaZ', 'maxY']	3
		4s	['q3Z', 'q1Y']	2
		8s	['q3Z']	1

Tabla 5.2: Selección de atributos basada en árboles de decisión

Sensor	Frecuencia interpolación	Tamaño ventana	Selección atributos	Total
Acelerómetro	12 Hz	1s	['skewX', 'ampKurtosisY', 'ampMinZ']	3
		2s	['ampRangoY', 'ampMaxY', 'ampDom1Y']	3
		4s	['freqDom2Y', 'ampKurtosisX', 'freqDom2Z']	3
		8s	['ampKurtosisX', 'ampKurtosisY', 'freqDom2Z']	3
	24 Hz	1s	['ampKurtosisZ', 'stdY', 'rangoY']	3
		2s	['ampKurtosisY', 'ampSkewY', 'ampKurtosisZ']	3
		4s	['ampKurtosisY', 'ampSkewY', 'ampKurtosisZ']	3
		8s	['ampSkewX', 'ampKurtosisX', 'ampKurtosisY']	3
Giroscopio	12 Hz	1s	['skewY', 'freqDom2Y', 'freqDom2Z']	3
		2s	['ampQ3Y', 'medianaX', 'ampSkewY']	3
		4s	['q1Z', 'ampKurtosisX', 'ampKurtosisY']	3
		8s	['freqDom1X', 'minY', 'freqDom1Z']	3
	24 Hz	1s	['skewY', 'ampMedianaZ', 'ampMinX']	3
		2s	['q1Z', 'ampKurtosisX', 'ampStdZ']	3
		4s	['q1Z', 'ampKurtosisX', 'ampStdZ']	3
		8s	['freqDom1Y', 'freqDom2X', 'rangoZ']	3

Tabla 5.3: Selección de atributos mediante ReliefF

se obtendrá un valor escalado[15]:

$$v' = \frac{v - \min_A}{\max_A - \min_A} \quad (5.5)$$

5.4.3. Reducción de los datos

La selección y transformación de atributos es un proceso que reduce tanto la dimensionalidad de los datos como la complejidad de los clasificadores inducidos. Sin embargo, existen algoritmos de aprendizaje tanto supervisado como no supervisado para los que no solo una dimensionalidad de datos alta resulta contraproducente, sino también una gran cantidad de instancias repercuten en redundancia e ineficiencia. Típicamente se trata de los métodos de evaluación perezosa, siendo los métodos basados en instancias el mayor exponente de los mismos.

La solución más sencilla a esto consiste en realizar un **muestreo aleatorio** de los datos antes del entrenamiento en los métodos basados en instancias. Puesto que deseamos que el subconjunto de datos muestreado sea representativo de las características del conjunto global, realizaremos este muestreo de forma **estratificada**, de tal manera que la distribución de clases de las instancias del conjunto global sea la misma que la del conjunto muestreado.

Capítulo 6

Minería de datos y evaluación de resultados

6.1. Descripción del dataset procesado

Tras el procesamiento principal de datos indicado en la sección 5, nos encontramos ante **16 datasets** caracterizados según frecuencia de muestreo (12 Hz, 24 Hz), el tamaño de las ventanas (1s, 2s, 4s y 8s) y el sensor de captura de la misma (acelerómetro, giroscopio).

Cada instancia del dataset (ventana) presenta un total de **72 atributos numéricos continuos** y un atributo de clase, *actividad*, con **6 valores** distintos. La distribución del número de ventanas para cada caso se muestra en la figura 6.1. El número concreto de instancias de cada dataset viene reflejado en la tabla 6.1 y en la figura 6.2.

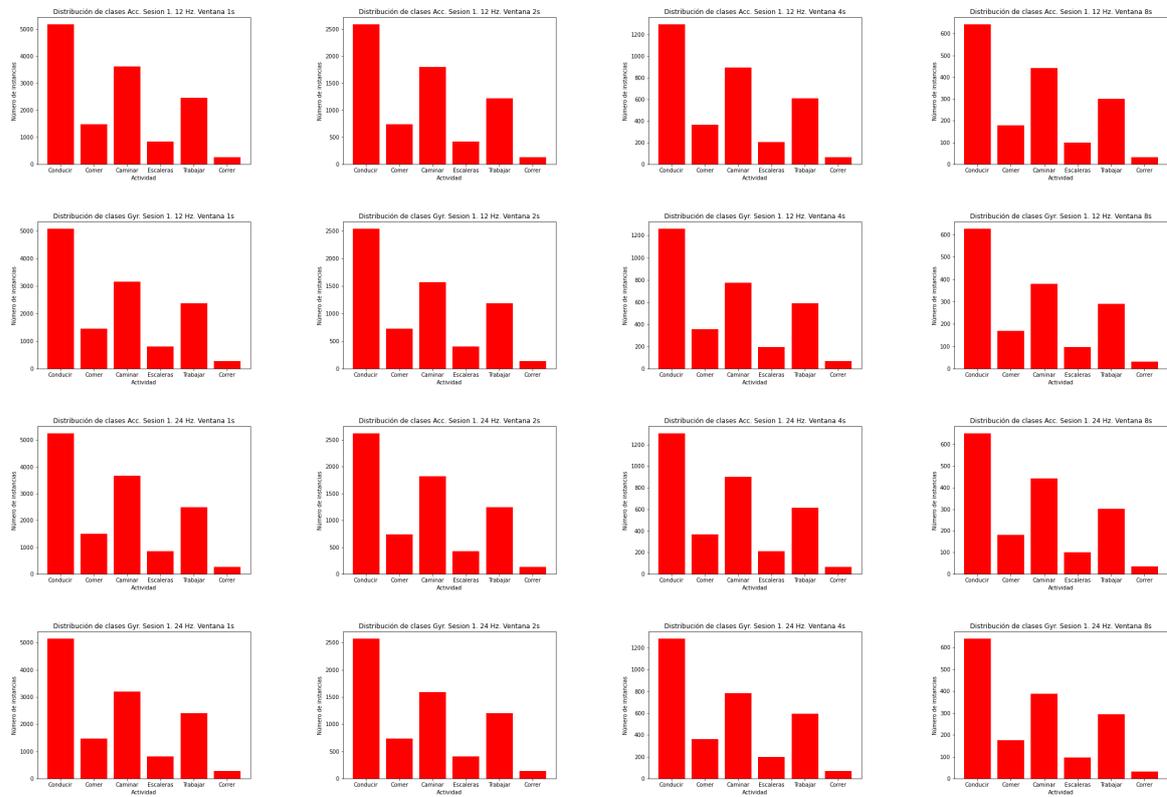


Figura 6.1: Distribución de clases de los datasets procesados según sensor y frecuencia de muestreo

Sensor	Frecuencia interpolación	Tamaño ventana	Número instancias
Acelerómetro	12 Hz	1s	13868
		2s	6909
		4s	3432
		8s	1697
	24 Hz	1s	14026
		2s	6985
		4s	3463
		8s	1710
Giroscopio	12 Hz	1s	13140
		2s	6538
		4s	3236
		8s	1593
	24 Hz	1s	13304
		2s	6626
		4s	3288
		8s	1629

Tabla 6.1: Tamaño de los datasets procesados

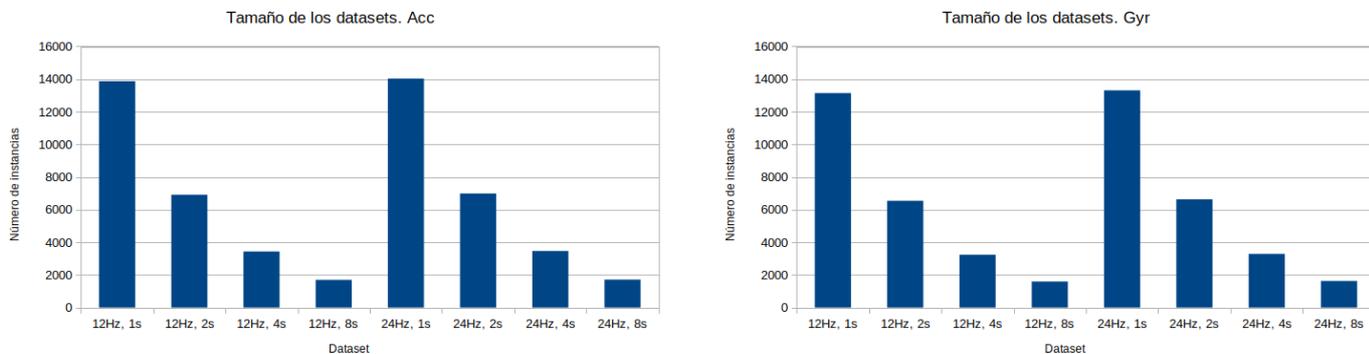


Figura 6.2: Visualización del tamaño de los datasets procesados por sensor

6.2. Metodología de evaluación

Para poder evaluar el rendimiento de los clasificadores, emplearemos las siguientes métricas adaptadas a la clasificación multinomial[25]:

- **Tasa de acierto:** mide la proporción de instancias correctamente predichas sobre el total de las predicciones.

$$\text{Tasa de acierto} = \frac{TP + TN}{TP + TN + FP + FN} \quad (6.1)$$

- **Precision:** mide el número de instancias correctamente predichas de cada clase con respecto a todas las instancias de las predichas como de la clase.

$$\text{Precision clase } c = \frac{tp_c}{tp_c + fp_c} \quad (6.2)$$

$$\text{Precision total} = \frac{1}{C} \left(\sum_{c=1}^C \frac{tp_c}{tp_c + fp_c} \right) \quad (6.3)$$

- **Recall**: mide el número de instancias correctamente clasificadas sobre el total de instancias de la clase.

$$\text{Recall clase } c = \frac{tp_c}{tp_c + fn_c} \quad (6.4)$$

$$\text{Recall total} = \frac{1}{C} \left(\sum_{c=1}^C \frac{tp_c}{tp_c + fn_c} \right) \quad (6.5)$$

- **F1**: media armónica entre precisión y recall.

$$F1 = \sum_{c=1}^C 2 \left(\frac{n_c}{N} \right) \frac{\text{precision}_c \cdot \text{recall}_c}{\text{precision}_c + \text{recall}_c} \quad (6.6)$$

Donde:

- C es el número de clases del dataset.
- tp_c es la tasa de ciertos positivos de la clase c y $TP = \sum_{c=1}^C tp_c$ es la tasa de ciertos positivos total.
- fp_c es la tasa de falsos positivos de la clase c y $FP = \sum_{c=1}^C fp_c$ es la tasa de falsos positivos total.
- tn_c es la tasa de ciertos negativos de la clase c y $TN = \sum_{c=1}^C tn_c$ es la tasa de ciertos negativos total.
- fn_c es la tasa de falsos negativos de la clase c y $FN = \sum_{c=1}^C fn_c$ es la tasa de falsos negativos total.

La técnica de evaluación que adoptaremos sobre los conjuntos de datos dependerá de la naturaleza del algoritmo de aprendizaje. En el caso de los algoritmos de aprendizaje supervisado, dada la alta dispersión de la distribución de clases de los datasets (figura 6.1), emplearemos como metodología de evaluación de los clasificadores la **validación cruzada estratificada de 15 particiones con barajado aleatorio**. Por otro lado, en el caso del aprendizaje no supervisado, emplearemos como metodología la **estimación de resubstitución**, que consiste en emplear el conjunto completo de datos tanto para inducir como para evaluar el modelo.

Para la implementación de la evaluación de clasificadores de aprendizaje supervisado, haremos uso de la implementación en la clase `StratifiedKFold` del módulo `model_selection` de la librería `sklearn` para Python. A su vez, para la obtención de las métricas emplearemos las funciones `accuracy_score`, `precision_score`, `recall_score` y `f1_score` del módulo `metrics` de la librería `sklearn` para Python.

6.3. Configuración experimental

6.3.1. Parámetros de experimentación

Tras el procesamiento de los datos, hemos obtenido un total de 16 datasets en función tanto del **sensor de recogida de datos** como de la **frecuencia de interpolación** y el **tamaño de ventana**. Analizaremos la configuración que mejores resultados obtiene basándonos en la metodología de evaluación previamente indicada en la sección 6.2, en función de la inducción de clasificadores a partir del total de los subconjuntos de atributos obtenidos en la selección y transformación de atributos indicada en la sección 5.4.1:

- Atributos **sin selección**.
- Atributos tras selección basada en **correlación**.

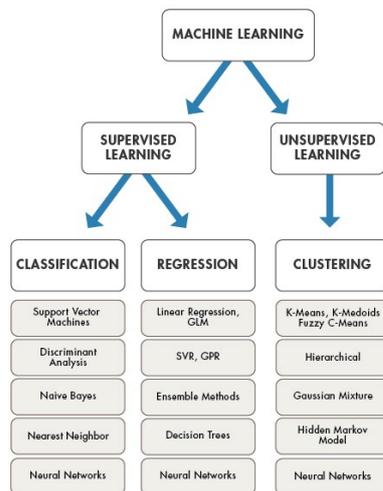


Figura 6.3: Algoritmos de aprendizaje más populares según el paradigma de aprendizaje automático

- Atributos tras selección mediante **ReliefF**.
- Atributos tras **análisis de componentes principales**.

A su vez, experimentaremos con los argumentos relevantes de cada algoritmo de aprendizaje para la mejor configuración de entre los anteriores parámetros.

6.3.2. Algoritmos de aprendizaje

Dado que nos encontramos ante una investigación de carácter prospectivo, hemos aplicado los algoritmos más relevantes de las dos modalidades principales del aprendizaje automático (figura 6.3) con el fin de obtener el tipo de algoritmo que mejor se comporta para el reconocimiento de actividades humanas relajado tras recogida de muestras en entorno natural.

6.3.2.1. Aprendizaje supervisado

El aprendizaje automático supervisado es el paradigma de aprendizaje automático en el que los modelos aprenden a relacionar variables de entrada con su salida esperada mediante el ajuste a ejemplos etiquetados con relaciones conocidas[36]. Esto es tanto como decir que el aprendizaje supervisado se encarga de inferir una función que generalice la relación entre instancias y salidas. Así, dependiendo de la naturaleza de las relaciones, nos encontraremos ante un problema de **clasificación** si los valores de salida son valores categóricos, o un problema de **regresión** si lo que se pretende inferir son valores numéricos.

El problema de reconocimiento de actividades humanas corresponde a un problema de clasificación. Dado un conjunto de muestras de características de señales como entrada y una etiqueta de actividad asociada como salida, el objetivo del aprendizaje supervisado consistirá en inferir un modelo que sea capaz de aprender la relación subyacente entre las características de las señales y la actividad a la que corresponden.

Los algoritmos de aprendizaje automático que emplearemos son los siguientes:

- **Árbol de decisión.** Un árbol de decisión es un modelo de predicción que es capaz de asignar un valor de clase a cada instancia de entrada en función de condiciones cumplidas simultáneamente por sus atributos. El proceso de obtención de un árbol de decisión se conoce como inducción, y representa una de las familias de algoritmos de aprendizaje automático más populares. Su funcionamiento se basa en teoría

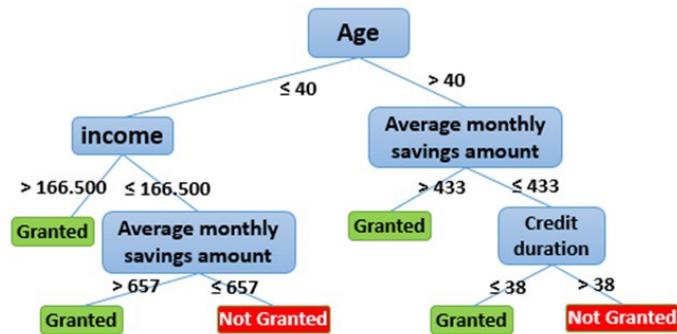


Figura 6.4: Ejemplo de árbol de decisión para la concesión de préstamos bancarios

de la información, discriminando los valores de los atributos en función de su ganancia de información con respecto a los valores de clase. Una de sus ventajas fundamentales es la interpretabilidad[37], dado que de un árbol se pueden extraer reglas legibles para la clasificación. Un ejemplo claro de la interpretabilidad de un árbol de decisión lo encontramos en la figura 6.4, correspondiente a un árbol de decisión para la concesión de préstamos bancarios. Como podemos observar, el criterio para la concesión o no de un préstamo es completamente comprensible y explicitable en términos de lenguaje natural.

Para la inducción de árboles de decisión, no es necesario que realicemos **ningún tratamiento ad hoc previo** a los datos procesados. Los árboles de decisión tienen la capacidad de discretizar los atributos numéricos según su ganancia de información, por lo que no se ven afectados por la escala de los mismos. Además, los árboles de decisión tienen la ventaja de ser robustos frente al ruido en los datos[37], por lo que si la suposición que hicimos en la sección 5.1 sobre valores atípicos de la amplitud de las señales es incorrecta y sí existe ruido asistemático, previsiblemente el desempeño de la clasificación mediante árboles no se verá afectado.

Sin embargo, este tipo de algoritmos son propensos a sobreajustar los modelos cuando los árboles inducidos son de gran profundidad[37], por lo que trataremos de evitar este efecto variando la altura máxima de inducción de los árboles de decisión hasta obtener la configuración óptima.

La implementación del algoritmo de inducción de árboles de decisión que vamos a utilizar es la correspondiente a la clase `DecisionTreeClassifier` del módulo `sklearn.tree`. La configuración por defecto de este árbol corresponde a la construcción de un árbol binario sin límite de profundidad que emplea como criterio de discriminación de atributos la impureza Gini. Modificaremos el parámetro `max_depth` para controlar la altura y evitar el sobreajuste.

- **k-NN**. El algoritmo de los k vecinos más próximos (k-NN) es un algoritmo de aprendizaje basado en instancias y de evaluación perezosa, que a cada instancia de prueba le asigna el valor de clase modal de sus k vecinos más próximos de entre las instancias de entrenamiento, en función de la definición de distancia deseada[37]. Por tanto, es un algoritmo que basa la clasificación en la similitud entre instancias.

Como hemos comentado en la sección 5.4.2, la efectividad de este tipo de algoritmos está fuertemente ligada a la escala de los atributos, por lo que aplicaremos una normalización de los datos. Variaremos el valor de k desde 1 hasta 10 para conocer la tendencia del desempeño en función de la configuración del algoritmo, aplicando en todo caso la distancia euclídea entre las instancias.

Por otro lado, dada la evaluación perezosa del algoritmo, su coste computacional está ligado tanto al número de instancias de entrenamiento empleadas como a la dimensión de las mismas. Manejamos el efecto de la dimensionalidad de los datos a través de la selección y transformación de atributos explicada en la sección 5.4.1. Por otro lado, reduciremos el número de instancias a través de un muestreo aleatorio

estratificado de un 50 % de las mismas[37]. Como norma general, si a un método basado en instancias que se ha demostrado útil para la clasificación se le aplica una reducción del conjunto de entrenamiento, tenderá a ser capaz de mantener o incluso aumentar su desempeño[38].

Emplearemos la implementación de k-NN de la clase `KNeighborsClassifier` del módulo `sklearn.neighbors`, manteniendo todos los parámetros constantes salvo `n_neighbors`, con el que controlaremos el valor de los k vecinos a considerar para la clasificación.

- **Naive Bayes.** El algoritmo Naive Bayes es un método de inducción de clasificadores basado en probabilidad. Parte de la suposición de la independencia de los atributos entre sí para poder estimar la probabilidad condicionada a pertenecer a una clase de una instancia como el producto de las probabilidades condicionadas de sus atributos, aproximadas a partir las instancias del conjunto de entrenamiento[37].

De la misma forma que los árboles de decisión, el uso de Naive Bayes no requiere de un procesamiento ad-hoc de los datos, dado que en su versión para atributos numéricos se realiza una discretización de los mismos. Su buen funcionamiento depende principalmente de la corrección en la suposición inicial de independencia de los atributos.

Tras la extracción de características, disponemos de atributos cuya construcción se basa en la combinación del valor de otros, por lo que es de esperar que este algoritmo no sea uno de los que mejores resultados de clasificación otorgue, al menos no sin una selección de atributos previa como la basada en correlación, que procura mantener los atributos de mayor independencia lineal entre sí.

Vamos a emplear la implementación del algoritmo Naive Bayes presente en la clase `GaussianNB` del módulo `sklearn.naive_bayes` con su configuración por defecto.

- **SVM.** Las máquinas de vectores soporte (SVM) son un clasificador lineal. La forma en la que este tipo de algoritmos funciona consiste en la búsqueda de hiperplanos capaces de separar el espacio de las instancias del conjunto de entrenamiento según la clase a la que pertenezcan, de tal forma que la distancia entre la separación realizada por los hiperplanos sea lo más ancha posible y estos sean definidos a partir de los vectores de menor distancia entre clases, los vectores soporte.

Para encontrar hiperplanos capaces de separar clases, las instancias tienen que ser cuasi-separables linealmente en el espacio de características. Sin embargo, en muchas ocasiones, esto no sucede. Para resolver el problema, SVM utiliza funciones kernel con las que expande la dimensión de los datos de entrenamiento y los lleva a aquella dimensión en la que sí sean aproximadamente separables linealmente, de tal forma que sea posible obtener el hiperplano de separación apropiado para la generalización[39] (figura 6.5).

La cuasi-separabilidad lineal se basa en la severidad con la que SVM permite cometer fallos a la hora de la separación lineal de las muestras. Este aspecto se controla a través del hiperparámetro de tuning C , que indica la forma en la que se penalizan las muestras que violan el margen y el hiperplano. Estos son precisamente los vectores soporte, que son las únicas instancias de las que deriva el hiperplano, lo que hace que SVM sea robusto frente al ruido, o sea, observaciones muy alejadas del hiperplano.[39]

En nuestro caso, evaluaremos la clasificación mediante SVM modificando la función kernel como parámetro de la clasificación, y trataremos de determinar qué función brinda los mejores resultados. Por otro lado, SVM tiende a funcionar mejor normalizando las instancias de trabajo. Esto se debe a que las funciones kernel emplean en muchos casos distancias euclídeas entre las muestras para compararlas[40], por lo que aplicaremos la normalización previamente indicada en la sección 5.4.2.

La implementación de SVM que emplearemos será la correspondiente a la clase `SVC` del módulo `sklearn.svm`. Mantendremos el hiperparámetro $C=1.0$ por defecto y variaremos el valor del parámetro `kernel` entre los valores `{'linear', 'poly', 'rbf', 'sigmoid'}`.

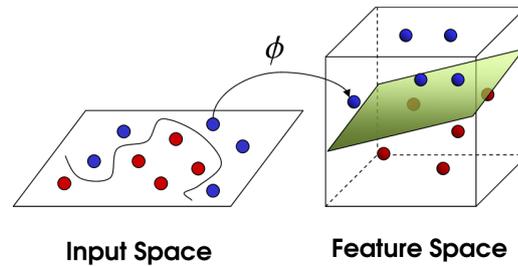


Figura 6.5: Efecto de la aplicación de funciones kernel en SVM

- Perceptrón multicapa (MLP).** Un perceptrón multicapa (MLP) es una red neuronal en la cual todas las neuronas están agrupadas en capas, de tal forma que las neuronas de una capa están conectadas con todas las neuronas de la capa inmediatamente posterior. Una neurona es la unidad básica de una red neuronal. Representa un elemento de cómputo al que le llega una serie de valores de entrada y entrega un valor de salida resultado de una suma ponderada de los valores de entrada y la aplicación a estos de una función no lineal denominada función de activación. El esquema básico de una neurona viene representado en la figura 6.6.

En un MLP como el de la figura 6.7, cada nodo representa una neurona completa. Los perceptrones clasifican las instancias de entrada a través de los valores de las neuronas de la capa de salida, de tal forma que se tiene un nodo de salida por cada clase y un nodo de entrada por cada atributo de las instancias. A cada instancia se le asigna la clase en función del valor más alto de los nodos de salida. Este valor depende de los pesos que cada neurona asigna a cada señal de entrada. Un MLP se entrena propagando el error cometido en la clasificación desde las neuronas de la capa de salida hasta las de entrada, en un proceso denominado *backpropagation*, de tal forma que en cada iteración de entrenamiento los pesos de las neuronas se ajustan en función del error de las capas posteriores hasta obtener la salida deseada para cada entrada.

Los perceptrones se caracterizan por ser aproximadores universales, es decir, tienen la capacidad de representar cualquier función de entrada. Esta es una propiedad que hace muy práctico el uso de perceptrones multicapa en cualquier tipo problema de aprendizaje automático. Sin embargo, esta propiedad presenta un inconveniente clave: los MLP tienden al sobreajuste de los valores de entrenamiento. La forma en la que trataremos de paliar este efecto es limitando el número de iteraciones de entrenamiento, épocas, de manera que los pesos de las neuronas no lleguen a ajustarse completamente al conjunto de entrenamiento para que la red no pierda la capacidad de generalizar.

La implementación que emplearemos de MLP es la correspondiente a la clase `MLPClassifier` del módulo `sklearn.neural_network`. Mantendremos la configuración por defecto de dos capas, una oculta de 100 neuronas y una de salida, pero variaremos el número de iteraciones a través del parámetro `max_iter`, estableciéndolas en un máximo de 100 en caso de no llegar a la convergencia.

6.3.2.2. Aprendizaje no supervisado

A diferencia del aprendizaje supervisado, el aprendizaje no supervisado es un paradigma de aprendizaje automático en el que los modelos tratan de aumentar el conocimiento estructural de las instancias de entrada[36].

Los algoritmos de aprendizaje no supervisado más empleados son los algoritmos de clustering, que tratan de agrupar los datos en base a la similitud de sus atributos. La forma más habitual en la que este tipo de

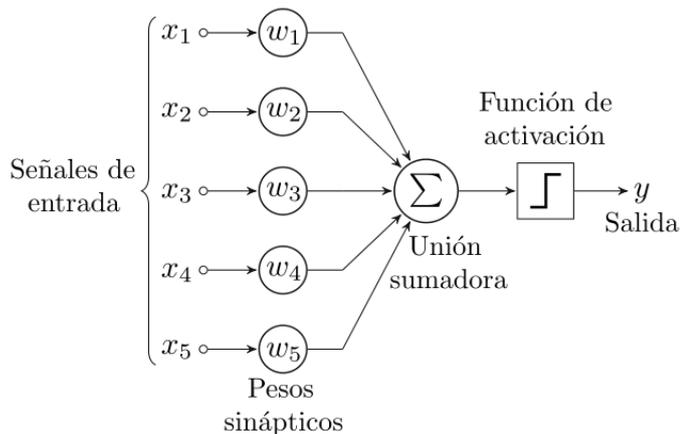


Figura 6.6: Esquema de funcionamiento de un perceptrón simple (neurona) de cinco entradas. Alejandro Cartas, CC BY-SA 4.0, via Wikimedia Commons

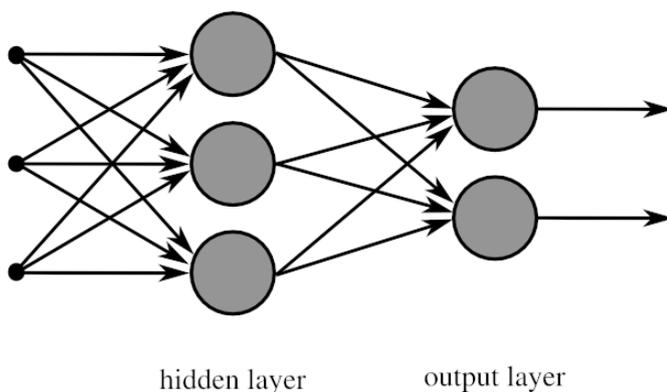


Figura 6.7: Esquema de un perceptrón multicapa de 2 capas con tres neuronas ocultas para la clasificación binaria. Chrislb, CC BY-SA 3.0, via Wikimedia Commons

algoritmos realiza el agrupamiento de las instancias es a partir de la definición previa de un número fijo de centros sobre los que se formarán los grupos, clusters, que se pretenden obtener. Según la similitud de las instancias con los centros, cada instancia quedará asignada a un cluster u otro.

A pesar de que nuestro caso corresponde a un problema típico de aprendizaje supervisado, podemos aprovechar el aprendizaje no supervisado para obtener una función de clasificación de las instancias de entrada. Para ello, agruparemos las instancias en clusters y comprobaremos las clases (actividades) de las instancias pertenecientes a cada cluster. Dependiendo del cluster en el que caiga cada muestra, se le asignará como predicción de la actividad que representa la clase mayoritaria de ese cluster.

Los algoritmos de clustering que emplearemos son los siguientes:

- **k-medias.** Es de lejos el algoritmo de clustering más popular y uno de los más eficientes y empleados. El concepto es sencillo: a partir de un número dado de centroides cuyas ubicaciones se definen de forma aleatoria, se van formando clusters agrupando las instancias más cercanas a los mismos y ajustando la ubicación de los centroides en base a las instancias del cluster formado. El método evoluciona hasta que los centroides convergen a una posición fija[41].

Este método tiende a formar clusters circulares y puede caer en mínimos locales. Para tratar de evitar esto se puede elegir la ubicación inicial de los centroides de una forma más efectiva para la convergencia que una elección aleatoria, precomputando la distancia entre muestras a través de lo que se conoce como `k-medias++`[42].

Emplearemos la implementación de k-medias de la clase `KMeans` del módulo `sklearn.cluster`, cuya

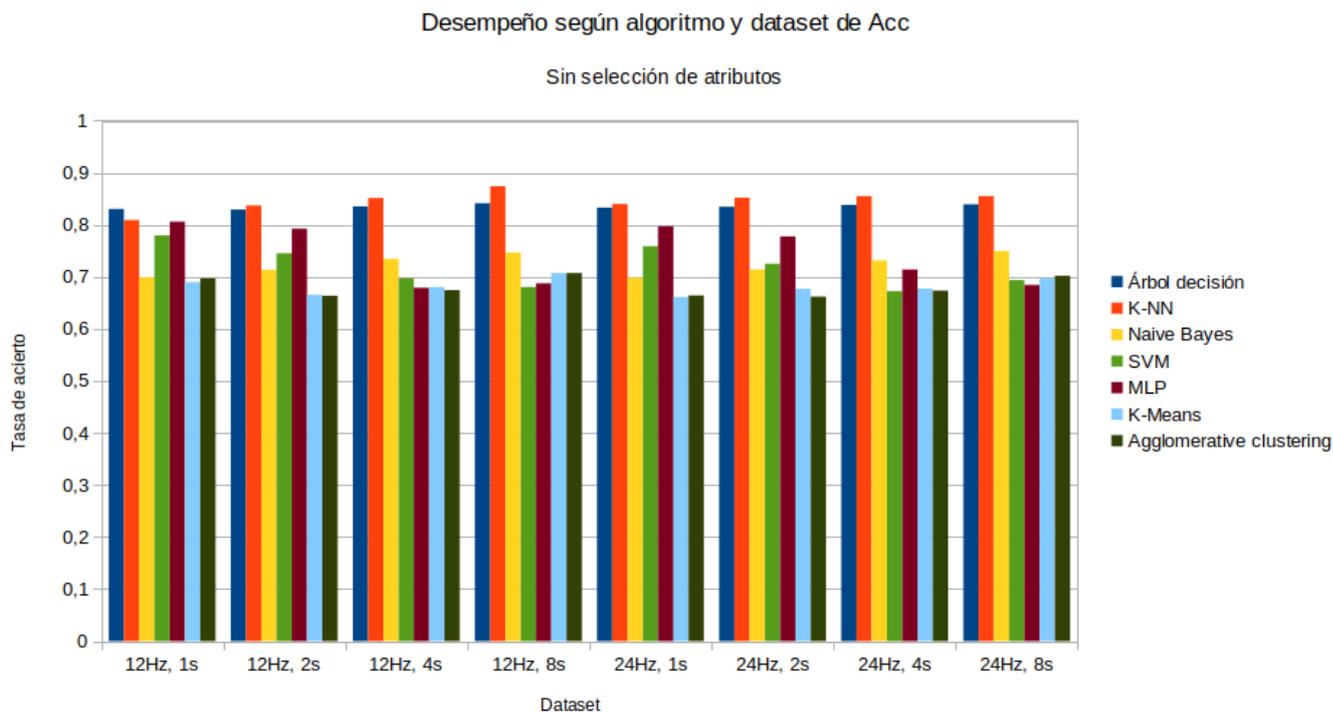


Figura 6.8: Desempeño según algoritmo de clasificación supervisada para el sensor de acelerómetro

elección inicial de centros se basa en el criterio de las `k-medias++` e iremos variando el parámetro `n_clusters` desde 6, que es el mínimo dado que tenemos 6 clases, hasta 26.

- Clustering aglomerativo.** El clustering aglomerativo es una aproximación bottom-up al proceso de clustering en la cual se parte de las instancias agrupadas en clases a las que solo pertenecen ellas mismas y se van uniendo a partir de las mínimas distancias entre clusters, hasta llegar a un único cluster que agrupa a todas las instancias. Según el número de clusters deseado, el proceso finaliza cuando existen tantas agrupaciones como se ha indicado, antes de llegar a la convergencia en el cluster único.

Emplearemos la implementación del clustering aglomerativo de la clase `AgglomerativeClustering` del módulo `sklearn.cluster`, variando el parámetro `n_clusters` desde 6, que es el mínimo dado que tenemos 6 clases, hasta 26.

6.4. Evaluación de resultados

6.4.1. Rendimiento global

Tras realizar el entrenamiento para la evaluación del desempeño de los algoritmos de aprendizaje previamente mencionados, hemos obtenido las tasas de acierto indicadas en las figuras 6.8 y 6.9 para cada dataset según el sensor de recogida.

Como podemos observar, **el uso del acelerómetro obtiene unos mejores resultados en general que el giroscopio**. En el caso de las mejores combinaciones, la clasificación de los datos del acelerómetro obtiene más de un 85% de tasa de acierto mediante la mejor configuración del algoritmo de los `k` vecinos más próximos para una frecuencia de interpolación de 12Hz y una ventana de 8 segundos, mientras que la clasificación de los datos del giroscopio otorgan una tasa de acierto máxima de poco más del 70% en el mismo algoritmo para una frecuencia de interpolación de 24Hz y una ventana de 8 segundos. Conviene recordar que la tasa de acierto del reconocimiento de actividades humanas tras recogidas en entornos sin supervisión

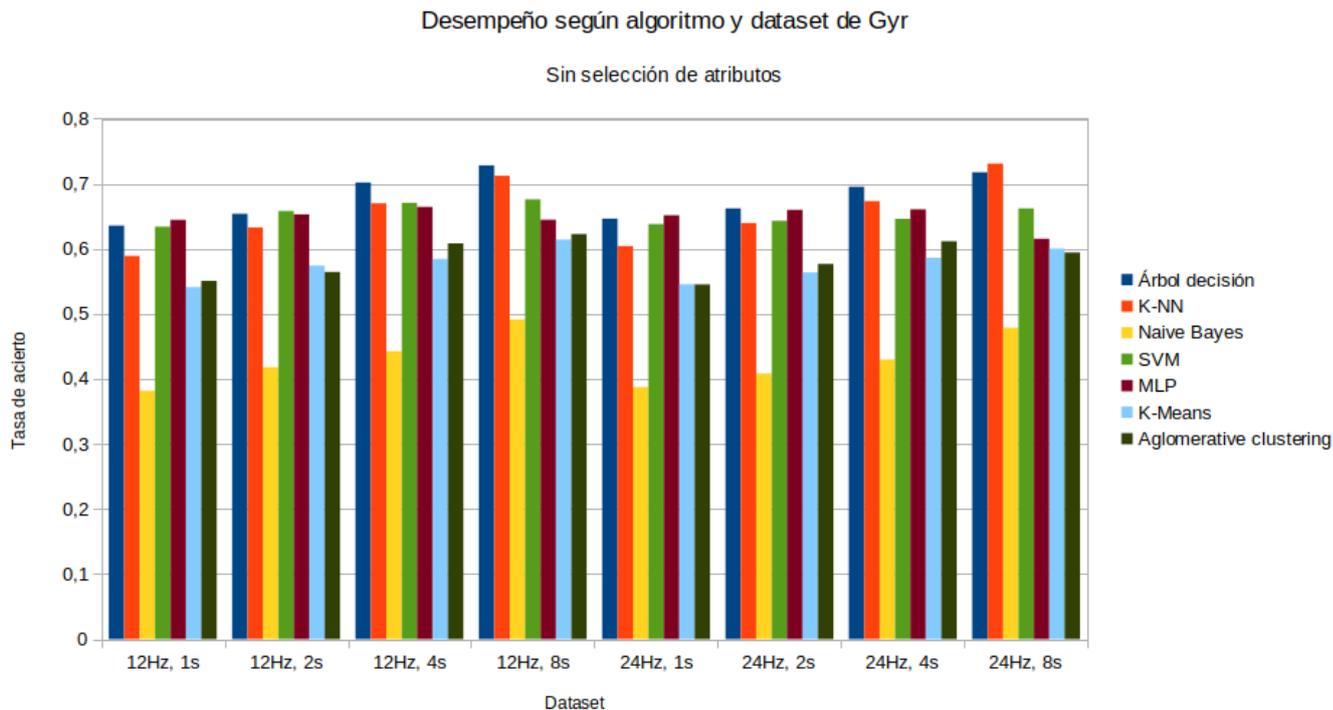


Figura 6.9: Desempeño según algoritmo de clasificación supervisada para el sensor de giroscopio

no suele alcanzar valores superiores al 90%, por lo que se puede afirmar que nos encontramos, en líneas generales, ante unos buenos resultados.

En un análisis más pormenorizado de los resultados según la **frecuencia de interpolación** (figura 6.12), nos damos cuenta de que la variación de la misma **no redonda en grandes diferencias en el desempeño** de la clasificación. Confirmamos así la intuición presentada en la sección 3.2.2 sobre la frecuencia de muestreo de dispositivos de movimiento y su efecto en la obtención de diferencias significativas.

Sin embargo, el **tamaño de ventana** juega un papel fundamental en este mismo aspecto. Esto se debe a que determina tanto la representatividad de las características extraídas sobre la señal original como el tamaño de los datasets tras el procesamiento, por lo que tiene un efecto directo en el sobreajuste de los clasificadores y en su capacidad de generalización. Este es precisamente el motivo por el cual el efecto del tamaño de ventana varía según la naturaleza del algoritmo de clasificación. Tomando el caso el acelerómetro para la frecuencia de interpolación de 24 Hz, que se aproxima más a la señal original que la de 12 Hz, el **aumento del tamaño de ventana** mantiene constante e **incluso incrementa el desempeño de los árboles de decisión, los k vecinos más próximos y Naive Bayes**, y por otro lado tiende a **disminuir la tasa de acierto de SVM y el perceptrón multicapa**. En el caso del giroscopio para los mismos conjuntos de datos, el desempeño de todos los clasificadores **sufre un aumento salvo en el uso del perceptrón y el clustering aglomerativo**. Ambos efectos vienen representados en las figuras 6.10 y 6.11.

Al aumentar el tamaño de ventana, disminuye el número de instancias procesadas (figura 6.2), por lo que los algoritmos son entrenados con una menor cantidad de datos. Los algoritmos que requieren de grandes cantidades de datos para una buena generalización tenderán a empeorar debido a este motivo. Es el caso del perceptrón multicapa, que, además, al tener limitado el número de épocas su sobreajuste está muy determinado por el número de instancias de entrenamiento.

Si atendemos a la naturaleza de los métodos de aprendizaje, **los algoritmos de aprendizaje supervisado funcionan mejor que los algoritmos de clustering no supervisado**. Tanto las k-medias como el clustering aglomerativo rinden de forma similar; sin embargo, ambos muestran un desempeño inferior al

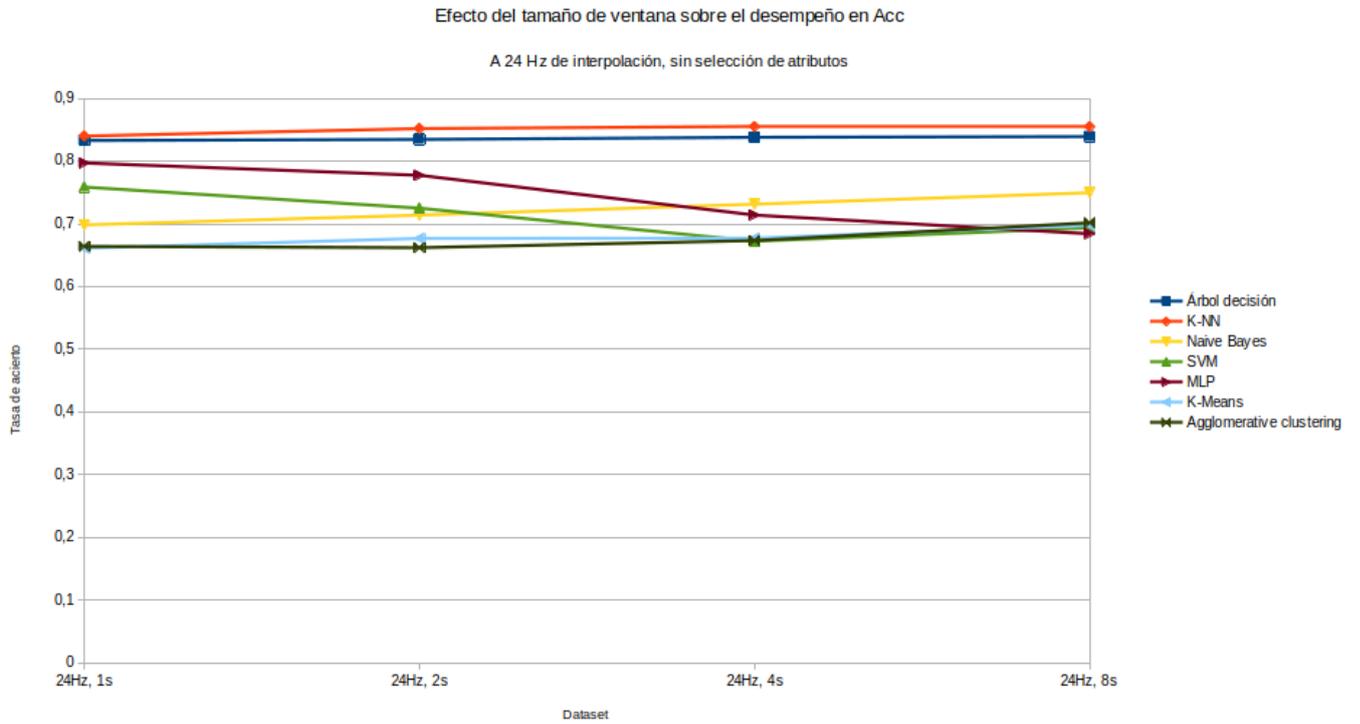


Figura 6.10: Efecto del tamaño de ventana sobre el desempeño de los clasificadores para el sensor de acelerómetro

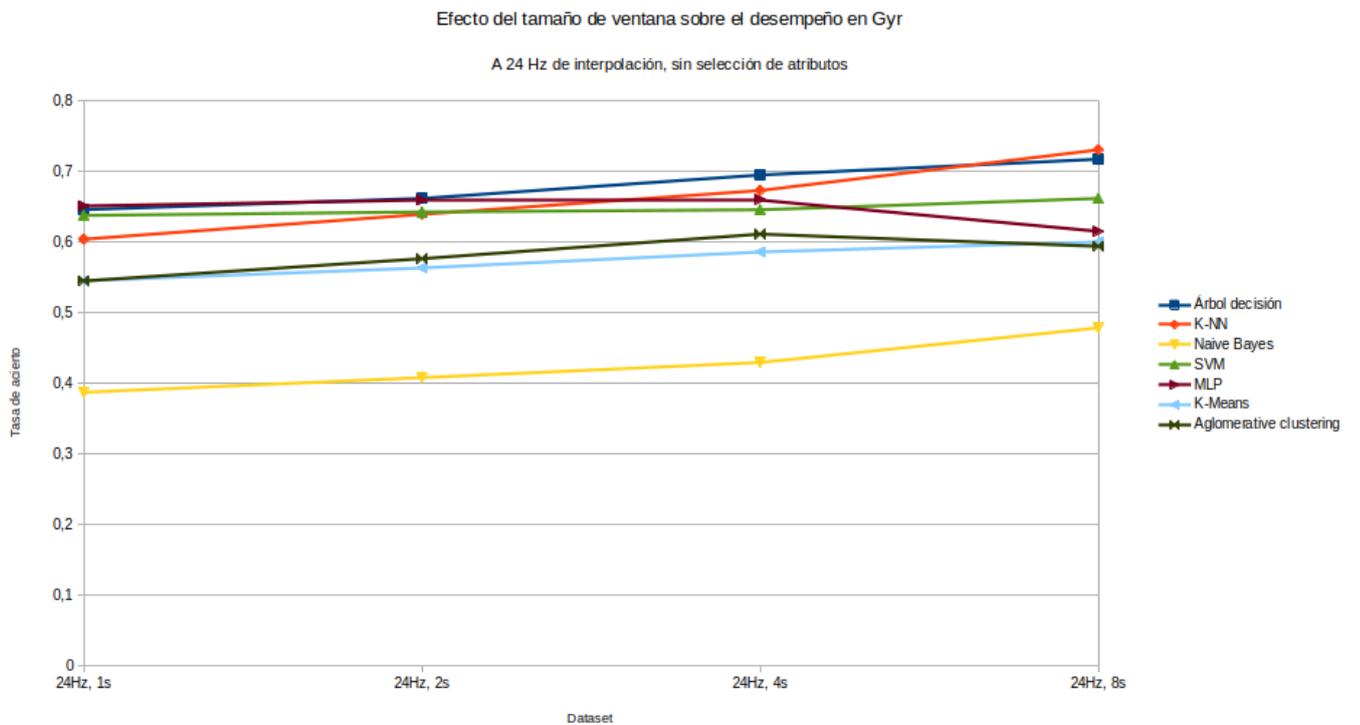


Figura 6.11: Efecto del tamaño de ventana sobre el desempeño de los clasificadores para el sensor de giroscopio

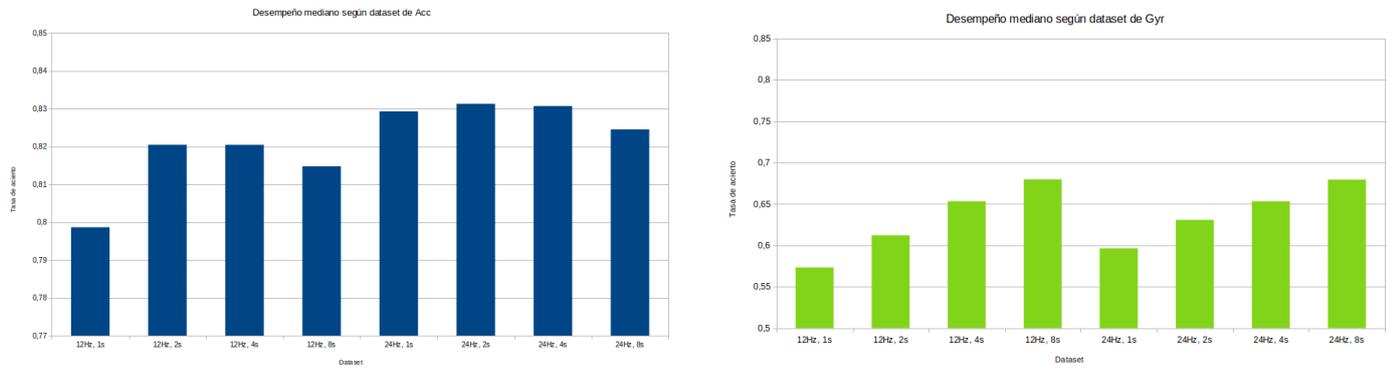


Figura 6.12: Mediana de las tasas de acierto de la clasificación de los datasets para los sensores de acelerómetro y giroscopio

resto de algoritmos salvo con respecto a Naive Bayes en el caso del giroscopio. En el caso de los algoritmos supervisados, tanto los k vecinos más próximos como los árboles de decisión muestran un buen rendimiento, y en el caso del giroscopio a estos se les suma SVM.

6.4.2. Evaluación individual de clasificadores

Para la evaluación individual de clasificadores, dados los resultados tan parejos, tomaremos como referencia la configuración de 24 Hz de frecuencia de interpolación, dada su representatividad con respecto a las señales originales, y 4 segundos de tamaño de ventana, dados sus buenos resultados conjuntos con respecto a acelerómetro y giroscopio.

6.4.2.1. Árboles de decisión

El primer experimento llevado a cabo en la inducción de árboles de clasificación consiste en la obtención de la **altura óptima del árbol** para evitar el sobreajuste en el entrenamiento. Para ello, se han ido induciendo árboles de profundidad variable desde 1 hasta 15 con los resultados en el desempeño indicados en la figura 6.14. La forma habitual de identificar el sobreajuste de un clasificador se realiza a través del análisis de bias-varianza. Dado que nuestra evaluación de los modelos se realiza a través de validación cruzada, la tasa de acierto obtenida consigue un balance entre bias y varianza que nos permite aproximar el momento de sobreajuste a través de la misma [43, 44]. Teóricamente, el sobreajuste afecta a la tasa de error (acierto) según la complejidad del modelo tal como aparece en la figura 6.13. Encontramos la tasa óptima de acierto con una profundidad de **8 nodos**, a partir de la cual tiende a estabilizarse o disminuir, dependiendo del método de selección de atributos.

En cuanto a las **selecciones de atributos**, el total de los atributos es el conjunto que mejor se comporta, seguido de la selección basada en correlación, árbol de decisión y ReliefF, que están a la par. El peor desempeño viene del análisis de componentes principales. El resumen de las métricas para el árbol de profundidad 8 se encuentra en la tabla 6.2

Atendiendo a la **precisión en la predicción de cada clase** en particular, hemos calculado la métrica F1 para la predicción de cada clase, obteniendo los resultados representados en las figuras 6.15 y 6.16. En el caso del acelerómetro, las clases con menor precisión en la predicción corresponde a las actividades de *escaleras* y *correr*, con una F1 ligeramente superior a 0.5 en el caso de ninguna selección de atributos. Por otro lado, las clases con menor precisión en la predicción a partir de datos del giroscopio corresponden a las actividad de *comer* y *correr*, que no llegan ni al 0.5. Como contraparte, las clases con mayor precisión en

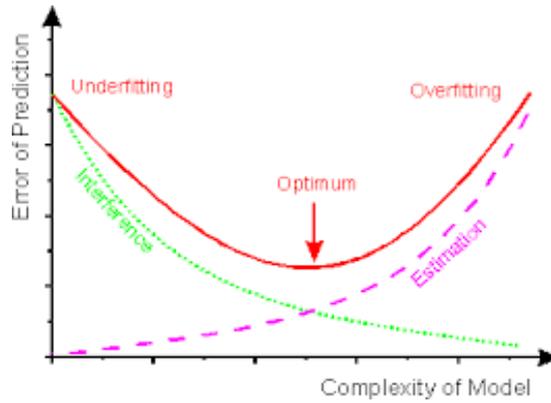


Figura 6.13: Efecto del sobreajuste en la tasa de error del modelo según su complejidad

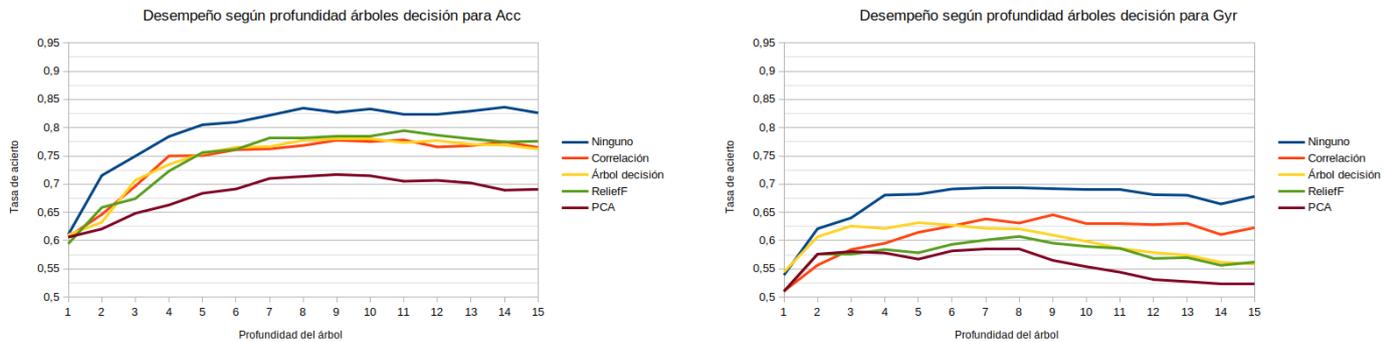


Figura 6.14: Evolución de la tasa de acierto según la profundidad de inducción del árbol de decisión de 1 a 15 nodos de profundidad

Sensor	Selección atributos	Métrica			
		Tasa de acierto	Precision	Recall	F1-score
Acelerómetro	Ninguno	0,835	0,833	0,835	0,831
	Correlación	0,768	0,766	0,768	0,760
	Árbol decisión	0,778	0,779	0,778	0,768
	ReliefF	0,781	0,785	0,781	0,773
	PCA	0,714	0,707	0,714	0,696
Giroscopio	Ninguno	0,682	0,678	0,682	0,670
	Correlación	0,638	0,634	0,638	0,615
	Árbol decisión	0,639	0,622	0,639	0,619
	ReliefF	0,622	0,614	0,622	0,610
	PCA	0,567	0,548	0,568	0,542

Tabla 6.2: Resultados de la clasificación mediante árbol de inducción de 8 nodos de profundidad

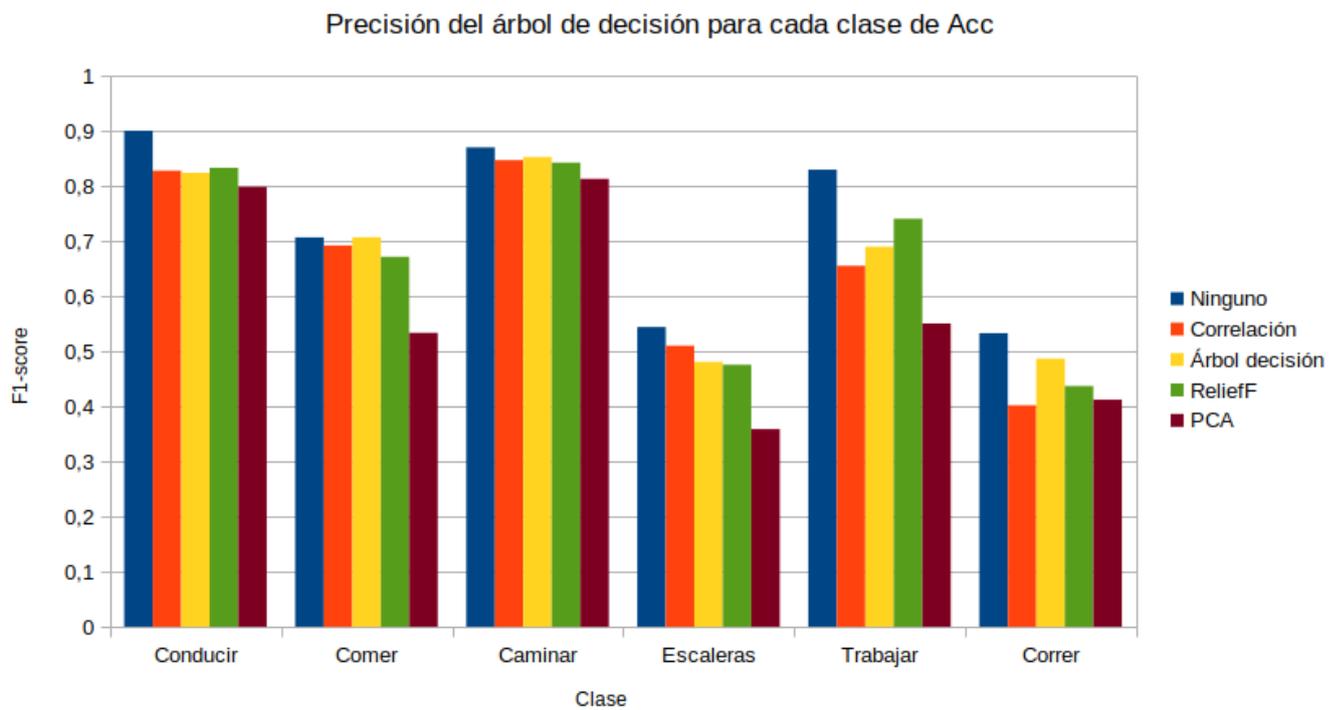


Figura 6.15: Precisión del árbol de decisión para cada clase del sensor de acelerómetro

ambos casos coinciden en ser *caminar* y *conducir*, siendo la primera la mejor detectada por el giroscopio y esta última por el acelerómetro. El ranking de la precisión de las clases quedaría de la siguiente forma (sin selección de atributos):

- Acelerómetro:

1. *Conducir*
2. *Caminar*
3. *Trabajar*
4. *Comer*
5. *Escaleras*
6. *Correr*

- Giroscopio:

1. *Caminar*
2. *Conducir*
3. *Trabajar*
4. *Escaleras*
5. *Correr*
6. *Comer*

6.4.2.2. k-NN

El uso de los k vecinos más próximos como algoritmo de clasificación suele llevar aparejado como experimento la **búsqueda del k óptimo** para la clasificación. Hemos variado el valor de k desde 1 hasta 10,

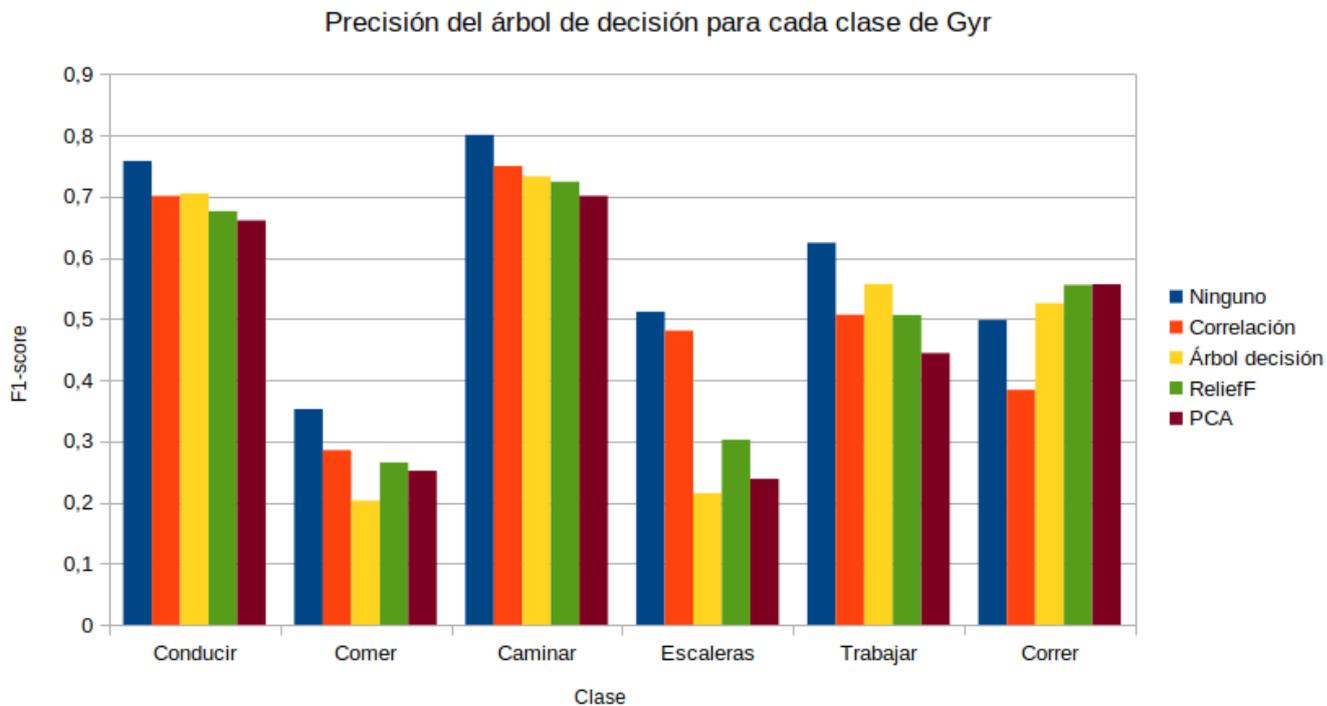


Figura 6.16: Precisión del árbol de decisión para cada clase del sensor de giroscopio

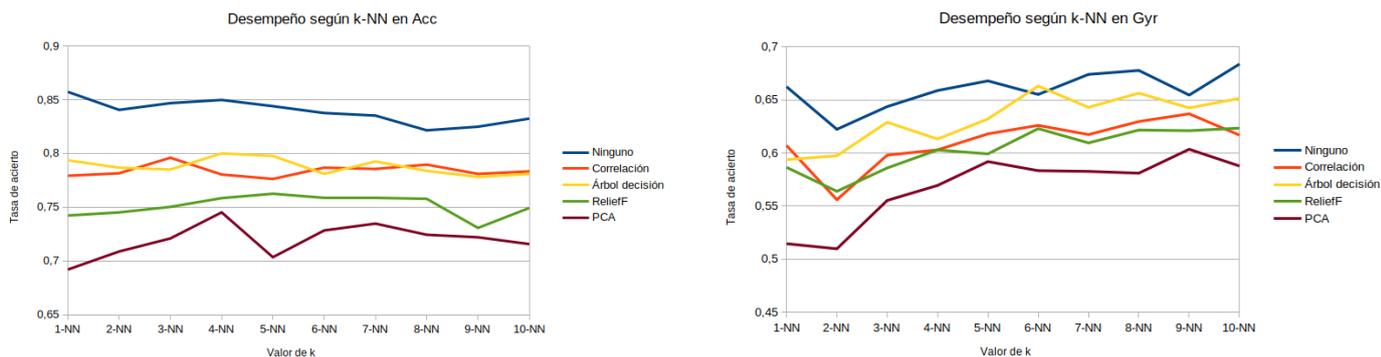


Figura 6.17: Evolución de la tasa de acierto de k-NN según el valor de k para acelerómetro y giroscopio

obteniendo las tasas de acierto representadas en la figura 6.17. Las tendencias del rendimiento según el sensor parecen ser opuestas: la tasa de acierto del acelerómetro disminuye al aumentar el valor de k cuando la del giroscopio aumenta.

Como el criterio resulta ser algo ambiguo, hemos representado la evolución de la precisión en la clasificación de cada clase según el valor de k para la clasificación sin selección de atributos en la figura 6.18. Confirmada la tendencia, obtenemos que la mayor precisión general para el **acelerómetro** se obtiene en la aplicación de **1-NN**, mientras que en el caso del **giroscopio** parece haber un valor óptimo de la precisión global en la aplicación de **8-NN**. Estos serán los valores de k con los que nos quedaremos.

De nuevo, la selección de atributos que mejores resultados brinda es el conjunto total de atributos, seguido por la selección basada en correlación y árbol de decisión a la par. En el caso del giroscopio, a estas se le suma la selección mediante en ReliefF.

Al igual que en el caso de los árboles de decisión, las clases con mayor precisión en la predicción son *conducir* y *caminar*, con valores de F1 cercanos a 0.9 en el caso del acelerómetro y en torno a 0.75 en el caso del giroscopio. Por otro lado, las clases menos precisamente predichas tienen valores más altos en el caso de

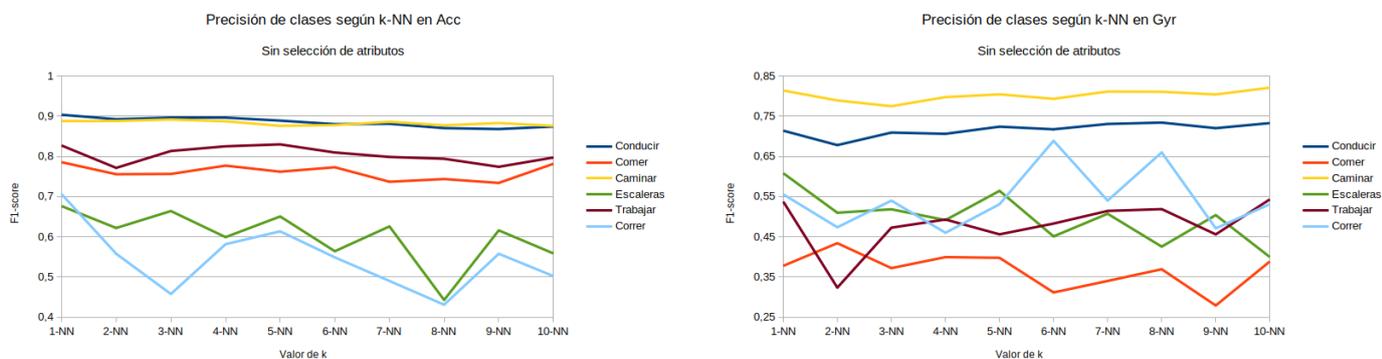


Figura 6.18: Evolución de la precisión de clase de k-NN según el valor de k sin selección de atributos para acelerómetro y giroscopio

1-NN para el acelerómetro que en el caso del árbol de decisión, situándose en torno a 0.7 de F1. El ranking de la precisión en la predicción de clases sin selección de atributos para k-NN sería el siguiente:

- Acelerómetro (1-NN):

1. *Conducir*
2. *Caminar*
3. *Trabajar*
4. *Comer*
5. *Correr*
6. *Escaleras*

- Giroscopio (8-NN):

1. *Caminar*
2. *Conducir*
3. *Correr*
4. *Trabajar*
5. *Escaleras*
6. *Comer*

Los resultados globales de la aplicación de k-NN quedan por tanto reflejados en la tabla 6.3.

6.4.2.3. Naive Bayes

La clasificación mediante Naive Bayes brinda unos resultados aceptables para la clasificación a partir de datos de acelerómetro, pero pésimos en el caso del giroscopio. La tasa de acierto en el primer caso se sitúa en torno al 70 %, mas en el segundo apenas supera el 40 %. Podemos suponer, por tanto, que las características extraídas del acelerómetro cumplen mejor con la condición de independencia del clasificador que las características extraídas del giroscopio. Los resultados completos se muestran en la tabla 6.4.

En cuanto a la precisión en la predicción de las clases para el caso del acelerómetro, se obtienen los resultados mostrados en las figuras 6.19 y 6.20. Como se puede observar, apenas tiene precisión para predecir la clase *escaleras* ni *correr*. Sin embargo, la precisión para el resto de clases es relativamente aceptable.

Sensor	Selección atributos	Métrica			
		Tasa de acierto	Precision	Recall	F1-score
Acelerómetro (1-NN)	Ninguno	0,862	0,865	0,862	0,859
	Correlación	0,772	0,774	0,772	0,769
	Árbol decisión	0,789	0,794	0,789	0,788
	ReliefF	0,742	0,744	0,742	0,738
	PCA	0,701	0,708	0,701	0,701
Giroscopio (8-NN)	Ninguno	0,662	0,655	0,662	0,637
	Correlación	0,620	0,607	0,6205	0,599
	Árbol decisión	0,635	0,613	0,635	0,609
	ReliefF	0,660	0,660	0,660	0,635
	PCA	0,591	0,579	0,591	0,567

Tabla 6.3: Resultados de la clasificación mediante k-NN

Sensor	Selección atributos	Métrica			
		Tasa de acierto	Precision	Recall	F1-score
Acelerómetro	Ninguno	0,731	0,751	0,731	0,732
	Correlación	0,640	0,612	0,640	0,614
	Árbol decisión	0,678	0,629	0,678	0,645
	ReliefF	0,629	0,592	0,629	0,594
	PCA	0,627	0,564	0,627	0,574
Giroscopio	Ninguno	0,429	0,505	0,429	0,413
	Correlación	0,419	0,452	0,419	0,385
	Árbol decisión	0,435	0,438	0,435	0,391
	ReliefF	0,404	0,426	0,404	0,361
	PCA	0,402	0,379	0,402	0,362

Tabla 6.4: Resultados de la clasificación mediante Naive Bayes

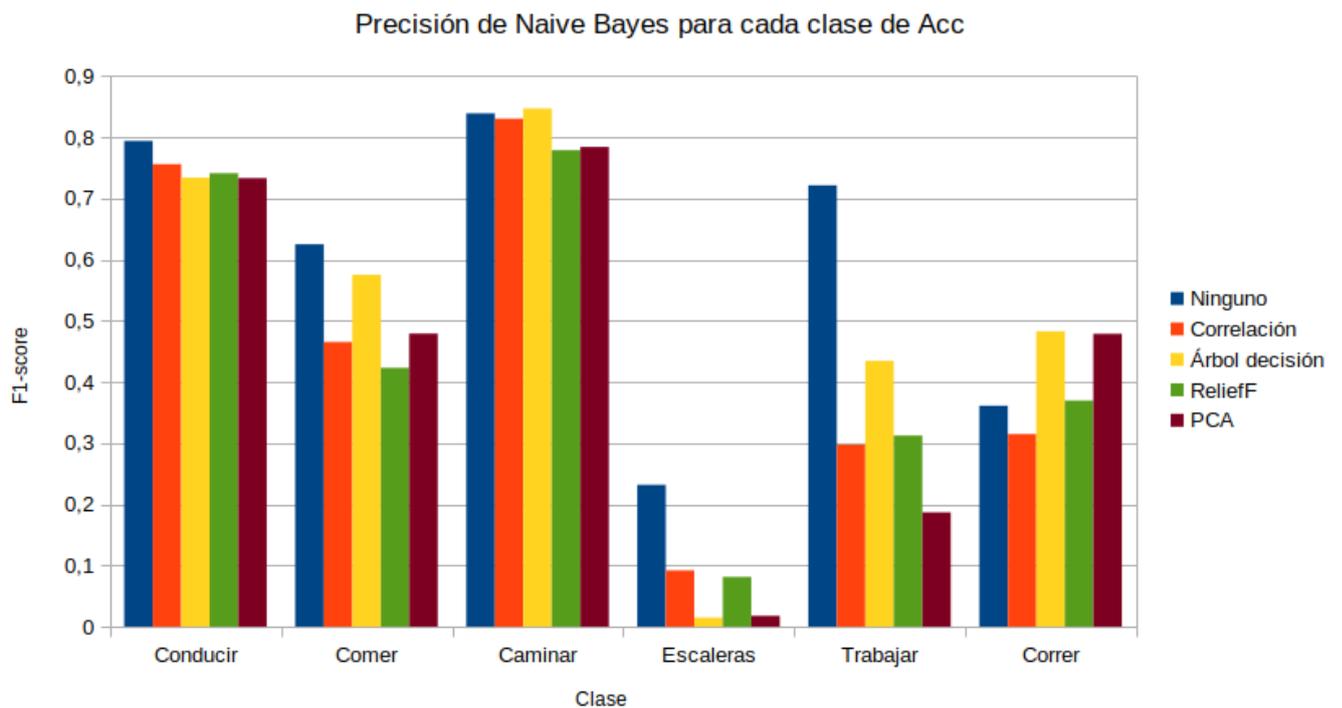


Figura 6.19: Precisión de Naive Bayes para cada clase del sensor de acelerómetro

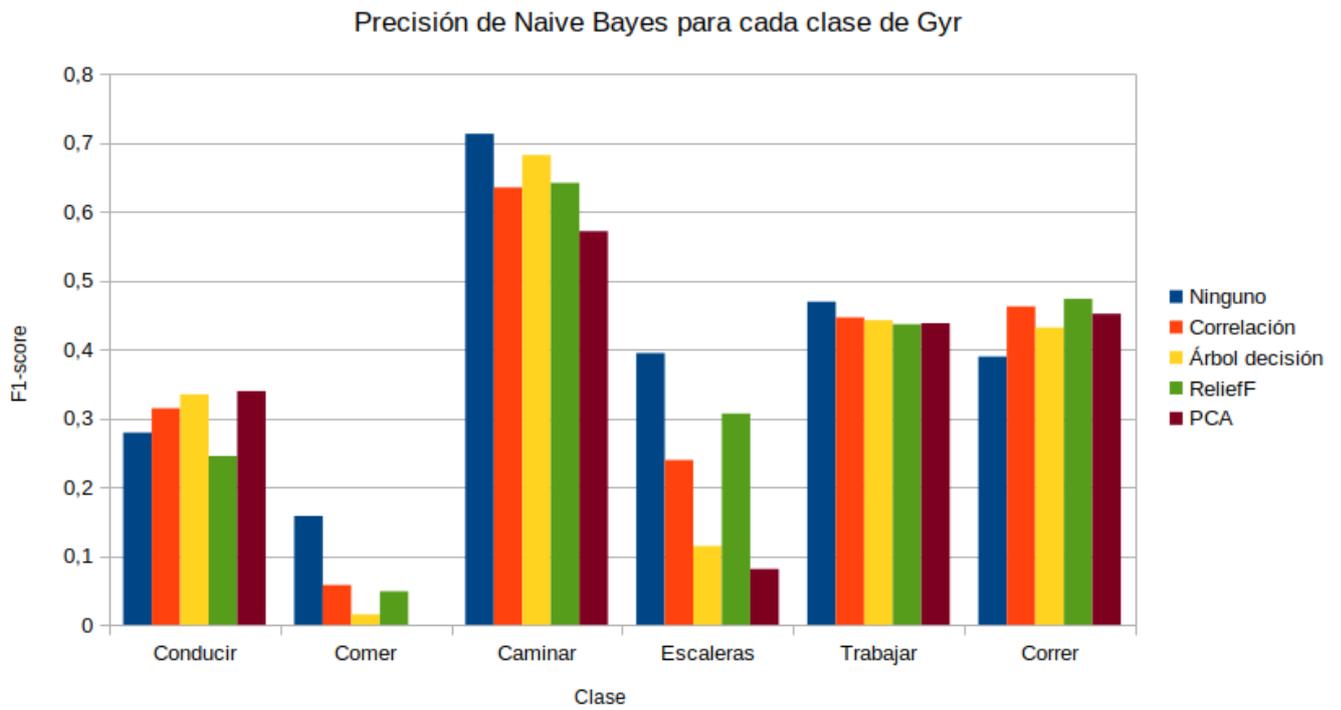


Figura 6.20: Precisión de Naive Bayes para cada clase del sensor de giroscopio

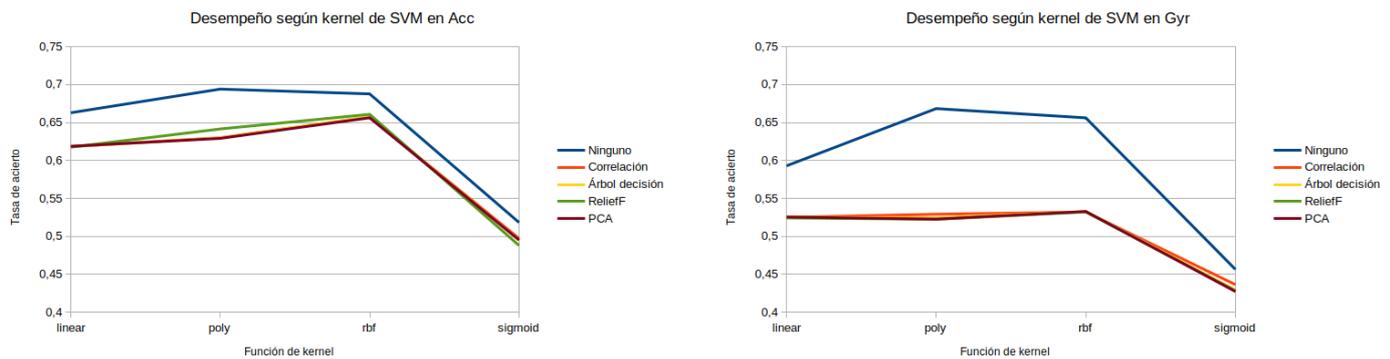


Figura 6.21: Evolución de la tasa de acierto de SVM según la función kernel para acelerómetro y giroscopio

6.4.2.4. SVM

Para obtener el clasificador óptimo mediante SVM, hemos variado la función kernel del mismo de entre cinco opciones para evaluarlas y determinar la más apropiada. Las funciones que hemos considerado son: "linear", "poly", "rbf", "sigmoid". El rendimiento de cada uno se muestra en la figura 6.21. Se alcanzan las tasas de acierto más altas para los kernels polinómicos y de función de base radial en ambos sensores.

Por otro lado, atendiendo a la precisión en la predicción de las clases presentada en la figura 6.22, observamos que a partir de los datos del acelerómetro SVM es incapaz de predecir la actividad de *escaleras* para ninguna función de kernel, y las actividades de *trabajar* y *comer* para la función sigmoide. En el caso de los datos del giroscopio, SVM no predice ni *correr* ni *comer*. Ambos clasificadores predicen con precisión alta tanto la actividad de *caminar* como la actividad de *conducir*.

Teniendo en cuenta tanto la tasa de acierto como la precisión de las clases, el kernel apropiado es el **polinómico**, obteniendo los resultados presentados en la tabla 6.5. Son unos resultados bastante mediocres en general, que no llegan al 70% de tasa de acierto. Sin embargo, sí cabe destacar la alta precisión en el

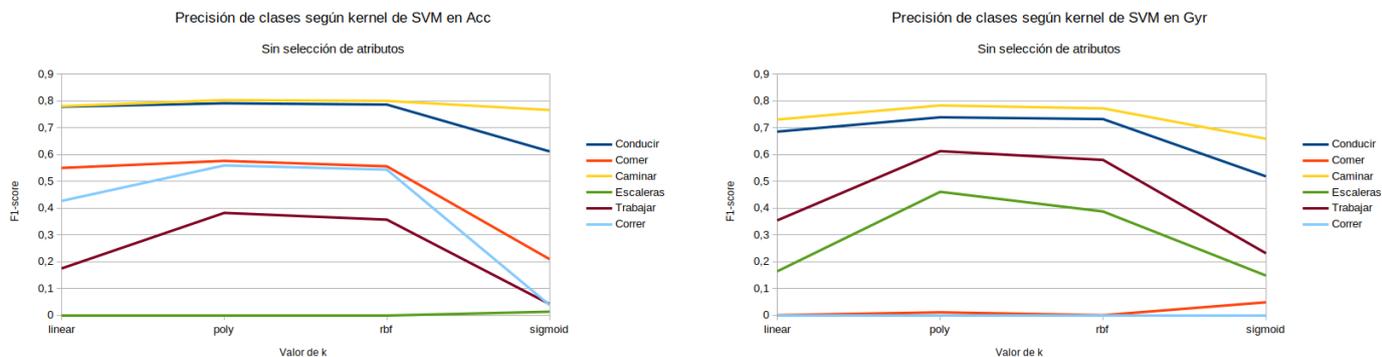


Figura 6.22: Evolución de la precisión de clase de SVM según la función kernel sin selección de atributos para acelerómetro y giroscopio

Sensor	Selección atributos	Métrica			
		Tasa de acierto	Precision	Recall	F1-score
Acelerómetro	Ninguno	0,694	0,657	0,694	0,646
	Correlación	0,630	0,551	0,630	0,571
	Árbol decisión	0,630	0,551	0,630	0,572
	ReliefF	0,642	0,581	0,642	0,593
	PCA	0,629	0,550	0,629	0,571
Giroscopio	Ninguno	0,669	0,604	0,669	0,615
	Correlación	0,529	0,338	0,529	0,410
	Árbol decisión	0,525	0,335	0,525	0,406
	ReliefF	0,522	0,332	0,522	0,404
	PCA	0,522	0,333	0,523	0,404

Tabla 6.5: Resultados de la clasificación mediante SVM con kernel polinómico

reconocimiento de las actividades de *caminar* y *conducir*.

6.4.2.5. MLP

Atendiendo solo a la tasa de acierto, el uso del perceptrón multicapa muestra unos resultados aceptables para el caso de la clasificación de datos del acelerómetro, y mediocres en el caso de los del giroscopio. Podemos ver los mismos en la tabla 6.6.

Sin embargo, atendiendo a la precisión en la predicción de las clases representada en las figuras 6.23 y 6.24, el uso del MLP es inservible para detectar las clases de *escaleras* a partir de datos del acelerómetro, y poco útil para *comer* y *correr* a partir de datos del giroscopio. En el caso del resto de clases, en general la precisión de la predicción de las mismas a partir de datos de ambos sensores es bastante aceptable.

De entre las selecciones de atributos, destaca el conjunto completo de los mismos. El resto de selecciones rinde de forma muy similar.

6.4.2.6. Clustering

Referido a los algoritmos de clustering, el comportamiento tanto de k-medias como del clustering aglomerativo ha sido prácticamente el mismo. El aprendizaje no supervisado no se ha demostrado efectivo para hacer frente a nuestro problema de reconocimiento de actividades humanas.

Hemos variado el número de clusters en ambos casos desde 6 hasta 26, obteniendo la evolución del rendimiento representada en las figuras 6.25 y 6.27. Como tendencia general, un aumento del número de

Sensor	Selección atributos	Métrica			
		Tasa de acierto	Precision	Recall	F1-score
Acelerómetro	Ninguno	0,774	0,745	0,774	0,748
	Correlación	0,673	0,610	0,673	0,625
	Árbol decisión	0,664	0,600	0,664	0,617
	ReliefF	0,658	0,606	0,658	0,614
	PCA	0,651	0,591	0,651	0,603
Giroscopio	Ninguno	0,653	0,651	0,653	0,646
	Correlación	0,598	0,602	0,598	0,593
	Árbol decisión	0,561	0,558	0,561	0,555
	ReliefF	0,608	0,616	0,608	0,607
	PCA	0,491	0,483	0,491	0,484

Tabla 6.6: Resultados de la clasificación mediante MLP

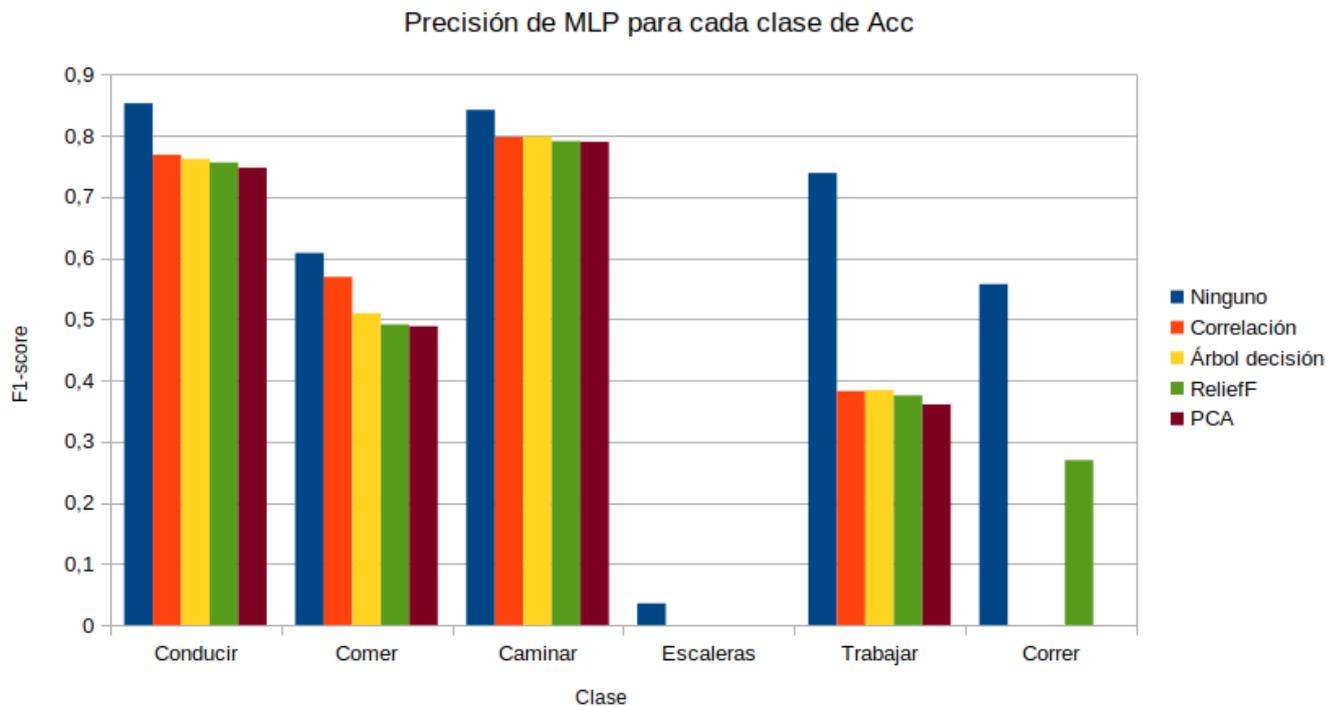


Figura 6.23: Precisión de MLP para cada clase del sensor de acelerómetro

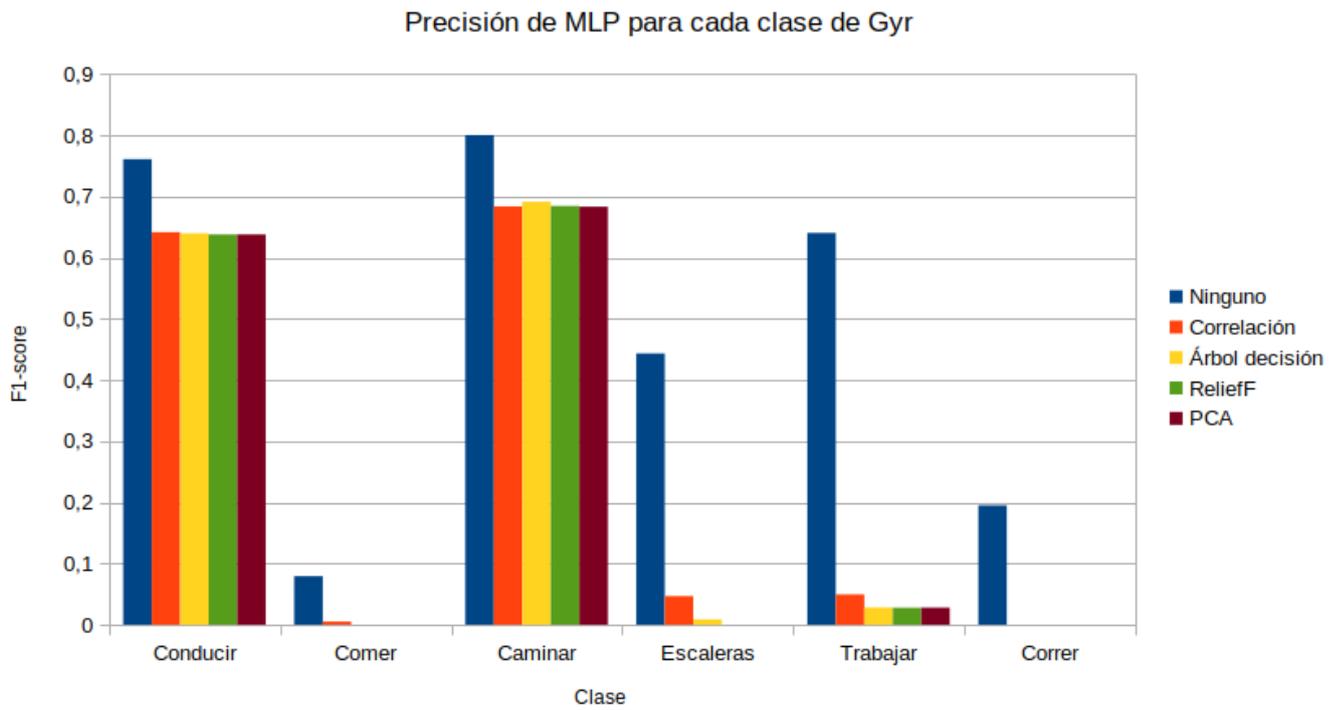


Figura 6.24: Precisión de MLP para cada clase del sensor de giroscopio

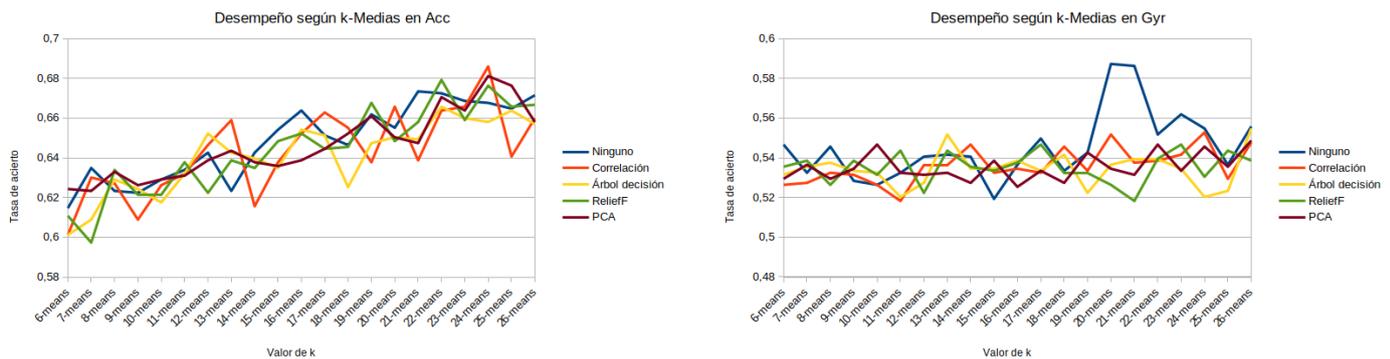


Figura 6.25: Evolución de la tasa de acierto de k-medias según el valor de k para acelerómetro y giroscopio

clusters no ha afectado al rendimiento de la clasificación de los datos del giroscopio pero si ha incrementado la tasa de acierto en la clasificación de los datos de acelerómetro.

Todas las selecciones de atributos se han comportado de formas muy similares, pero de entre ellas destaca el desempeño alcanzado con **24 clusters del análisis de componentes principales**. Si analizamos la precisión de las clases en base a esta selección (figuras 6.26 y 6.28), obtenemos las predicciones más precisas para las actividades de *caminar* y *conducir*, y las de menor precisión para *correr* (nula), *trabajar* y *escaleras*.

Los resultados globales tanto del uso de las k-medias como del clustering aglomerativo quedan reflejados en las tablas 6.7 y 6.8.

6.4.3. Comparación de clasificadores

Los algoritmos de aprendizaje que claramente han mostrado un mejor rendimiento son tanto los árboles de decisión como los k vecinos más próximos. Para poder comprobar si las diferencias que presentan estos algoritmos con respecto a los demás son realmente significativas, realizaremos una comparativa entre los

Sensor	Selección atributos	Métrica			
		Tasa de acierto	Precision	Recall	F1-score
Acelerómetro (24-means)	Ninguno	0,668	0,614	0,668	0,623
	Correlación	0,686	0,657	0,686	0,649
	Árbol decisión	0,658	0,583	0,658	0,609
	ReliefF	0,676	0,636	0,676	0,639
	PCA	0,681	0,647	0,681	0,655
Giroscopio (20-means)	Ninguno	0,587	0,503	0,587	0,537
	Correlación	0,552	0,432	0,552	0,451
	Árbol decisión	0,537	0,345	0,537	0,416
	ReliefF	0,526	0,444	0,526	0,412
	PCA	0,534	0,347	0,534	0,415

Tabla 6.7: Resultados de la clasificación mediante k-medias

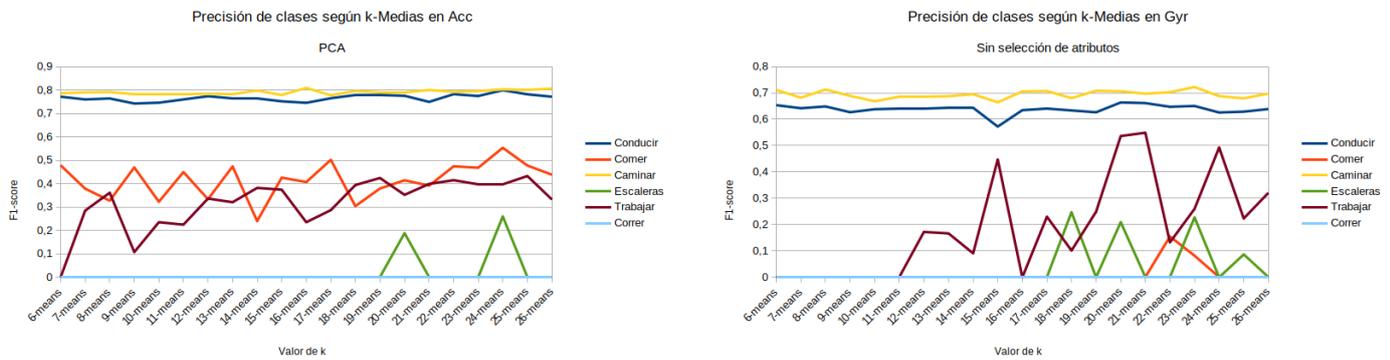


Figura 6.26: Evolución de la precisión de clase de k-means según el valor de k con PCA para acelerómetro y sin selección de atributos para giroscopio

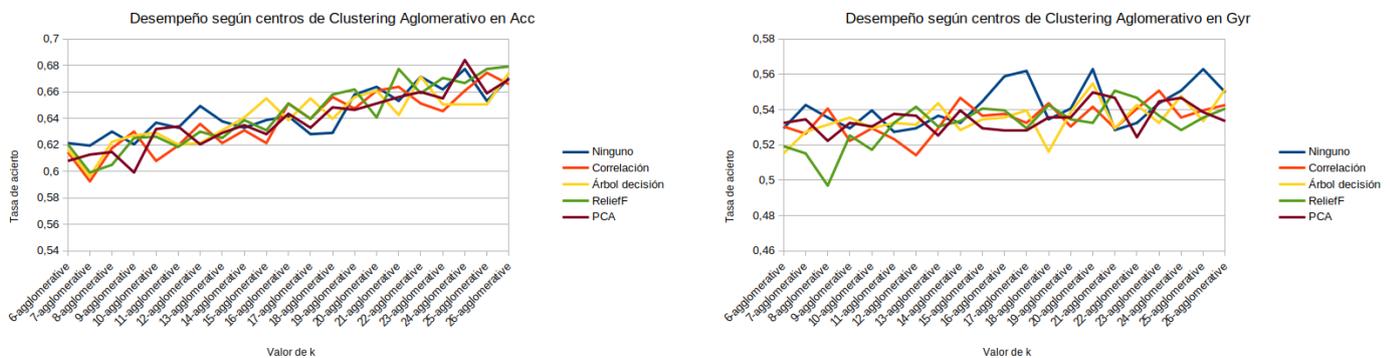


Figura 6.27: Evolución de la tasa de acierto de clustering aglomerativo según el número de centros para acelerómetro y giroscopio

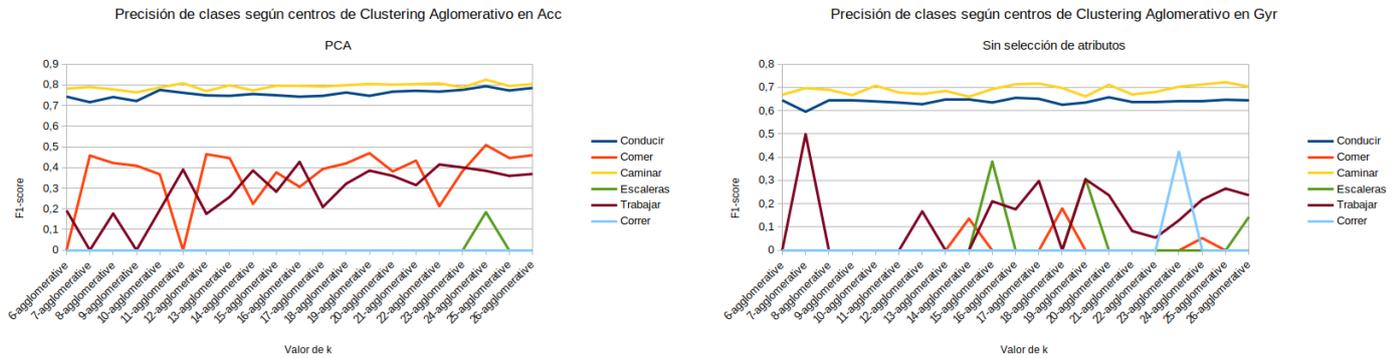


Figura 6.28: Evolución de la precisión de clase de clustering aglomerativo según el número de clusters con PCA para acelerómetro y sin selección de atributos para giroscopio

Sensor	Selección atributos	Métrica			
		Tasa de acierto	Precision	Recall	F1-score
Acelerómetro (24-agglomerative)	Ninguno	0,677	0,631	0,677	0,640
	Correlación	0,661	0,621	0,661	0,619
	Árbol decisión	0,650	0,587	0,650	0,602
	ReliefF	0,667	0,609	0,667	0,622
	PCA	0,684	0,645	0,684	0,646
Giroscopio (20-agglomerative)	Ninguno	0,563	0,466	0,563	0,469
	Correlación	0,542	0,375	0,542	0,430
	Árbol decisión	0,555	0,469	0,555	0,449
	ReliefF	0,532	0,338	0,532	0,412
	PCA	0,550	0,438	0,550	0,438

Tabla 6.8: Resultados de la clasificación mediante clustering aglomerativo

	Algoritmo	Algoritmo base: Árbol de decisión	1-NN	8-NN	Naive Bayes	SVM (poly)	MLP
Sensor							
Acelerómetro		82.60	86.59 v	83.82	72.04 *	81.89	85.20
Giroscopio		66.46	68.52	68.12	45.77 *	67.31	68.35

Tabla 6.9: Comparación de la diferencia significativa en el desempeño de la inducción de árbol de decisión frente al resto de clasificadores

clasificadores a través de un contraste de hipótesis mediante el test de Student remuestreado pareado. En el caso del giroscopio, realizaremos también la comparativa con respecto al uso de SVM como algoritmo de clasificación.

Para realizar el contraste de hipótesis emplearemos la herramienta **Experimenter** de **Weka**, configurando la experimentación mediante hold out aleatorizado 2/3 - 1/3 repetido 10 veces para el dataset completo tanto de características de acelerómetro y giroscopio interpolado a 24 Hz con ventana de 4 segundos. Emplearemos la siguiente configuración experimental de algoritmos:

- J48 -C 0.25 -M 2
- IB1
- IBk -K 8 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A weka.core.EuclideanDistance -R first-last"
- NaiveBayes
- SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -C 250007 -E 1.0"
- MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H 100

Los resultados obtenidos se muestran en las tablas 6.9, 6.10 y 6.11, donde viene representada la tasa de acierto en tanto por ciento y la diferencia significativa mediante un código de dos indicadores:

- * si la diferencia entre los resultados del algoritmo base y el algoritmo comparado es significativamente relevante.
- v si la diferencia entre los resultados del algoritmo comparado y el algoritmo base es significativamente relevante.

A modo de resumen, para el caso de los datos del acelerómetro, **1-NN** es el algoritmo que reporta un desempeño significativamente superior a la mayoría de clasificadores, con la salvedad del MLP, del que no se puede decir que sea significativamente peor. El árbol de decisión resulta significativamente superior a la clasificación a través de Naive Bayes y SVM, pero es vencido por 1-NN.

En el caso del giroscopio, tanto 1-NN como 8-NN y el árbol de decisión solamente resultan reportar un desempeño significativamente superior a la clasificación mediante Naive Bayes. Del resto de clasificadores no se puede afirmar que las diferencias resulten significativas.

Como conclusión, la elección de 1-NN es la más apropiada para el reconocimiento de actividades humanas a partir de la recogida no supervisada de datos mediante dispositivos ponibles.

Sensor	Algoritmo	Algoritmo base:	Árbol de decisión	8-NN	Naive Bayes	SVM (poly)	MLP
		1-NN					
Acelerómetro		86.59	82.60 *	83.82 *	72.04 *	81.89*	85.20
Giroscopio		68.52	68.52	68.12	45.77 *	67.31	68.35

Tabla 6.10: Comparación de la diferencia significativa en el desempeño de 1-NN frente al resto de clasificadores

Sensor	Algoritmo	Algoritmo base:	Árbol de decisión	1-NN	Naive Bayes	SVM (poly)	MLP
		8-NN					
Acelerómetro		83.82	82.60	86.59 v	72.04 *	81.89 *	85.20
Giroscopio		68.12	68.52	68.52	45.77 *	67.31	68.35

Tabla 6.11: Comparación de la diferencia significativa en el desempeño de 8-NN frente al resto de clasificadores

Capítulo 7

Conclusiones y trabajo futuro

7.1. Conclusiones

Tras el trabajo realizado, concluimos que se han cumplido todos los objetivos planteados inicialmente.

En primer lugar se ha realizado una recogida de datos no supervisada con voluntarios participantes, gracias a los cuales se ha iniciado la construcción de una base de datos dedicada al reconocimiento biométrico a partir de actividades humanas útil para el presente trabajo y para el fin futuro del reconocimiento biométrico conductual mediante actividades.

Posteriormente, se han analizado y procesado los datos capturados, identificando los problemas presentes en las señales y la solución más adecuada para los mismos. Este proceso ha incluido la limpieza de señales de frecuencias de muestreo anómalas, la interpolación a frecuencia de muestreo fijas de 12 Hz y 24 Hz, la división de las señales en ventanas de igual tamaño función de la ventana mínima de la actividad de caminar, y la extracción de características para el dominio del tiempo y el dominio de la frecuencia de las señales limpias e interpoladas para el aprendizaje automático.

En último lugar, se ha llevado a cabo el proceso de inducción de clasificadores mediante aprendizaje automático para lograr los sistemas de reconocimiento de actividades humanas, concluyendo que es posible llegar a los mismos a partir de datos capturados con dispositivos ponibles de uso comercial en un entorno natural. En concreto, el uso de los sensores de movimiento de acelerómetro y giroscopio resulta suficiente para un reconocimiento con una tasa de acierto superior al 85 % con una definición de seis clases de actividades: *caminar*, *correr*, *subir y bajar escaleras*, *trabajar*, *comer* y *conducir*. Más concretamente, el uso de acelerómetro brinda un rendimiento mayor que el del giroscopio, diferenciándose en la precisión con la que los clasificadores son capaces de predecir el valor de las clases.

El mejor clasificador para el HAR, tal y como lo hemos planteado, ha sido **k vecinos más próximos**, con unos rendimientos significativamente superiores a la mayoría, tanto en el caso de la clasificación a partir de datos de acelerómetro (1-NN) como en el caso de datos del giroscopio (1-NN y 8-NN). A su vez, hemos determinado que el clustering para esta aproximación al HAR no es una línea conveniente de actuación, debiéndose centrar los esfuerzos en la explotación del aprendizaje supervisado.

Para el mejor clasificador, la precisión de reconocimiento de las clases es relativamente alta en general, destacando los casos de *conducir* y *caminar* con un valor de f1 superior a 0,9 en el mejor de los casos. Además, el valor de f1 de las clases con menor precisión de predicción no es inferior a 0,7, por lo que se pueden considerar que en general el clasificador guarda un buen equilibrio entre precisión y exhaustividad (recall).

Para terminar, cabe destacar que las selecciones de atributos solo han superado al conjunto total de atributos en contadas ocasiones, por lo que no representan una decisión acertada, debido principalmente al carácter no supervisado de la recogida de datos y el límite inherente a los modelos de predicción inducidos a partir de los mismos.

7.2. Líneas de trabajo futuras

A partir de las conclusiones extraídas tras la minería de datos, quedan sentadas las bases para el aprovechamiento del reconocimiento de actividades humanas mediante ponibles comerciales y su aplicación en los campos que se considere más oportunos. Consideramos de especial interés el trabajo en el reconocimiento biométrico conductual y la generalización de los sistemas de reconocimiento mediante la forma de caminar, que es una aplicación completamente novedosa y con muchas perspectivas de futuro.

Por otro lado, cabe mejorar el presente trabajo, por ejemplo, a través tanto de la combinación de datos de distintos sensores como de la combinación de distintos clasificadores mediante el uso de ensembles. A pesar del desempeño comparativamente subóptimo de muchos de los algoritmos de aprendizaje para el reconocimiento de actividades humanas, un ensamblado según los valores de clases de interés podría llegar a aumentar las precisiones de las predicciones ya alcanzadas y mejorar el reconocimiento de las actividades más similares.

Además, tampoco hemos determinado el impacto de la mano de recogida en el reconocimiento de actividades, ni el impacto del paso del tiempo en la forma de realizar las mismas, por lo que conviene continuar con la obtención de datos para la base de reconocimiento conductual con más usuarios y mayor número de sesiones para poder generalizar los clasificadores inducidos.

Por último, la recogida de datos emplea sensores de relojes Wear OS. Sería interesante ampliar los sistemas compatibles con la recogida, con el fin de una aproximación mucho más realista al uso de dispositivos comerciales y con mucha mayor escalabilidad y proyección de futuro en vistas a una aplicación práctica. Se pueden mencionar como ejemplos de interés relojes y móviles con sistemas operativos Tizen OS o iOS.

Bibliografía

- [1] Lisa Schrader, Agustín Vargas Toro, Sebastian Konietzny, Stefan Rüping, Barbara Schäpers, Martina Steinböck, Carmen Krewer, Friedemann Müller, Jörg Güttler, and Thomas Bock. Advanced sensing and human activity recognition in early intervention and rehabilitation of elderly people. *Journal of Population Ageing volume*, jun 2020.
- [2] Lysa Myers. Introducción a la autenticación: cómo probar que realmente eres tú, may 2016. <https://www.welivesecurity.com/la-es/2016/05/04/autenticacion-como-probar-que-eres-tu/> (Último acceso: 28 de junio de 2021).
- [3] Michael Fairhurst. *Biometrics: A Very Short Introduction*. Oxford University Press, first edition, 2018.
- [4] Panda Security. Los 20 pins que no deberías utilizar con tu tarjeta de crédito, aug 2013. <https://www.pandasecurity.com/es/mediacenter/consejos/20-pins-no-utilizar-tarjeta-credito/> (Último acceso: 17 de junio de 2021).
- [5] José Luis Sanz. Cambia ya tus contraseñas, se han filtrado casi 9000 millones en la deep web. *El País*, jun 2021.
- [6] Vernon Haberstetzer. Los puntos débiles de sistemas de autenticación para combatir a los hackers. *TechTarget*, 2005. <https://searchdatacenter.techtarget.com/es/consejo/Los-puntos-debiles-de-sistemas-de-autenticacion-para-combatir-a-los-hackers> (Último acceso: 17 de junio de 2021).
- [7] Ecertic. Biometría del comportamiento: ¿el sistema más seguro? *Ecertic*, 2018. <https://ecertic.com/blog/biometria-del-comportamiento-sistema-seguro/> (Último acceso: 11 de mayo de 2021).
- [8] BBVA Communications. ¿qué es la biometría conductual?, may 2019. <https://www.bbva.com/es/que-es-la-biometria-conductual/> (Último acceso: 10 de mayo de 2021).
- [9] IBM. Biometría conductual: combata el fraude, no la productividad. *IBM Security*, 2016. <https://www.muycomputerpro.com/wp-content/uploads/2017/02/Biometr%C3%ADa-conductual.pdf>.
- [10] Sudeep Sarkar and Zongyi Liu. *Handbook of Biometrics*, chapter Gait Recognition. Springer, 2008.
- [11] Juan Muñoz Galán. Wearables: Análisis de dispositivos y recogida de datos en android para estudios biométricos. Master's thesis, Escuela de Ingeniería Informática. Universidad de Valladolid, 2016.
- [12] Daniel González Alonso. Estudio preliminar del uso de wearables en reconocimiento biométrico de personas. Master's thesis, Escuela de Ingeniería Informática. Universidad de Valladolid, 2017.
- [13] Irene Salvador Ortega. Reconocimiento biométrico mediante dispositivos ponibles (wearables). Master's thesis, Escuela de Ingeniería Informática. Universidad de Valladolid, 2019.
- [14] Brian Kennett. *Planning and Managin Scientific Research*. Australian National University Press, 2014.

- [15] Salvador García, Julián Luengo, and Francisco Herrera. *Data Preprocessing in Data Mining*, volume 72 of *Intelligent Systems Reference Library*. Springer, 2015.
- [16] Excellence in research for australia (era) fields of research (for). https://www.griffith.edu.au/___data/assets/pdf_file/0029/294527/ERAFoRCodes.pdf.
- [17] INCIBE, editor. *Gestión de riesgos. Una guía de aproximación para el empresario*, 2016.
- [18] Liming Chen and Chris D. Nugent. *Human Activity Recognition and Behaviour Analysis For Cyber-Physical Systems in Smart Enviroments*. Springer, 2019.
- [19] Ludwig Von Mises. *La acción humana. Tratado de economía*, chapter 1. El hombre en acción. Acción deliberada y reacción animal. Unión Editorial, fourth edition, 1986.
- [20] Jorge Luis Reyes Ortiz. *Smartphone-Based Human Activity Recognition*. Springer, 2015.
- [21] Miguel A. Labrador and Oscar D. Lara Yejas. *Human Activity Recognition Using Wearable Sensors and Smartphones*. CRC Press, 2014.
- [22] Miikka Ermes, Juha Pärkkä, Jani Mäntyjärvi, and Ilkka Korhonen. Detection of daily activities and sports with wearable sensor in controlled and uncontrolled conditions. *IEEE Transactions on Information Technology in Biomedicine*, 12(1), jan 2008.
- [23] Seungeun Chung, Jiyoun Lim, Kyoung Ju Noh, Gague Kim, and Hyuntae Jeong. Sensor data acquisition and multimodal sensor fusion for human activity recognition using deep learning. *MDPI*, apr 2019.
- [24] Stylianos Paraschiakos. Activity recognition using wearable sensors for tracking the elderly. *Springer*, 2020.
- [25] Abdulmajid Murad and Jae-Young Pyun. Deep recurrent neural networks for human activity recognition. *MDPI*, page 10, nov 2017.
- [26] Android. *Guías para desarrolladores. Sensores de movimiento*. https://developer.android.com/guide/topics/sensors/sensors_motion (Último acceso: 7 de junio de 2021).
- [27] Shivam Chaudhary. Why “1.5” in iqr method of outlier detection?, sep 2019. <https://towardsdatascience.com/why-1-5-in-iqr-method-of-outlier-detection-5d07fdc82097> (Último acceso: 1 de junio de 2021).
- [28] Discrete fourier transform (wikipedia), jun 2021. https://en.wikipedia.org/wiki/Discrete_Fourier_transform (Último acceso: 9 de junio de 2021).
- [29] J.S. Kawalec. 12 - mechanical testing of foot and ankle implants. In Elizabeth Friis, editor, *Mechanical Testing of Orthopaedic Implants*, pages 231–253. Woodhead Publishing, 2017.
- [30] La transformada discreta de fourier (d.f.t). <http://www.ehu.eus/Procesadodesenales/tema7/ty2.html> (Último acceso: 8 de junio de 2021).
- [31] Juan Luis Cano. Transformada de fourier discreta en python con scipy, sep 2012. <https://pybonacci.org/2012/09/29/transformada-de-fourier-discreta-en-python-con-scipy/> (Último acceso: 8 de junio de 2021).
- [32] SciPy.org. *scipy.stats.skew*. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.skew.html> (Último acceso: 2 de junio de 2021).

- [33] Ricardo Aler Mur. Selección y generación de atributos-i. [http://ocw.uc3m.es/ingenieria-informatica/analisis-de-datos/transparencias/SELECCION %20DE %20ATRIBUTOS.pdf](http://ocw.uc3m.es/ingenieria-informatica/analisis-de-datos/transparencias/SELECCION%20DE%20ATRIBUTOS.pdf) (Último acceso: 8 de junio de 2021).
- [34] Yash Dagli. Feature selection using relief algorithms with python example., jul 2019. <https://medium.com/@yashdagli98/feature-selection-using-relief-algorithms-with-python-example-3c2006e18f83> (Último acceso: 3 de junio de 2021).
- [35] Cristina Gil Martínez. Análisis de componentes principales (pca), jun 2018. https://rpubs.com/Cristina_Gil/PCA (Último acceso: 21 de junio de 2021).
- [36] Fernando Sancho Caparrini. Aprendizaje supervisado y no supervisado, dec 2020. <http://www.cs.us.es/~fsancho/?e=77> (Último acceso: 21 de junio de 2021).
- [37] Miroslav Kubat. *An Introduction to Machine Learning*. Springer, second edition, 2017.
- [38] Randall Wilson and Tony R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning (Kluwer Academic Publishers)*, page 264, 2000.
- [39] Joaquín Amat Rodrigo. Máquinas de vector soporte (support vector machines, svms), apr 2017. https://www.cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_support_vector_machines (Último acceso: 21 de junio de 2021).
- [40] ¿es necesario normalizar el vector de entrada para la predicción en svm? - aprendizaje automático, svm, normalización. (living-sun.com), jun 2020. <https://living-sun.com/es/machine-learning/550283-is-there-a-need-to-normalise-input-vector-for-prediction-in-svm-machine-learning-svm-normalization.html> (Último acceso: 19 de junio de 2021).
- [41] Ian H. Witten and Eibe Frank. *Data Mining. Practical Machine Learning Tools and Techniques*. Elsevier, second edition, 2005.
- [42] K-means++ (wikipedia). [https://es.wikipedia.org/wiki/K-means %2B %2B](https://es.wikipedia.org/wiki/K-means%2B%2B) (Último acceso: 21 de junio de 2021).
- [43] Joaquín Amat Rodrigo. Validación de modelos predictivos: Cross-validation, oneleaveout, bootstrapping, nov 2020. https://www.cienciadedatos.net/documentos/30_cross-validation_oneleaveout_bootstrap (Último acceso: 19 de junio de 2021).
- [44] Sai Nikhilesh Kasturi. Underfitting and overfitting in machine learning and how to deal with it !!!, jun 2019. <https://towardsdatascience.com/underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6fe4a8a49dbf> (Último acceso: 19 de junio de 2021).