Universidad Internacional de la Rioja (UNIR)

ESIT

Máster Universitario en Inteligencia Artificial

Detección de caídas con Wearables y GRU

Trabajo Fin de Máster

presentado por: Aritz Beraza Garayalde

Dirigido por: Sonia Valladares Rodriguez

Ciudad:Paris

Fecha: 23:38

Índice de Contenidos

Índice de Ilustraciones

Índice de Tablas

Resumen

En sociedades cada vez más envejecidas aumenta la necesidad de controlar de forma contínua la salud de las personas mayores, y en concreto identificar las caídas, una de las principales causas de mortalidad en personas mayores. A este respecto existen hoy en día soluciones que sirven únicamente a este propósito, altamente intrusivas, o soluciones basadas en un reloj inteligente y una unidad de procesado independiente que reduce la autonomía. En este trabajo se presenta un sistema de detección de caídas que funciona de forma autónoma en un reloj inteligente o smartwatch ofreciendo un alto grado de precisión en un sistema no intrusivo y autónomo como es un reloj de pulsera.

Palabras Clave: Detección de caídas, RNN, LSTM, Detección Anomalías

Abstract

En sociedades cada vez más envejecidas aumenta la necesidad de controlar de forma contínua la salud de las personas mayores, y en concreto identificar las caídas, una de las principales causas de mortalidad en personas mayores. A este respecto existen hoy en día soluciones que sirven únicamente a este propósito, altamente intrusivas, o soluciones basadas en un reloj inteligente y una unidad de procesado independiente que reduce la autonomía y portabilidad. En este trabajo se presenta un sistema de detección de caídas que funciona de forma autónoma en un reloj inteligente o smartwatch ofreciendo un alto grado de precisión en un sistema no intrusivo y autónomo como es un reloj de pulsera.

Palabras Clave: Detección de caídas, Detección de Actividad, RNN, LSTM, Detección Anomalías

The need to continuously monitor elder people's health, and specially detect falls, one of the main death causes of this population, increases in an aging society. To solve this many solutions exist nowadays, but they are either based in expensive propietary one pourpose hardware which is highly obtrusive or in the capture of data through a wearable and a linked device that processes it making it less portable and not quite autonomous. This work will present a fall detection system based purely in a smartwatch, offering high precission detection in an unobtrusive and autonomous wristwatch.

Keywords: Fall Detection, Human Activity Detection, RNN, LSTM, Outliers detection

Capítulo 1

Intro

1.1. Introducción

Resumen esquemático de cada una de las partes del trabajo. Leer esta sección ha de dar una idea clara de lo que se pretendía y las concluseiones a las que se han llegado y del proceso seguido Es uno de los capítulos mas importantes

1.1.1. Motivación

Problema a tratar, posiles causas, relevancia del problema

Este trabajo resulta de especial interés en el contexto actual en el que las sociedades de los países desarrollados sufren un rápido envejecimiento de sus poblaciones. El control contínuo de diferentes parámetros de salud de poblaciones propensas a estos permite ofrecer atención de calidad a menor coste. En este contexto es especialmente importante la detección de caidas. Las caídas son la principal causa de mortaldad de personas mayores o con problemas motores que viven solos.

Existen infinidad de dispositivos para la detección de caídas en el mercado existente. Desde sistemas de propósito específico como las alarmas de detección de caídas en el baño basadas en una correa atada a la muñeca, a sistemas multipropósito basados en un dispositivo de captura y otro de procesado por separado, pasando por complejos métodos de captura y procesado de imágenes. Todos estos sistemas adolecen o bien de restricciones geográficas (funcionan o bien en una habitación o entorno geográfico limitado por el alcance de las cámaras o cobertura del enlace radio con la base de procesado) o bien resultan poco prácticos e incluso obtrusivos en una sociedad no acostumbrada a

vivir en un mundo dominado por la tecnología.

1.1.2. plantemiento del trabajo

cómo se puede resolver el problema qué se propone descripción de objetivos en términos generales

La evolución de la capacidad de cálculo de los microprocesadores y miniaturización de los sensores ha permitido generalizar y extender y popularizar el uso de dispositivos "*llevables*"(por *wearables* del inglés) al grueso de la sociedad. Estos avances se aprovechan ya parcialmente en algunas soluciones a este problema, donde se utilizan los datos capturados por el acelerómetro de una pulsera de actividad o reloj inteligente ya sea para realizar una detección simplista de una caida (una aceleración fuerte y abrupta) con altas tasas de fallos o bien para enviar este flujo de datos a un segundo sistema, normalmente un móvil, para realizar la detección. Ambas soluciones tienen sus pros y contras. La primera tiene a su favor la alta disponibilidad al aunar captura y tratamiento en la misma unidad, pero falla al usar algoritmos poco eficaces, al contrario que la segunda opción que se aprovecha de la mayor potencia de un segundo centro de cómputo para mejorar la detección a costa de una mayor complejidad en el sistema que reduzca su usabilidad y disponibilidad. Hay que tener en cuenta que el público objetivo de esta tecnología es un segmento de población con escasos conocimientos de nuevas tecnologías.

La solución propuesta es aprovechar la mejora en rendimiento de los procesadores de sistemas portables como los usados en relojes inteligentes para, al igual que en la primera solución, realizar tanto captura como tratamiento y detección de caídas en la misma unidad. La diferencia es que el algoritmo usado se base en los últimos avances en tecnologías de predicción temporal con inteligencia artificial, redes recurrentes LSTM bidireccionales con mecanismos de atención y detección de anomalías para implementar un sistema de detección autónomo en quasi-tiempo real que a la vez sea lo menos obtrusivo para el usuario final, como podría ser el simple hecho de llevar un reloj puesto.

El objetivo es por tanto implementar una solución de detección de caídas con alta precisión, siendo esta al menos comparable a la conseguida en sistemas con cómputo externo, que funcione exclusivamente en un reloj inteligente.

1.1.3. Estructura de la memoria

qué hay en cada uno de los subsiguientes capítulos

Desarrollar

Un capítulo de estado del arte, otro de objetivos (qué aportamos sobre el estado del arte).

Un capítulo sobre la descripción del problema, el dataset y su estructura (requisitos). Explicar peculiaridades sobre las señales del acelerómetro usadas tanto a nivel físico (frecuencia, valores) como fisiológico. En este sentido explicar qué se cosiderará un episodio, su longitud y magnitudes elegidas justificar dichas decisiones así como exponer todo tratamiento de datos realizado al dataset. Explicar como se consigue la estacionariedad de la señal o qué efecto tiene la estacionariedad en la decisión del tamaño del episodio.

Otro sobre la aplicación y el algoritmo de detección final usado y como presenta los resultados (IHM).

Finalmente la presentación de resultados, evaluación con respecto a las soluciones existentes (simple detección por "threshold", sistemas basados en RNN para el smartphone e incluso un modelo arima o similar no basado en redes neuronales. Incluir precisión y si es posible latencia. Explicar mecanismos de test y como se ha conseguido el conjunto de datos de test. Si da tiempo a probarlo lo suficiente como para tener un conjunto de datos estadísticamente significativo, añadir los resultados.

Resumen de la conclusión.

Capítulo 2

Contexto y Estado del Arte

Históricamente, los modelos de predicción de caídas están íntimamente ligados al estudio del reconocimiento de actividades humanas (*HAR* del inglés *Human Activity Recognition*): discernir a partir de los datos qué acción (*AQ* por *Actividad Quotidiana*, o *ADL* por sus siglas en inglés) estaba realizando el individuo. Una caída o golpe no es más que otra actividad.

2.1. Detección de actividad humana y caídas

2.1.1. Modelos matemáticos simples: cálculos basados en cotas

Los primeros acercamientos

2.1.2. Detección de caídas

2.1.3. Predicción de series temporales

Sobre cálculos basados en cotas, tenemos [Yoshida et al., 2005], Kangas[Kangas et al., 2008] lo compara con el uso ya sea de aceleración vertical o modulo de la aceleración y de nuevo obtiene resultados dispares según el tipo de caída simulada. Este índice probablemente lo usemos como algoritmo de comparación dado que no requiere informaión de postura/posición del acelerómetro.

En [Bagalà et al., 2012] se compara el rendimiento de los métodos basados en cotas sobre la aceleración registrada por un acelerómetro usando un teléfono como fuente de medida y análisis. Se desprenden los resultados: Los metodos con buenos resultados en

pruebas de laboratorio/controladas no dan buenos resultados en experimentos de uso real y viceversa (se achaca al valor de las cotas, calibradas con demasiada especificidad para el experimento en laboratorio) Los métodos con buena sensitividad tienen especificidad muy mala y al revés. Si queremos 100 % de sensitividad la especificidad puede descender hasta el 11 % (Bourke 1LFT) y para un 98 % de especificidad una sensitividad del 14 % (Kangas1D) Incluso el mejor de los algoritmos, Chen, da unos valores de especificidad 24 % y sensitividad 94 %. Claramente insuficientes para la detección de caidas. Los moviles se llevan en la mano o en la cintura.

Al ser estos métodos computacionalmente eficientes, son utilizados en sistemas embebidos donde tanto potencia de cálculo como la potencia son altamente restringidos. El acercamiento de usar relojes o pulseras para la captura de movimiento se presenta ya por [Vilarinho et al., 2015] usando tanto el smartphone como una pulsera para la captura, aunque el procesamiento se realiza en el smartphone. Comprara los resultados con iFall y Fade con resultados muy dispares según el tipo de caída aunque usa algoritmos básicos basados en modulo de la aceleración [Yoshida et al., 2005] (crean un dataset llamado project gravity solo con gente joven)

Luque[Luque et al., 2014] analiza multitud de sistemas de detección de caídas que usan un teléfono inteligente android para al menos una de las etapas (captura de datos, procesado y detección o proxy hacia un servidor externo) que hagan uso o no de terceros dispositivos. Los algoritmos estudiados son: 1- Monitorizado básico (usar módulo del vector aceleración y un threshold X) 2- Fall Index (Yoshida, T.; Mizuno, F.; Hayasaka, T.; Tsubota, K.; Wada, S.; Yamaguchi, T. A Wearable Computer System for a Detection and Prevention of Elderly Users from Falling. In Proceedings of the 12th International Conference on Biomedical and Medical Engineering (ICBME), Singapore, Singapore, 7–10 December 2005.) 3- PerFallD (Dai, J.; Bai, X.; Yang, Z.; Shen, Z.; Xuan, D. PerFallD: A Pervasive Fall Detection System using Mobile Phones. In Proceedings of the 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), Mannheim, Germany, 29 March–2 April 2010; pp. 292–297) (usa también un giroscopio) 4- iFall (Sposaro, F.; Tyson, G. IFall: An Android Application for Fall Monitoring and Response. In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2009), Minneapolis, MN, USA, 2–6 September 2009; pp. 6119–6122.)

Realmente el monitorizado básico no está tan lejos del resto de métodos e incluso

son mejores que algunas implementaciones propietarias comerciales. Recalca: alta variabilidad de los resultados segúl el tipo de caída (están demasiado optimizados para un tipo concreto, no son de uso general), también hace hincapié en los problemas de usabilidad por parte de usuarios no expertos.

2.1.4. IA para detección de caídas

Usando inteligencia artificial, Shi [Shi et al., 2020] describe un método basado en acelerómetro triaxial llevado en la cintura para predecir caidas, tratando una caida como una actividad humana más y usando métodos similares a los usados para la predicción de actividad humana (CNN). Casilari (paper ejemplar en estructura) [Casilari et al., 2020] usa una red de 4 capas convolucionales para lo mismo, y compara el rendimiento sobre multitud de datasets obteniendo que si bien se puede optimizar el rendimiento para un conjunto en concreto logrando resultados de precisión y especificidad del orden del 100 %, no es posible generar un modelo que generalice a todos los datasets [estudia también las precisiones variando el tamaño de la ventana y el threshold de potencia del modulo de la aceleración y llega a la coclusión de que ventanas de 1 segundo son las que mejor se comportan, casa con la definición de episodio de los resultados obtenidos del análisis del dataset casero. Hassan [Hassan et al., 2019] mejora los resultados, sin embargo ambos usan gran potencia de cálculo y no es aplicable al uso de sistemas embebidos. Madrano [Medrano et al., 2014] usa K-NearestNeighbours y SVM para detectar actividad y varios tipos de caídas, mostrando buenos resultados con SVM que además parece ser capaz de distinguir caídas para las que no ha sido entrenado.

Finalmente, en [Ramachandran and Karuppiah, 2020] tenemos un resumen con los últimos avances y resultados. Introduce otras fuentes de información como es el riesgo biológico, o predisposicion a la caída debido a la edad y otros factores de deterioro de la salud. Muy interesante. CUIDADO!

Un estudio similar al de Luque realiza Aziz[Aziz et al., 2016, Aziz et al., 2017] Llega a conclusiones similares respecto a la alta variabilidad de los resutlados, especialmente en los métodos basados en cotas. Sin embargo encuentra que SVM llega a generalizar incluso con tipos de caídas para las que no había sido entrenado. En la segunda obra analiza en más detalle la detección con SVM usando caídas reales y sugiere usar una combinación de métodos basados en cotas junto con técnicas de IA para obtener mejores estimadores.

2.2. GRU vs LSTM para predicción de series

Respecto al uso de LSTM para detectar anomalías en series temporales, Wang [Wang et al., 2020] usa LSTM para identificar anomalías en la señal de un motor (aunque el usa el error de recostrucción de la descomposición y recomposición wavelet de la señal como entrada a una triple red LSTM, nuestro enfoque es el contrario: usar LSTM a modo de transformada wavelet y luego comparar el error de recomposición).

Li[Li et al., 2020] usa redes con varias capas LSTM bidireccionales para la predicción de actividad humana y caidas usando acelerómetro y radar, capturando en la muñeca mientras consigue resultados comparables a las mejores técnicas de clasificación.

Qin[Qin, 2019] estudia el comportamiento de varias redes recurrentes para la predicción de la saturación de oxígeno en el agua y obtiene que las GRU son las que mejores predicciones (menos error) obtienen (Sobre LSTM e incluso RNN bidireccional). Kofi [Koudjonou and Rout, 2020] comparas LSTM y GRU para predecir mercado de valores con topologías muy variadas (número de celdas, de capas, stateless o no) y encuentra que GRU tiene mejor tasa de aciertos teniendo en cuenta el coste comutacional (y muchas veces sin tenerlo) y que no siempre tener dos capas de RNN da mejores resultados.

2.3. Pruning y técnicas para reducir complejidad de modelos

TODO

Capítulo 3

Objetivos

Puente entre el estudio y la contribución.

Debe contener:

1. objetivo general
2. objetivo específico
3. metodología de trabajo

Los objetivos deben ser *SMART*

- Specificl (objetivo claro)
- Measurable (se pueda medir el éxito o fracaso)
- Attainable (viable su conecución con el tiempo y recursos disponibles)
- Relevant (que tenga un impacto demostrable)
- Time-related (que se pueda realizar en un tiempo determinado)

3.1. Objetivo General

Un poco la descripción a grandes rasgos de qué se espera explicado al público general.

Ejemplo: Mejorar el servicio de audio gruía de un museo con una guía interactiva por voz valorada positivamente con un 4/5 al menos.

Implementar un sistema de detección de caídas que se ejecute en un smartwatch con una tasa de detección comparable a los sitemas basados en "thresholds"del acelerómetro y mucho menor ratio de falsos positivos.

3.2. Objetivos Específicos

- Desarrollar un conjunto de datos para el experimento
- Estudiar el comportamiento y características de las señales temporales del acelerómetro.
- Identificar y establecer una magnitud derivada de la lectura del acelerómetro que permita su posterior trabajo
- Desarrollar un sistema predictor basado en los puntos anteriores
- cuantificar el grado de satisfacción con el sistema obtenido y comparar con los sistemas existentes

Conjunto de objetivos más específicos alcanzables por separado. suelen sor los diferentes pasos a seguir para conseguir el objetivo general. Han de ser smart, los verbos deberian ser: • Analizar Calcular Clasificar Comparar Conocer Cuantificar Desarrollar Describir Descubrir Determinar Establecer Explorar Identificar Indagar Medir Sintetizar Verificar

3.3. Metodología de trabajo

Debe describir los pasos que se van a dar, el por qué de cada paso, instrumentos a utuilizar y cómo se van a analizar los resultados.

1. Generar un nuevo dataset [1]

   a) Los existentes se basan en detección de actividades

   b) Los pocos basados en detección de caidas se restringen a un subconjunt de caidas muy específico y muchas veces con posiciones del elemento de detección diferente de la usada en el estudio

   2- Estudiar el dataset obtenido a- comprender mejor el comportamiento de las señales capturadas b- buscar parámetros importantes para llevar a cabo la detección

2. Implementar un sistema basado en LSTM y detección de anomalías usando errores de predicción. Mecanismo atencional basado en potencia de señal de entrada?

[1] Se implementa una app para ello, y se realiza un procesado de los datos obtenidos

- Implementar una red LSTM FORWARD, y/o una FORWARD-BACKWARD, Asi como un sistema basado en threshold (usar GRU con RELU para mejorar la eficiencia de cómputo)
- Una vez entrenados los modelos con información de actividad ordinaria:
  - Implementar una app que o bien en tiempo real o cuando detecte un cierto nivel de actividad lance los modelos de predicción
  - Usar un sistema que comparando la señal real y la predicha decida si es una .ªnomalía.º no, lo más básico un RMS del error de predicción.

3. Comparar los resultados con el estado del arte
   - Entrenar el sistema usando al menos un dataset que incluya caídas.
   - Compararar con al menos métodos de cota fall index y modulo aceleración.
   - Comparar con estudios previos que usen el mismo dataset.
   - Analizar los resultados con el dataset propio.
4. Prueba en uso real. (Si da tiempo)
   - Entrenar el sistema con dataset creado.
   - Usar en interno
   - Probar en caídas simuladas
   - Si es posible conseguir ejemplos de caídas reales (app debe enviar todos los eventos registrados como caídas e indicar si han sido confirmados, rechazados o ignorados)

Capítulo 4

Identificación de requisitos

Indicar el trabajo previo realizado para guiar el desarrollo del software. Debe identificar adecuadamente el problema a tratar, contexto adecuado de uso y funcionamiento de la aplicación. Idealmente se debería realizar con expertos en la materia a tratar.

4.1. Preprocesado

4.1.1. Eliminar señal de Gravedad

En la lectura de los acelerómetros se mezcla la señal inercial a medir con lamedición constante de la gravedad. Dado que su efecto es contínuo, podemos eliminar o al menos mitigar su efecto realizando un filtrado paso alto. Para este caso realizaremos un filtro paso bajo IIR de primer orden.

$$y_n = \alpha x_n + (1 - \alpha) y_{n-1}$$

La señal obtenida equivale a un filtro con función de transferencia

$$H(z) = \frac{\alpha}{1 - (1 - \alpha) z^{-1}}$$

Donde

$$\alpha = \frac{\delta t}{\delta t + RC} f_c = \frac{1}{2\pi RC}$$

A la hora de elegir la frecuencia de corte f_c, debemos tener en cuenta que determinan que a 4Hz hay suficiente información útil como para realizar una predicción de la actividad. Dado que la señal de la Gravedad es contínua, optamos por una valor de frecuencia de corte $f_c = 1Hz$.

Despejando las eq. precedentes queda

$$f_m = 50Hz\delta t = 1/f_m = 0{,}02sRC = \frac{1}{2\pi f_c} = \frac{1}{2\pi} = 0{,}1591\alpha = \frac{0{,}02}{0{,}02 + RC} = 0,1116$$

Con este filtrado obtendremos la señal debida a la Gravedad. Si la restamos a la señal original, obtendremos la señal filtrada paso alto.

$$\vec{A_f}[n] = \vec{A}[n] - \vec{G}[n] \forall i \in [x, y, z], G_i[n] = \alpha A_i[n] + (1 - \alpha)G_i[n - 1] \vec{A_f}[n] =$$

Capítulo 5

Descripción de la herramienta desarrollada

Aportar detalles del proceso de desarrollo incluyendo fases e hitos del proceso, diagramas representativos de la arquitectura y funcionamiento, capturas de pantalla para ilustrar el funcionamiento, etc.

El desafío de consolidar en un sistema portable una aplicación autónoma de detección de caídas debe hacer frente a una serie de limitaciones.

5.1. Desafíos

El objetivo de todo modelo de detección de eventos es lograr un sistema que consiga capturar la totalidad de las realizaciones del mismo con el menos número posible de falsos positivos. En otras palabras, buscamos un sistema con una especificidad y sensibilidad de 100 %[Noury et al., 2007]. .

cias a papers, co-
re sensibilidad y es-

5.1.1. Usabilidad

El primer reto de toda aplicación es conseguir una experiencia de usuario adaptada al cliente final. De nada sirve lograr implementar una plataforma que cumpla perfectamente con todos los requisitos y objetivos funcionales si el producto resultante se utiliza.

Público objetivo

Como se ha mencionado en la introducción del trabajo, los daños relacionados con las caídas son una de las principales causas de mortaldad entre las personas mayores de 65 años . Es propio de este grupo de población la desafección por la tecnología y la carencia o desinterés por su uso. Esta condición ha de ser tenida en cuenta para el desarrollo de cualquier producto.

cita requerida

Las personas de edad avanzada suelen padecer así mismo de otras condiciones que pueden limitar su grado de movilidad, atención o memoria que impidan o reduzcan la posibilidad de adaptarse o incorporar nuevas rutinas. Los problemas motores y de percepción reducen notablemente la capacidad de mostrar información así como de interactuar con el usuario cuando se necesite una acción por su parte.

Se entiende por tanto que si se ha de realizar un producto para esta población, es requisito que sea lo menos obtrusivo posible, siendo recomendable incorporar la funcionalidad a un objeto de uso cotidiano para evitar la modificación de rutinas o la reticencia a incorporar nuevos procesos o elementos en su vida diaría. La interfaz de usuario debe ser mínima, usando un lenguaje visual que resulte familiar alejado de los estándares de las aplicaciones modernas. Así mismo, reducir o eliminar los procesos de configuración y manipulación, con un sistema que funcione al salir de la caja .

mala traducción de *ou*

Localización

Una de las decisiones con mayor impacto sobre la funcionalidad del prototipo es la elección de la posición del dispositivo de captura ya que influencia en gran medida a la capacidad de detección de caídas [Kangas et al., 2008]. Diversos estudios muestran que el mejor lugar para posicionar un sistema de medición de la aceleración para detectar caídas es la cintura, seguida de la cabeza siendo posible también usar un medidor en la muñeca[Chen et al., 2005, Kangas et al., 2008, Noury et al., 2007]. Si bien estos resultados se basan en el análisis de métodos analíticos basados en cotas, se desprende de ellos la actitud o posición del cuerpo es un buen indicador para la predicción de actividades, razón por la que realizar la captura en muñecas o tobillos, las extremidades más alejadas del tronco, sufren de mayores penalizaciones para conseguir buenas estimaciones.

Al optarse por un reloj o *pulsera de actividad* como plataforma para la implementación las opciones para posicionar la unidad de medida quedan reducida a una: la muñeca.

5.1.2. Conectividad

Hasta la fecha, los sistemas de detección de caídas se basan en arquitecturas cliente servidor para disociar el módulo de cómputo del de captura y conseguir que el dispositivo a llevar en si sea del menor tamaño posible. Este acercamiento que permite solventar el problema que derivaría de tener que llevar un obtrusivo sistema sobre si añade el problema de la necesidad de un enlace o comunicación con el módulo de cálculo. Ganamos en usabilidad pero perdemos en portabilidad.

cias a sistemas co-

El principal problema de la conectividad radica en el hecho de que a pérdida de comunicación entre ambos módulos deriva en un sistema con ninguna capacidad. El módulo de cálculo, privado de datos no es capaz de realizar ninguna detección. Por su parte el aparato de captura, por si mismo, no tiene capacidad alguna para realizar nada más que la lectura.

El sistema implementado opta por una arquitectura cliente-pesado y servidor ligero. El cliente tiene suficiente capacidad para realizar captura, cómputo y capacidad de alerta como para funcionar de forma aislada. El servidor realiza funciones de expansión de la capacidad de alerta de la plataforma, así como de distribución de actualizaciones o mejoras en modelos y parámetros, pero de ninguna manera resulta imprescindible para el funcionamiento del dispositivo portable. Este es el principio básico y director de este trabajo: implementar una plataforma autónoma de detección de caídas capaz de ejecutarse en un dispositibo llevable.

5.1.3. Rendimiento

Como en toda imlpementación, el

5.2. Plataforma

Para el servidor optamos por una arquitectura de microservicios usando la plataforma de AWS Lambda con almacenamiento en S3

Para el dispositivo móvil usamos un reloj inteligente Fossil Sport con sistema operativo WearOS y por tanto compatible con el ecosistema Android.

El la generación, entrenamiento, análisis y evaluación de modelos se realiza usando Keras/Tensorflow corriendo en la plataforma Google Colab.

5.3. Arquitectura

5.3.1. Arquitectura del sistema

5.3.2. Arquitectura de la aplicación

El sistema está compuesto por dos bloques funcionales tal y como se muestra en 5.1. Tenemos un bloque o paquete que incluye los servicios encargados del registro de datos de los sensores, gestión de entradas y notificación de eventos, implementación del algoritmo de detección y comunicación con el servidor. Este primer bloque contiene a su vez una aplicación de gestión del servicio que permite realizar tareas de mantenimiento y configuración. En un segundo bloque tenemos la interfaz del usuario principal, encargada de lanzar la aplicación, alertar en caso de caída y capturar la respuesta del usuario en caso de necesitarla.

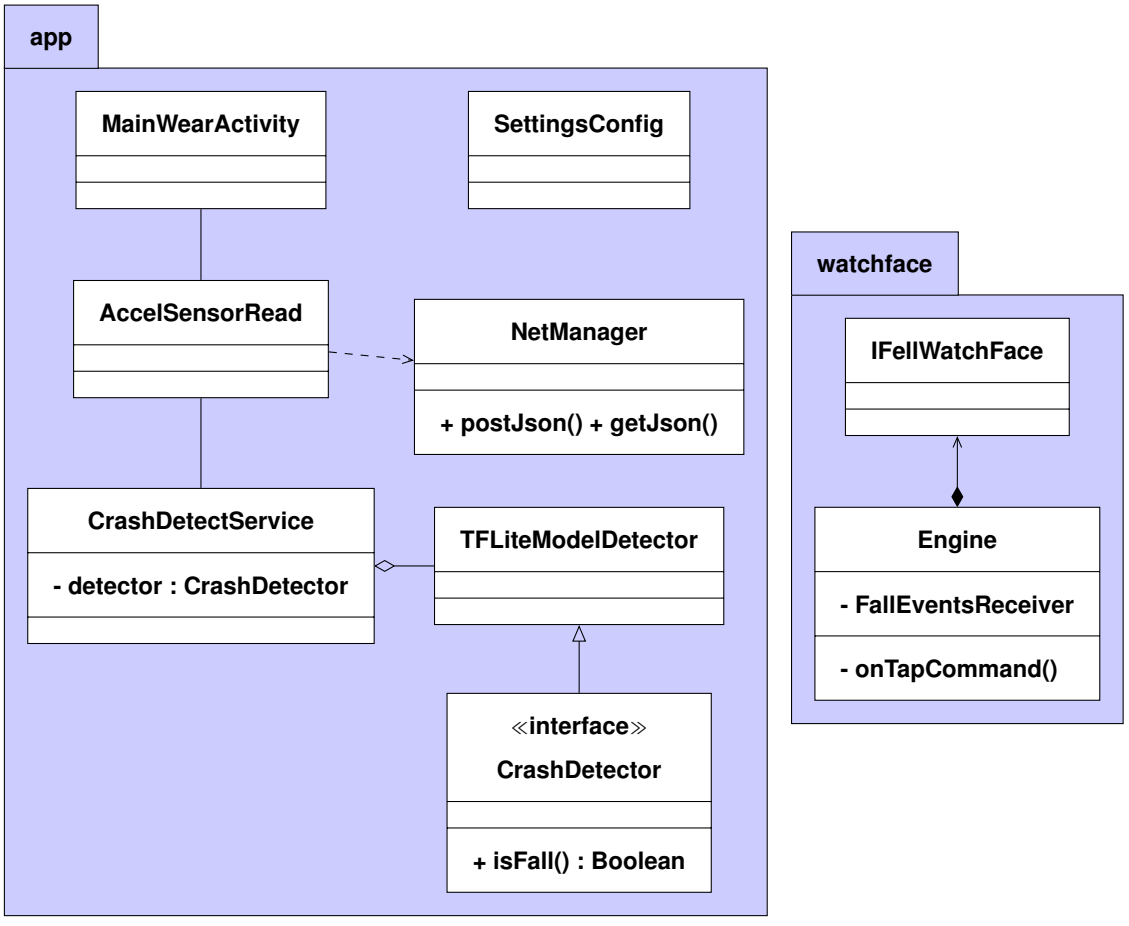


Figura 5.1: Diagrama de clases de la aplicación

Desde el punto de vista del usuario la aplicación provée dos puntos de entrada.

La interfaz principal toma la forma de una *watchface*. En *WearOS*, una *watchface* se corresponde con un tipo específico de aplicación que realiza principalmente la función de mostrar la hora al usuario. Realiza la función de .escritorio"del sistema y es por tanto el punto inicial de toda interacción del usuario con el sistema. Este acercamiento permite solventar dos problemas:

Facilita la experiencia de usuario al no requerir de ninguna acción por parte del usuario para poner en marcha la aplicación. Al convertirse en la aplicación principal del reloj con *WearOS* garantizamos que el propio sistema operativo lanzará en el arranque y mantendrá activa la actividad en todo momento.

La aplicación toma forma de un objeto cotidiano e interactúa con el usuario utilizando un concepto conocido para este: un reloj digital. La única información que se muestra al usuario es la hora (adicionalmente el propio sistema operativo sobreimprime indicaciones de batería baja, ausencia de conexión a internet y existencia de notificaciones de otros servicios y aplicaciones). De esta forma, para el usuario, el dispositivo se convierte en un objeto conocido con un uso muy extendido que de forma adicional a su función tradicional realiza el proceso de detección de caídas.

En esta aplicación se han reducido al máximo las interacciones requeridas por parte del usuario en todo momento, hasta el punto de no requerir ninguna. La aplicación funciona en todo momento com un reloj tradicional mostrando la hora de forma analógica mediante unas manecillas. En el caso de detectarse una caída o evento similar, emitirá de forma automática una serie de avisos visuales, acústicos y táctiles que podrán ser desactivados si se detecta actividad nuevamente. Aunque también existe la posibilidad de desactivarlos tocando la pantalla.

La segunda interfaz que provee la aplicación se encarga de las tareas de administración y configuración. Permite introducir un nombre del usuario y forzar el estado de funcionamiento de los servicios de captura y detección. Si bien ninguno de estos procesos es necesario, se ofrece para facilitar la puesta en marcha y prueba del sistema.

El proceso que contiene la lógica principal, descrita en la figura5.2, se encuentra en la clase *AccelSensorRead*. Este servicio es lanzado automáticamente al ejecutarse cualquiera de las dos interfaces de usuario provistas, y se mantiene ejecutándose de fondo. Se encarga de:

- Recupera configuración previa

- Configurar y leer las muestras del acelerómetro
- Realizar el primer proceso de detección (algoritmo basado en cotas)
- Lanzar el proceso de análisis usando el modelo ML y recuperar el resultado
- Alertar y notificar del evento tanto a los clientes locales como a los servidores remotos

El proceso de detección o filtrado usando el modelo implementado basado en redes recurrentes es el único que se exporta a una clase propia: *CrashDetectService*. Toma de nuevo la forma de un servicio asíncrono que es invocado únicamente cuando el modelo analítico ha detectado un positivo. De esta forma se pretende reducir drásticamente las necesidades de cómputo. Este servicio a su vez se subdivide en unos módulos que son los modelos de detección propiamente dichos. Como se aprecia en el diagrama UML de la arquitectura5.1, es la clase *TFLiteModelDetector* la que provee la cumunicación con el modelo de TFLite previamente generado y entrenado en Colab.

5.4. Implementación

5.4.1. Algoritmo

El algoritmo de detección consta de dos fases consecutivas, tal y como se muestra en el diagrama 5.2. Un primer bloque usando un algoritmo analítico basado en la variación del vector suma de la acceleración y una posterior etapa de depuración o filtrado basado en un modelo de detección de anomalías usando RNN.

Modelo Analítico

Los modelos analíticos aplicados a la detección de caídas tienen la gran ventaja de ser computacionalmente simples. Su mayor contraprestación es la dificultad de balancear las métricas de Especificidad y Sensibilidad. Al basarse en niveles o cotas, podemos aumentar la sensibilidad situando el nivel de estas de manera que la sensibilidad llegue al 100 %, como requieren los algoritmos de Bourke[Bourke et al., 2007]. Sin embargo esto afectará a la especificidad negativamente[Aziz et al., 2016].

Actualmente usa $SVTot_n - SVTot_{n-1} < 2G$ como algoritmo. Bourke en pruebas con 3.5G de límite. Sección a revisar

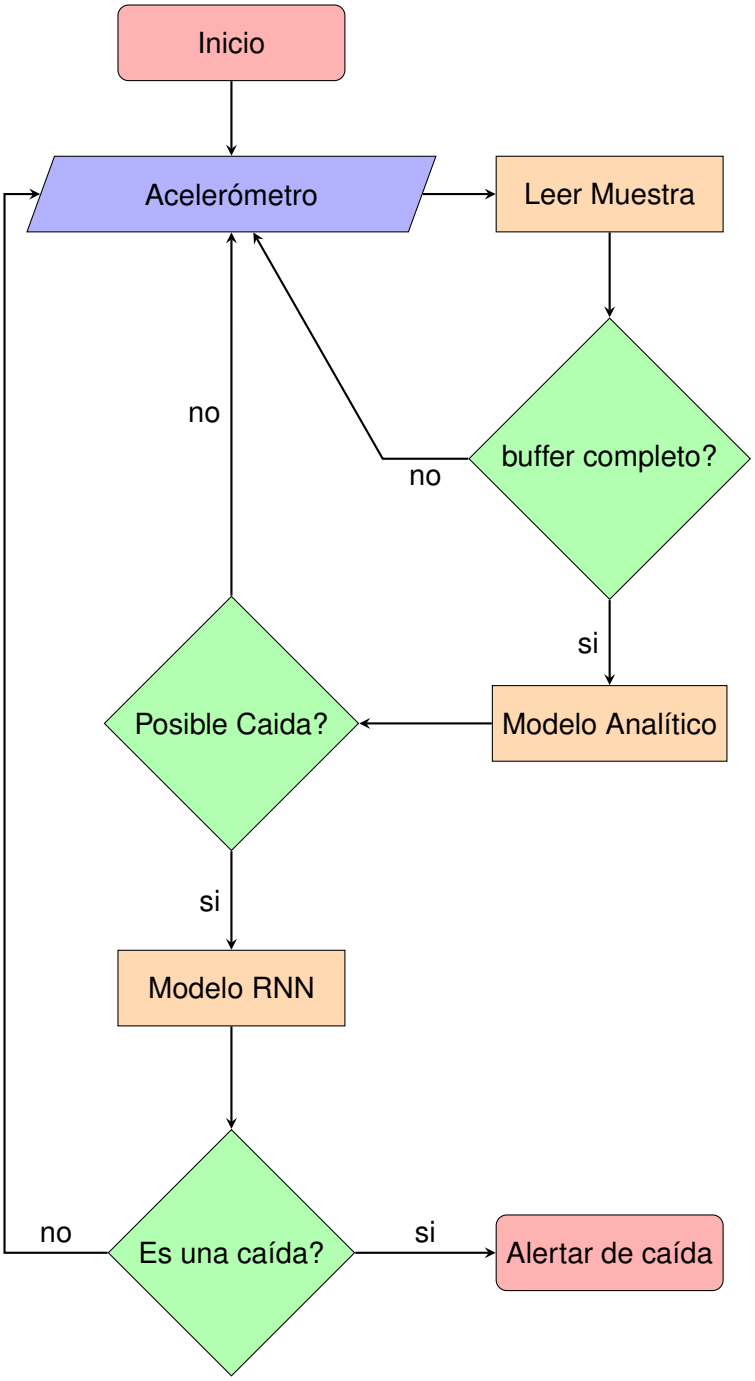


Figura 5.2: Flujo de trabajo de la aplicación de detección de caidas

En nuestro algoritmo general, el modelo analítico es similar a una capa de gestión de la atención del modelo RNN. Es por esta razón que la baja especificidad no resulta un problema ya que lo que nos interesa es que esta etapa tenga una sensibilidad del 100 %. El dispositivo captura información de la aceleración en 3 ejes con una medida máxima de $9G$ y a una frecuencia de muestreo $f_m = 50Hz$.

Como primera etapa se utiliza un algoritmo basado el el vecotor total de la aceleración ($SVTot = |\sqrt{a_x^2 + a_y^2 + a_z^2}|$) por su simplicidad de cálculo. En concreto, comparamos el valor absoluto de la diferencia de dos mediciones consecutivas de la aceleración y comparamos con un umbral de referencia.

$$ModeloAnalitico \rightarrow |SVTot_n - SVTot_{n-1}| \geqslant A_{umbral}$$

El valor de A_{umbral} se obtiene experimentalmente gracias al análisis de los datos obtenidos. Se fija en $A_{umbral} = 3G$ que resulta un buen equilibrio ya que conseguimos una cantidad reducida de eventos manteniendo un nivel bajo.

Si se sobrepasa el umbral en $t = 0$. El sistema envía instantáneamente al siguiente modelo las muestras entre $t = -225$ y $t = -25$ (4 segudnos previos al evento). En segundo plano sigue capturando muestras hasta $t = 25$. Este segundo bloque de 50 muestras (un segundo centrado en el evento) se envía también a la siguiente etapa. La decisión de separar el envío de datos a la siguiente etapa está motivada por la reducción de la latencia del sistema y obtener una clasificación o resultado con la menor dilación posible.

citar trabajos y medida
usa 3,5G, otros usan
altos, la mayoría de ca
niveles instantáneos d
periores a 6G

incluir trabajo estadíst
datos de aceleración

incluir gráfica tempora
del algoritmo

Modelo RNN

El modelo RNN comporta a su vez dos bloques. Un primer bloque de predicción y un segundo de detección de anomalías.

Predicción

Detalles del modelo pueden variar fácilmente. Actualmente usando LSTM bidireccional 50 unidades.

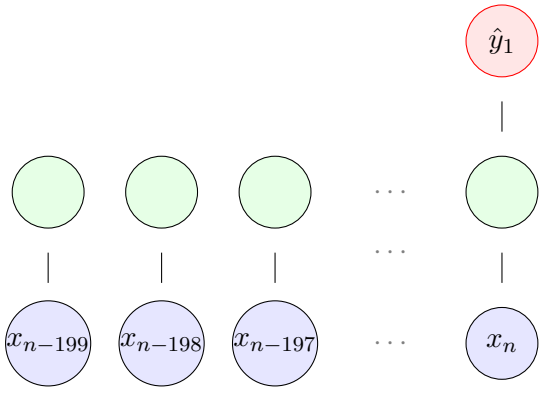


Figura 5.3: Modelo RNN

El modelo predictivo, se basa en la capacidad de las redes RNN y particularmente las

basadas en celdas LSTM y GRU para generar modelos predictivos de calidad para series temporales de señales estacionarias [Qin, 2019].

La arquitectura el modelo usa una red LSTM bidireccional con 50 unidades de memoria y una única salida. Tal y como se observa en el diagrama 5.3 a partir de la primera serie de 200 muestras recibidas el modelo es capaz de generar la predicción de la muestra siguiente.

Para conseguir la predicción de las 49 muestras restantes, debemos convertir dicho modelo en *many2many* (del inglés *de muchas a muchas*, por las muestras recibidas y las predichas). Tal manipulación se consigue alimentando iterativamente el modelo en cada paso con la predicción calculada en el paso previo $x_t = \hat{y}_t$ tal y como se muestra en 5.4

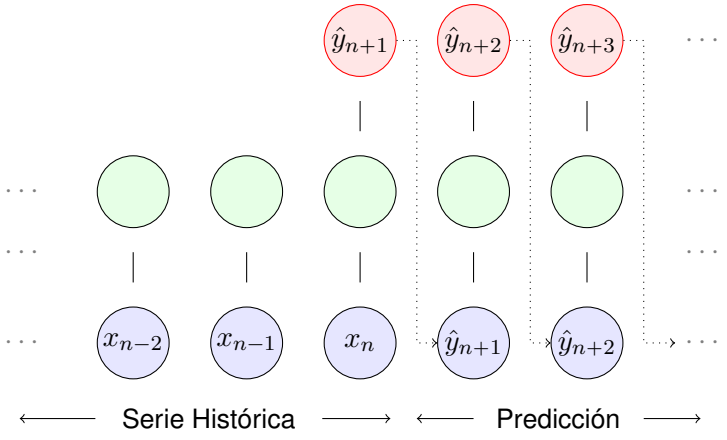


Figura 5.4: Conversión del modelo en *Many2Many*

Detección Este segundo bloque recibe por un lado la señal $SVTot(-25 : 25)$ del acelerómetro y por el segundo la predicción $S\hat{V}Tot(-25 : 25)$. Para clasificar el evento como una caída, usaremos un umbral sobre el error de predicción del modelo recurrente. La métrica de error empleada es la *Raíz del Error Cuadrático Medio* definida como:

$$RECM = \sqrt{1/T \sum_{t_1}^{t_2} |y - \hat{y}|^2}$$

Donde $t_1 = -25$, $t_2 = 25$ y $T = t_2 - t_1 = 50$.

Comparando el resultado del error obtenido con un nuevo umbral al que denominaremos umbral de detección U_d obtendremos finalmente la clasificación del evento en *Caída* o *no-Caida*. Dicho umbral se obtiene de nuevo de forma experimental.

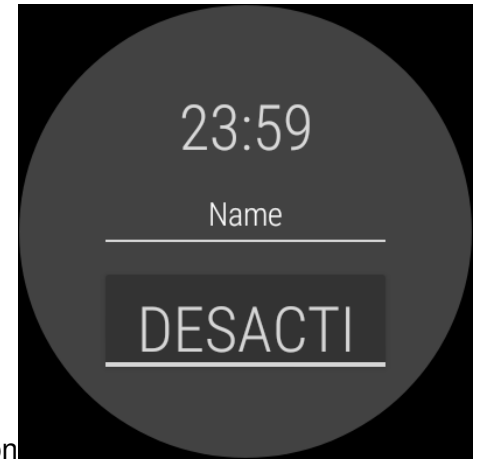
En este caso el valor utilizado se calcula mediante el análisis del RECM usado durante la validación del modelo durante el entrenamiento.

5.4.2. Optimización del modelo RNN

Quantización y prune

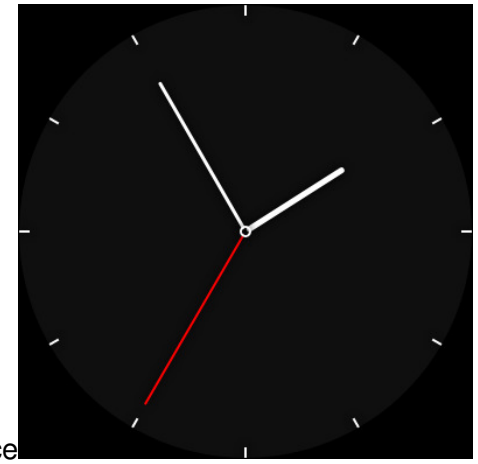
Dado que Tensorflow no permite realizar quantización ni prune sobre las capas LSTM/Gru bidireccionales estoy usando una versión propia de estas redes que implementa la interfaz prunable y por tanto debería permitirlo. Desgraciadamente todavía no he conseguido que funcione...

5.4.3. Interfaz



Aplicación de gestión

Figura 5.5: Interfaces de usuario



Watchface

Capítulo 6

Evaluación

al menos una mínima evaluación de usabilidad de la herramienta y su aplicabilidad para resolver el problema resuelto.

6.1. Evaluación del modelo

Un sistema de detección de caídas es en el fondo un clasificador con una única clase. Todos los resultados pueden por tanto agruparse en los que pertenecen o no a dicho conjunto. Con el fin de evaluar y comparar los resultados obtenidos usaremos dos métricas estadísticas:

- Sensitividad (Capacidad de identificar las caídas)

$$Sensitividad = \frac{TP}{TP + FN}$$

- Especificidad (Capacidad de discernir únicamente las caídas)

$$Especificidad = \frac{TN}{TP + FP}$$

Estas métricas son usadas habitualmente en otros trabajos similares[Noury et al., 2007, Chen et al., 2005, Bourke et al., 2007].

o de citas

Capítulo 7

Conclusiones y Trabajo Futuro

7.1. Conclusiones

7.2. Lineas de trabajo futuro

Lineas de trabajo que podrían aportar valor al TFM realizado. Perspectivas de futuro que abre el trabajo realizado. Justificar en qué modo puede emplearse la aportación desarrollada y en qué campos.

Bibliografía

[Anguita et al., 2013] Anguita, D., Ghio, A., Oneto, L., Parra, X., and Reyes-Ortiz, J. L. (2013). A public domain dataset for human activity recognition using smartphones. In *21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013*, Bruges, Belgium.

[Aziz et al., 2017] Aziz, O., Klenk, J., Schwickert, L., Chiari, L., Becker, C., Park, E. J., Mori, G., and Robinovitch, S. N. (2017). Validation of accuracy of SVM-based fall detection system using real-world fall and non-fall datasets. *PLoS One*, 12(7):e0180318. Name - Simon Fraser University; Copyright - © 2017 Aziz et al. This is an open access article distributed under the terms of the Creative Commons Attribution License: `http://creativecommons.org/licenses/by/4.0/` (the "License"), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2020-08-03; SubjectsTermNotLitGenreText - Canada; British Columbia Canada.

[Aziz et al., 2016] Aziz, O., Musngi, M., Park, E. J., Mori, G., and Robinovitch, S. N. (2016). A comparison of accuracy of fall detection algorithms (threshold-based vs. machine learning) using waist-mounted tri-axial accelerometer signals from a comprehensive set of falls and non-fall trials. *Med Biol Eng Comput*, 55(1):45–55.

[Bagalà et al., 2012] Bagalà, F., Becker, C., Cappello, A., Chiari, L., Aminian, K., Hausdorff, J. M., Zijlstra, W., and Klenk, J. (2012). Evaluation of accelerometer-based fall detection algorithms on real-world falls. *PLoS One*, 7(5):e37062.

[Bourke et al., 2007] Bourke, A., O'Brien, J., and Lyons, G. (2007). Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait & Posture*, 26(2):194 – 199.

[Casilari et al., 2020] Casilari, E., Lora-Rivera, R., and García-Lagos, F. (2020). A study on the application of convolutional neural networks to fall detection evaluated with multiple public datasets. *Sensors*, 20(5):1466.

[Chen et al., 2005] Chen, J., Kwong, K., Chang, D., Luk, J., and Bajcsy, R. (2005). Wearable sensors for reliable fall detection. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 3551–3554. IEEE.

[Hassan et al., 2019] Hassan, M. M., Gumaei, A., Aloi, G., Fortino, G., and Zhou, M. (2019). A smartphone-enabled fall detection framework for elderly people in connected home healthcare. *IEEE Netw.*, 33(6):58–63.

[Kangas et al., 2008] Kangas, M., Konttila, A., Lindgren, P., Winblad, I., and Jämsä, T. (2008). Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait & Posture*, 28(2):285–291.

[Koudjonou and Rout, 2020] Koudjonou, K. M. and Rout, M. (2020). A stateless deep learning framework to predict net asset value. *Neural Comput & Applic*, 32(14):1–19.

[Li et al., 2020] Li, H., Shrestha, A., Heidari, H., Le Kernec, J., and Fioranelli, F. (2020). Bi-LSTM network for multimodal continuous human activity recognition and fall detection. *IEEE Sensors J.*, 20(3):1191–1201.

[Liu et al., 2020] Liu, L., Hou, Y., He, J., Lungu, J., and Dong, R. (2020). An energy-efficient fall detection method based on FD-DNN for elderly people. *Sensors*, 20(15):4192. Copyright - © 2020. This work is licensed under `http://creativecommons.org/licenses/by/3.0/` (the “License”). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2020-08-26.

[Luque et al., 2014] Luque, R., Casilari, E., Morón, M.-J., and Redondo, G. (2014). Comparison and characterization of android-based fall detection systems. *Sensors*, 14(10):18543–18574.

[Medrano et al., 2014] Medrano, C., Igual, R., Plaza, I., and Castro, M. (2014). Detecting falls as novelties in acceleration patterns acquired with smartphones. *PLoS One*, 9(4):e94811.

[Noury et al., 2007] Noury, N., Fleury, A., Rumeau, P., Bourke, A., Laighin, G. O., Rialle, V., and Lundy, J. (2007). Fall detection - principles and methods. In *2007 29th Annual*

International Conference of the IEEE Engineering in Medicine and Biology Society, pages 1663–1666. IEEE.

[Qin, 2019] Qin, H. (2019). Comparison of deep learning models on time series forecasting : A case study of dissolved oxygen prediction.

[Ramachandran and Karuppiah, 2020] Ramachandran, A. and Karuppiah, A. (2020). A survey on recent advances in wearable fall detection systems. *Biomed Res. Int.*, 2020:1–17. Copyright - Copyright © 2020 Anita Ramachandran and Anupama Karuppiah. This is an open access article distributed under the Creative Commons Attribution License (the "License"), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License. http://creativecommons.org/licenses/by/4.0; Last updated - 2020-04-30.

[Reyes-Ortiz et al., 2014] Reyes-Ortiz, J.-L., Oneto, L., Ghio, A., Samá, A., Anguita, D., and Parra, X. (2014). Human activity recognition on smartphones with awareness of basic activities and postural transitions. In Wermter, S., Weber, C., Duch, W., Honkela, T., Koprinkova-Hristova, P., Magg, S., Palm, G., and Villa, A. E., editors, *Artificial Neural Networks and Machine Learning – ICANN 2014*, pages 177–184, Cham. Springer International Publishing.

[Shi et al., 2020] Shi, J., Chen, D., and Wang, M. (2020). Pre-impact fall detection with CNN-based class activation mapping method. *Sensors*, 20(17):4750. Copyright - © 2020. This work is licensed under http://creativecommons.org/licenses/by/3.0/ (the "License"). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2020-10-01.

[Sucerquia et al., 2017] Sucerquia, A., López, J., and Vargas-Bonilla, J. (2017). SisFall: A fall and movement dataset. *Sensors*, 17(12):198.

[Vavoulas et al., 2016] Vavoulas, G., Chatzaki, C., Malliotakis, T., Pediaditis, M., and Tsiknakis, M. (2016). The MobiAct dataset: Recognition of activities of daily living using smartphones. In *Proceedings of the International Conference on Information and Communication Technologies for Ageing Well and e-Health*, pages 143–151. SCITEPRESS - Science and and Technology Publications.

[Vavoulas et al., 2014] Vavoulas, G., Pediaditis, M., Chatzaki, C., Spanakis, E. G., and Tsiknakis, M. (2014). The MobiFall dataset. *International Journal of Monitoring and Surveillance Technologies Research*, 2(1):44–56.

[Vilarinho et al., 2015] Vilarinho, T., Farshchian, B., Bajer, D. G., Dahl, O. H., Egge, I., Hegdal, S. S., Lones, A., Slettevold, J. N., and Weggersen, S. M. (2015). A combined smartphone and smartwatch fall detection system. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. IEEE.

[Wang et al., 2020] Wang, R., Feng, Z., Huang, S., Fang, X., and Wang, J. (2020). Research on voltage waveform fault detection of miniature vibration motor based on improved WP-LSTM. *Micromachines*, 11(8):753. Copyright - © 2020. This work is licensed under `http://creativecommons.org/licenses/by/3.0/` (the “License”). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2020-10-01.

[Yoshida et al., 2005] Yoshida, T., Mizuno, F., Hayasaka, T., Tsubota, K.-i., Wada, S., and Yamaguchi, T. (2005). A wearable computer system for a detection and prevention of elderly users from falling. *Proc ICBM*, 12:179–182.

Apéndice A

Captura de datos y construcción del dataset

A.1. Captura de datos y creación del dataset

A.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 caracter de prototipo experimental y la carestía de conjuntos de datos aplicables al probelma 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 [Anguita et al., 2013, Reyes-Ortiz et al., 2014], (estadísticas en `https://github.com/srvds/Human-Activity-Recognition`). Casilari [Casilari et al., 2020] lista todos los datasets disponibles así como detalles de los mismos (actividades, sensores, captura y posición...), Si no los principales parecen ser SisFall [Sucerquia et al., 2017] (dos acelerómetros y giroscopio, caídas simuladas), MobiFall [Vavoulas et al., 2014], MobiAct [Vavoulas et al., 2016] (incluye caídas, puede ser buen ejemplo para benchmark), tFall[Medrano et al., 2014], DLR y ProjectGravity [Vilarinho et al., 2015]

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 comparacion directa, si debería servir para obtener una aproximación del rendimiento.

A.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 detecatada por el dispositivo

A.1.3. Dispositivo de captura

Especificiaciones

A.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

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. Cardiaca	Si
Giroscopio	Si
Aclerómetro	3 ejes

Tabla A.1: Especificaciones del dispositivo de referencia

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 caida). 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 [Liu et al., 2020] que usa este método con una DeepNN basada en capas CNN + 2xLSTM + Fully Connected + Softmax).

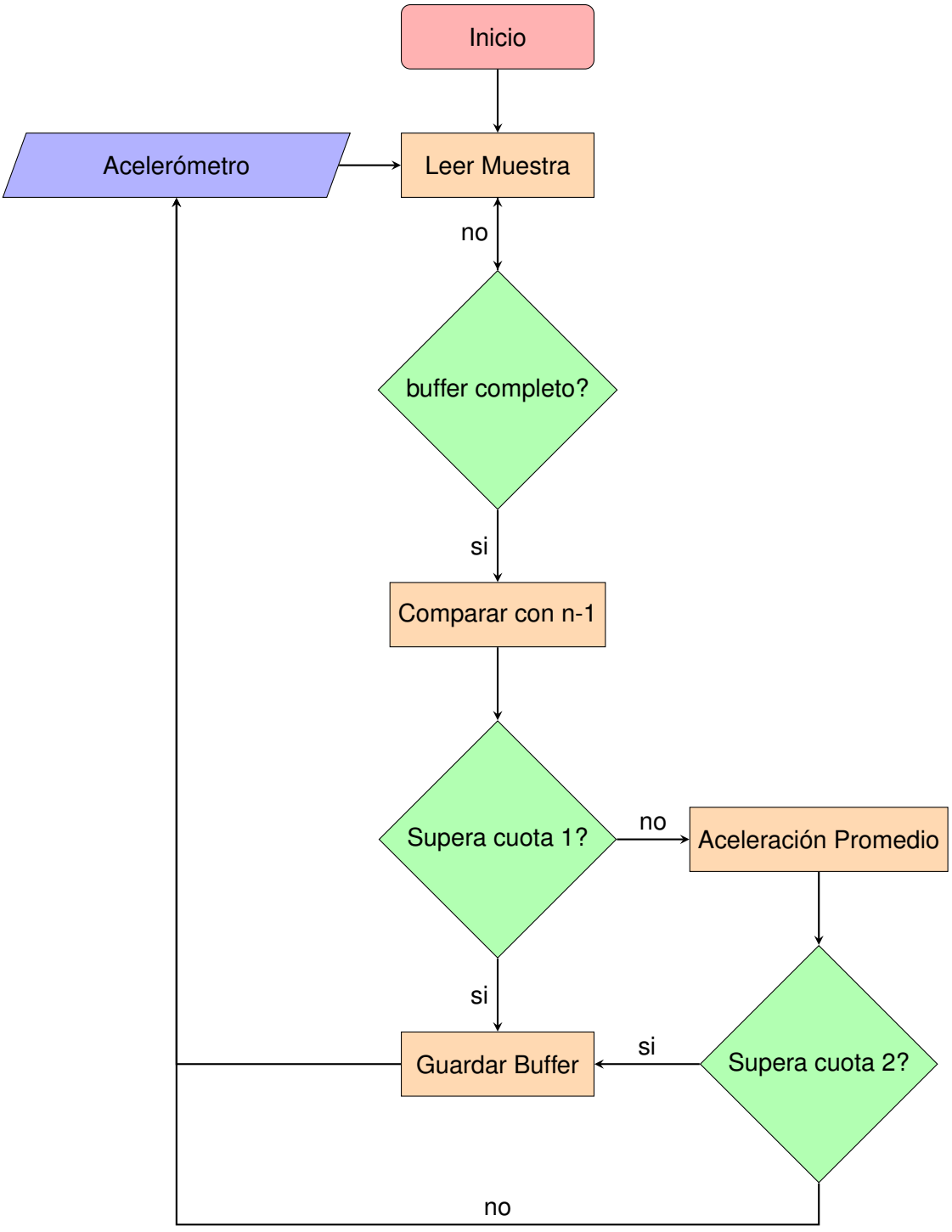


Figura A.1: Flujo de trabajo de la aplicación de captura de datos

Apéndice B

Artículo