

## 0.1. Captura de datos y creación del dataset

### 0.1.1. Necesidad

Dos son las principales causas que motivan la tarea de generar un conjunto de datos propios para la realización de este trabajo: su carácter de prototipo experimental y la carestía de conjuntos de datos aplicables al problema y materiales disponibles públicamente. Como mencionaremos, ambas causas suponen un fuerte peso en la decisión final.

Los conjuntos de datos que podemos encontrar disponibles públicamente que puedan asociarse directamente con el proyecto con una calidad y documentación adecuada son escasos. En la mayoría de los casos los datasets disponibles son demasiado específicos al experimento para el que fueron creados o al sistema material utilizado o con información irrelevante para este trabajo. Ejemplos usados son los datasets para el reconocimiento de actividades (? , ? , ?), (estadísticas en <https://github.com/srvds/Human-Activity-Recognition>). Casilari (? , ?) lista todos los datasets disponibles así como detalles de los mismos (actividades, sensores, captura y posición...). Si no los principales parecen ser SisFall (? , ?) (dos acelerómetros y giroscopio, caídas simuladas), MobiFall (? , ?), MobiAct (? , ?) (incluye caídas, puede ser buen ejemplo para benchmark), tFall(? , ?), DLR y ProjectGravity (? , ?)

En segunda instancia, al ser el objetivo último una implementación funcional de un prototipo, las exigencias sobre el tipo y formato de los datos quedan supeditadas a las disponibles en la plataforma de desarrollo y test. Aunque factores como la frecuencia de muestreo, resolución o valores máximos de los datos pueden ser adaptadas de los conjuntos de datos pre-existentes, prescindiremos en gran medida de estos para evitar el coste de realizar dicho tratamiento así como reducir las posibles fuentes de errores en el experimento entrenando el prototipo final con el dataset generado.

Este conjunto de datos será también usado durante la etapa de validación de los resultados con respecto al estado del arte, aunque no pueda ser utilizado para una comparación directa, si debería servir para obtener una aproximación del rendimiento.

Nombre	Fossil Gen3 Sport
Tamaño Pantalla	1,4"
Formato de Pantalla	Circular
Tipo de Pantalla	Táctil color (AMOLED)
Resolución Pantalla	454 x 454 px
Bluetooth	4.1
Wifi	802.11 b/g/n
ROM	4GBytes
RAM	512MBytes
CPU	Snapdragon 2100
OS	WearOS 2
Sensor Freq. Cardíaca	Si
Giroscopio	Si
Acclerómetro	3 ejes

Cuadro 1: Especificaciones del dispositivo de referencia

### 0.1.2. Captura de datos: AccelCapture

Con el fin de obtener un conjunto de datos para el entrenamiento, test y validación del sistema, se opta por implementar una primera aplicación para la plataforma de desarrollo.

#### Datos Capturados

Frecuencia de muestreo del acelerómetro Rango de valores Error/Resolución del acelerómetro Nombre del Portador del dispositivo Si disponibles (y no es nunca) actividad realizada tal y como es automáticamente detectada por el dispositivo

### 0.1.3. Dispositivo de captura

#### Especificaciones

### 0.1.4. Preprocesado de los datos

En un primer instante, durante el proceso de captura de datos se analiza un subconjunto con unas 200 muestras capturadas para observar las propiedades de los datos. Mediante un estudio de la periodicidad usando autoconvolución de las secuencias para buscar la posible periodicidad de los eventos. Como puede observarse en los resultados, obtenemos que las secuencias tienen poca correlación consigo mismas y por tanto a medio y largo plazo ninguna dependencia temporal. Así pues y como conviene a nuestro experimento decidimos trabajar con subsecuencias que llamaremos eventos. Estos eventos tendrán una relación de XXX muestras y como puede comprobarse al repetir los análisis de autocorrelación, si guardan una mayor dependencia temporal consigo mismo y con los eventos vecinos de la misma secuencia temporal. El objetivo de estos eventos es poder contener una acción completa (ya sea el gesto de mirar la hora, una zancada o una caída). Esta noción de evento será reciclada en el transcurso de la implementación de solución final convirtiéndose en el episodio a predecir de la red LSTM.

#### Filtrado de ruido

Los estudios siguientes demuestran que el filtrado de ruido mejora la capacidad de tratamiento posterior Tian, T.; Sun, S.; Lin, H. Distributed fusion filter for multi-sensor systems with finite-step correlated noises. Inf. Fusion 2019, 46, 128-140.

Luego, para el caso de natación Xiao, D.; Yu, Z.; Yi, F.; Wang, L.; Tan, C.C.; Guo, B. Smartswim: An infrastructure-free swimmer localization system based on smartphone sensors. In Proceedings of the International Conference on Smart Homes and Health Telematics, Wuhan, China, 25-27 May 2016; pp. 222-234.

decide que una promediado por ventana flotante de tamaño M es el que mejores resultados da:  $G_{filter} = \frac{1}{M} \sum_{i=0}^M G_i$ . (Explicado en Liu (?, ?) que usa este método con una DeepNN basada en capas CNN + 2xLSTM + Fully Connected + Softmax).