Escola Tècnica Superior d'Enginyeria
Electrònica i Informàtica La Salle

Treball Final de Màster

Màster Universitari en Enginyeria de Telecomunicació

Sistema escalable para la monitorización acústica a tiempo real de personas parcialmente dependientes en entornos distribuidos

Alumna

Ester Vidaña Vila

Professors Ponents

Dra. Rosa Maria Alsina-Pagès
Dr. Joan Navarro

ACTA DE L'EXAMEN DEL TREBALL FI DE CARRERA

Reunit el Tribunal qualificador en el dia de la data, l'alumne

Dña. Ester Vidaña Vila

va exposar el seu Treball de Fi de Carrera, el qual va tractar sobre el tema següent:

Sistema escalable para la monitorización acústica
a tiempo real de personas parcialmente
dependientes en entornos distribuidos

Acabada l'exposició i contestades per part de l'alumne les objeccions formulades pels Srs. membres del tribunal, aquest valorà l'esmentat Treball amb la qualificació de

Barcelona,

VOCAL DEL TRIBUNAL VOCAL DEL TRIBUNAL

PRESIDENT DEL TRIBUNAL

SISTEMA ESCALABLE PARA LA MONITORIZACIÓN ACÚSTICA A TIEMPO REAL DE PERSONAS PARCIALMENTE DEPENDIENTES EN ENTORNOS DISTRIBUIDOS

ESTER VIDAÑA VILA

Máster en Ingeniería de Telecomunicaciones
Departamento de Ingeniería
La Salle
Universitat Ramon Llull

2018

Angels on Earth never have wings.

RESUMEN

Los sistemas de *Ambient Assisted Living* se han convertido en una alternativa muy potente para mejorar la calidad de vida de las personas de edad avanzada y parcialmente dependientes.

En este sentido, los sistemas de tele asistencia se han convertido en un tópico de interés para investigadores de todo el mundo. Estos sistemas normalmente recolectan datos mediante un conjunto de sensores desplegados en casa de los pacientes y concentran todos los datos en un sistema central. Sin embargo, cuando el tamaño del escenario o el número de pacientes crece, estos sistemas tienen problemas para procesar los datos recolectados y proporcionar resultados a tiempo real.

El objetivo de este trabajo es presentar una arquitectura distribuida inspirada en el paradigma del *fog computing* y analizar e identificar hasta nueve eventos acústicos que representan un comportamiento anormal o peligroso para personas parcialmente dependientes en un escenario de gran escala. Concretamente, la plataforma propuesta recolecta datos de varias fuentes de sensores acústicos sin cables e implementa un clasificador automático de dos capas de eventos acústicos para decidir cuándo el paciente se encuentra en situación de alarma.

Los experimentos se han llevado a cabo a partir de los requerimientos de la *Fundació Ave Maria*.

Palabras clave: Clasificación automática, red de sensores acústicos, vida cotidiana asistida por el entorno, monitorización acústica, sistemas distribuidos.

RESUM

Els sistemes d' *Ambient Assisted Living* s'han convertit en una alternativa molt potent per millorar la qualitat de vida de les persones d'edat avançada i parcialment depenents.

En aquest sentit, els sistemes de tele assistència s'han convertit en un tòpic d'interès per a investigadors de tot el món. Aquests sistemes normalment recol·lecten dades mitjançant un conjunt de sensors desplegats a casa dels pacients i concentren totes les dades en un sistema central. No obstant això, quan la grandària de l'escenari o el nombre de pacients creix, aquests sistemes tenen problemes per processar les dades recol·lectades i proporcionar resultats a temps real.

L'objectiu d'aquest treball és presentar una arquitectura distribuïda inspirada en el paradigma de *fog computing* i analitzar i identificar fins a nou esdeveniments acústics que representen un comportament anòmal o perillós per a persones parcialment depenents en un escenari a gran escala. Concretament, la plataforma proposada recol·lecta dades de diverses fonts de sensors acústics sense fils i implementa un classificador automàtic de dues capes d'esdeveniments acústics per decidir si el pacient es troba en situació d'alarma.

Els experiments s'han dut a terme a partir dels requeriments de la *Fundació Ave Maria*.

Paraules clau: Clasificació automàtica, xarxa de sensors acústics, vida cuotidiana assistida per l'entorn, monitorització acústica, sistemes distribuïts.

ABSTRACT

Ambient Assisted Living has become a powerful alternative to improving the life quality of elderly and partially dependent people in their own living environments.

In this regard, tele-care and remote surveillance AAL applications have emerged as a hot research topic in this domain. These services aim to infer the patients' status by means of centralized architectures that collect data from a set of sensors deployed in their living environment. However, when the size of the scenario and number of patients to be monitored increase (e.g., residential areas, retirement homes), these systems typically struggle at processing all associated data and providing a reasonable output in real time.

The purpose of this work is to present a fog-inspired distributed architecture to collect, analyze and identify up to nine acoustic events that represent abnormal behavior or dangerous health conditions in large-scale scenarios. Specifically, the proposed platform collects data from a set of wireless acoustic sensors and runs an automatic two-stage audio event classification process to decide whether or not to trigger an alarm.

The conducted experiments are based on the priorities and requirements of the Fundació Ave Maria health experts.

Keywords: Automatic classification, sensors network, ambient assisted living, acoustic surveillance, distributed systems.

PUBLICACIONES

[1] Joan Navarro, **Ester Vidaña-Vila**, Rosa Ma Alsina-Pagès y Marcos Hervás. «Real-Time Distributed Architecture for Remote Acoustic Elderly Monitoring in Residential-Scale Ambient Assisted Living Scenarios». En: *Sensors* 18.8 (2018). ISSN: 1424-8220. DOI: 10.3390/s18082492. URL: http://www.mdpi.com/1424-8220/18/8/2492.

You have to be unique,
and different,
and shine in your own way...
Live your eyeliner,
breathe your lipstick.

— Lady Gaga

AGRADECIMIENTOS

Me gustaría dar las gracias a todos aquellos que habéis hecho posible este trabajo.

En especial, quisiera agradecer el trabajo, la dedicación y el apoyo recibido por parte de Joan Navarro y Rosa Maria Alsina-Pagès. Sin vosotros nada de esto hubiera sido posible.

Finalmente, agradecer a mi familia que siempre haya estado a mi lado.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE CUADROS

ACRÓNIMOS

AAD *Acoustic Activity Detection*

AAL *Ambient Assisted Living*

ADSL *Asymmetric Digital Subscriber Line*

ANN *Artificial Neural Network*

CBR *Case-Based Reasoning*

DWT *Discrete Wavelet Transform*

GPGPU *General Purpose Graphics Processing Unit*

GPU *Graphics Processing Unit*

GTCC *GammaTone Cepstral Coefficients*

IoT *Internet of Things*

IPv4 *Internet Protocol version 4*

k-NN *K-Nearest Neighbours*

LAN *Local Area Network*

LCS *Learning Classifier System*

MDPI *Multidisciplinary Digital Publishing Institute*

MFCC *Mel Frequency Cepstral Coefficients*

QoS *Quality of Service*

ReLU *Rectified Linear Unit*

RFID *Radio Frequency IDentification*

SNE *Stochastic Neighbor Embedding*

SVM *Support Vector Machine*

TFM Trabajo de Fin de Máster

t-SNE *t-Distributed Stochastic Neighbor Embedding*

UDP *User Datagram Protocol*

WAS *Wireless Acoustic Sensor*

WASN *Wireless Acoustic Sensor Network*

WiFi *Wireless Fidelity*

ZCR *Zero Crossing Rate*

1 INTRODUCCIÓN

Resumen. En este capítulo se introduce el Trabajo de Fin de Máster (TFM). Se empieza hablando de la motivación del trabajo (es decir, por qué es importante); se continua detallando el marco de trabajo del proyecto y, finalmente, se explican los objetivos del mismo.

1.1 MOTIVACIÓN

La esperanza de vida humana cada vez es más alta en la sociedad moderna, y continuará en esta tendencia a lo largo de este siglo [1]. En la Figura 1 se puede ver la evolución de la esperanza de vida de la población desde 1910 hasta 2010.

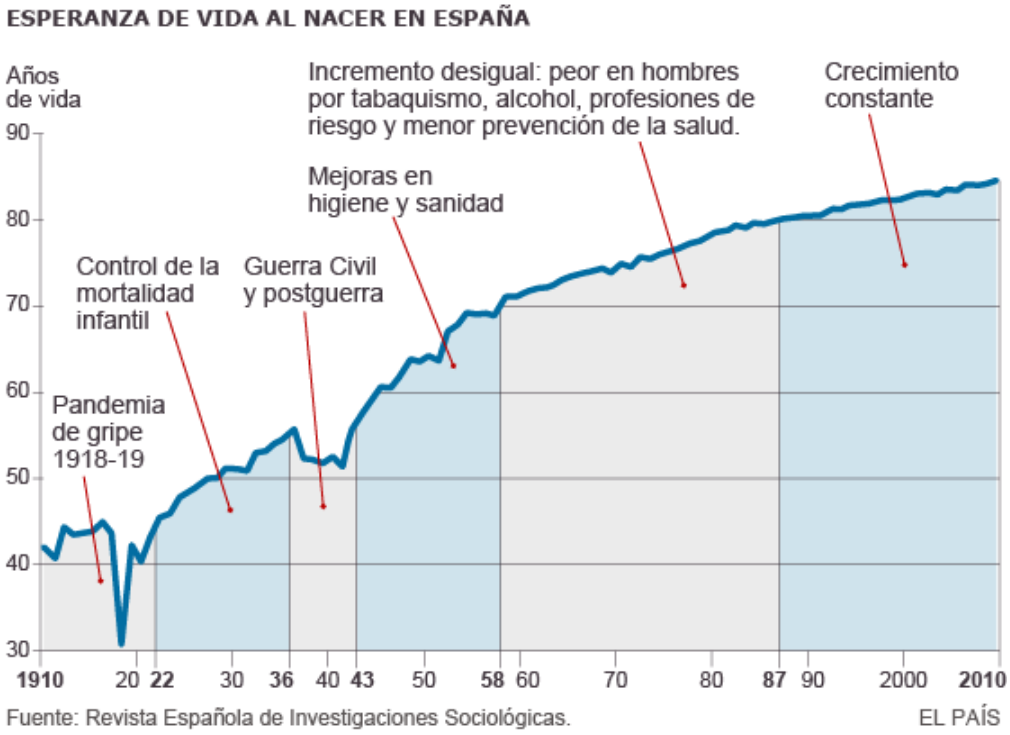

Figura 1: Tendencia de la esperanza de vida de la población [2].

Hay una fuerte razón económica en los gobiernos para motivar tanto a las personas de edad avanzada como a las personas semi-dependientes a vivir independientemente o, por lo menos, con el mínimo de servicios de cuidado posibles.

Esto minimizaría los costes públicos destinados a este colectivo y mejoraría la calidad de vida independiente de estas personas.

En este sentido, la asistencia a personas de estos colectivos en casa (o *Ambient Assisted Living* (AAL) [3]) se ha convertido en una estrategia muy popular para adaptar el entorno de los pacientes a sus necesidades específicas, lo que mejora la calidad de vida de los pacientes y optimiza los recursos para los cuidados de salud de los mismos [4].

En este sentido, las aplicaciones de AAL pueden ir desde la monitorización de las constantes vitales del paciente (la temperatura corporal, la presión sanguínea, las pulsaciones, etc.) hasta la vigilancia de los pacientes para prevenir accidentes domésticos (caídas, ataques, etc.); incluyendo la monitorización del entorno (incendios, inundaciones, etc.).

Así pues, esta necesidad de mejora en este tipo de sistemas AAL motiva la creación de un nuevo proyecto con la finalidad de poder hacer una aportación científica contextualizando el marco del trabajo en una fundación para personas parcialmente dependientes situada en Cataluña. Concretamente, este proyecto está motivado por los retos de escalabilidad que presentan actualmente los sistemas de AAL. En este trabajo, se quiere dar un énfasis especial a aquellos sistemas AAL que monitorizan a los pacientes a partir de sensores que identifican eventos acústicos destacables que indican problemas en la vida cotidiana del paciente.

1.2 MARCO DE TRABAJO

El marco de este trabajo se centra en la *Fundació Ave Maria*, una fundación para personas parcialmente dependientes que está mirando de invertir en desarrollo e investigación en el ámbito del AAL para mejorar la calidad de vida de sus pacientes.

A diferencia de otros tipos de sensores para la monitorización, los sensores acústicos sin cables (de ahora en adelante denominados *Wireless Acoustic Sensor* (WAS)) se proponen como una buena solución para facilitar la interacción entre los pacientes y las soluciones AAL ya que:

1. Son fáciles de instalar en casi cualquier tipo de infraestructura.
2. Sirven para múltiples propósitos: desde medir la actividad acústica para la vigilancia [5] hasta mejorar los sistemas de detección acústicos [6]. En algunos casos, incluso pueden servir para monitorizar el comportamiento de los pacientes [7].

De hecho, los sensores acústicos para hacer medidas también se están empleando en escenarios urbanos para mejorar la calidad de vida y salud de los ciudadanos, por lo que no están limitados únicamente a escenarios de AAL. Algunos ejemplos de uso pueden encontrarse en [8] o [9], donde se utilizan para monitorizar el ruido de tráfico en escenarios urbanos. Esto es muy importante ya que la contaminación acústica puede empeorar la salud de los ciudadanos causando, por ejemplo, que los ciudadanos no puedan dormir de noche [10],

que haya problemas en el aprendizaje [11, 12], que empeore la hipertensión o los problemas de corazón [13] y, sobretodo, que los ciudadanos no estén cómodos y se vuelvan más irascibles debido al ruido [14]. Esto es especialmente importante en los edificios públicos — escuelas, hospitales o escenarios de AAL— donde estos efectos pueden ser aún más nocivos para la salud de los habitantes ya que suelen ser más sensibles y vulnerables que la población general [15].

Por otro lado, la monitorización en casa del paciente está especialmente pensada para aquellos pacientes que tienen un nivel de autonomía alto [4] pero que necesitan constantemente estar monitorizados debido a su edad o enfermedad [16]. Entonces, para el desarrollo de estos sistemas de AAL se tienen dos restricciones fundamentales: el coste y la privacidad. Así pues, los WAS podrían cumplir con estos dos requerimientos ya que se podría hacer que respeten la privacidad de los pacientes encriptando los datos y también se podrían utilizar sensores de bajo coste.

Una vez estén los sensores desplegados en el escenario, se puede proceder a hacer detección acústica de eventos.

En la actualidad, existen varias soluciones AAL satisfactorias que se han implementado en entornos de dimensiones modestas (p. ej., laboratorios, habitaciones, etc.) y aislados [17, 18, 19, 20, 21]. Sin embargo, y aunque son un éxito, aún tienen ciertas restricciones como que el paciente tiene que estar en un área muy reducida (por ejemplo, una habitación o como máximo un piso).

Esto es debido a que hay varios retos a la hora de construir un sistema de AAL a gran escala. En primer lugar, la cantidad de datos a procesar incrementa proporcionalmente a la superficie o número de pacientes a monitorizar, lo que requiere dispositivos potentes a la hora de procesar y almacenar toda la información y, además, debe hacerse en un tiempo razonable [22].

Por otra parte, todos los datos que los sensores, pacientes y médicos están continuamente intercambiándose deben transpotarse por redes de comunicación y tecnologías heterogénas (como Bluetooth, *Wireless Fidelity* (WiFi), Internet, etc.), que normalmetne tienen problemas a la hora de cumplir con los niveles de calidad de servicio o *Quality of Service* (QoS) necesarios en aplicaciones de AAL [23].

Sin embargo, los últimos avances en sistemas distribuidos (*fog computing*, *parallel computing*, etc.) y en el procesado de datos (*data mining*, *cloud computing*) pueden contribuir en expandir los sistemas de AAL que existen en la actualidad para que sean desplegados en escenarios a gran escala.

1.3 OBJETIVOS

El objetivo principal de este proyecto es proponer una plataforma de adquisición, procesado y análisis de datos acústicos que permita desplegar servicios de AAL en entornos a gran escala.

A tal efecto, a continuación se desglosan los objetivos de este TFM teniendo en cuenta el marco de trabajo del mismo:

1. En primer lugar, se propone hacer un estudio del estado del arte que existe actualmente en el ámbito del AAL para ver qué tecnologías se están utilizando y qué proyectos de investigación se están llevando a cabo.

2. En segundo lugar, se quiere analizar qué tipo de sonidos domésticos pueden dar información sobre el paciente para ver si todo va bien o si hay que generar una alarma. Cuando se sepan, se tendrá que confeccionar un *data set* para poder hacer la evaluación experimental.

3. En tercer lugar, se propone analizar qué parámetros acústicos (o *features*) modelan mejor los sonidos domésticos dentro de un conjunto finito de *features*.

4. A continuación, se propone crear un sistema de aprendizaje automático que permita clasificar los distintos tipos de sonidos acontecidos en el ámbito del AAL.

5. Por otra parte, también se propone diseñar una arquitectura distribuida que permita que el proyecto de AAL pueda cubrir superficies de gran escala y así satisfacer las necesidades de la *Fundació Ave Maria*.

6. Finalmente, se pretenden consolidar y ampliar los conocimientos actuales en el ámbito de procesado de audio, programación, minería de datos e inteligencia artificial; así como también se quieren ampliar los conocimientos en arquitecturas distribuidas y *fog computing*.

1.4 ESTRCUCTURA DE LA MEMORIA

A continuación, se presenta cómo están organizados los capítulos que conforman esta memoria:

Capítulo 2. Estado del arte: Introduce los avances tecnológicos en el ámbito del AAL. En él se revisan distintas publicaciones de investigadores de todo el mundo con relación al tema de este proyecto y se investiga acerca de qué tecnologías han proporcionado mejores resultados.

Capítulo 3. *Fundació Ave Maria*: Detalla a qué se dedica la fundación en la que se quiere implementar el sistema y las necesidades que se encuentran en la misma. Se estudia la estructura de los edificios para poder así poder diseñar la arquitectura del proyecto más adelante.

Capítulo 4. Eventos acústicos a considerar: Explica detalladamente qué tipo de eventos se van a considerar para llevar a cabo la generación automática de alarmas para mejorar la calidad de vida de los pacientes de la fundación y ayudar a los cuidadores de los mismos a saber cuándo hay situaciones que podrían generar problemas.

Capítulo 5. Clasificación: Se presenta la última — y definitiva — versión del clasificador propuesto para el caso de uso de la Fundación Ave María. Este es un clasificador de dos capas que se implementa con una ANN, lo que permite obtener porcentajes de acierto muy elevados al variar los parámetros del clasificador y un *Case-Based Reasoning* (CBR) que permite recoger la información de varios sensores acústicos, así como tener una capa de memoria para poder considerar cómo varían los eventos a lo largo del tiempo.

Capítulo 6. Arquitectura del sistema: Se presenta la arquitectura definitiva propuesta para instalar en los escenarios de la *Fundació Ave Maria* teniendo en cuenta el clasificador.

Capítulo 7. Conclusiones y líneas de futuro: Se reflexiona acerca de lo aprendido en el TFM y se presentan las principales conclusiones a las que se ha llegado así como el impacto científico que han tenido. También se habla de las posibles líneas de futuro que ha abierto este TFM y se hace un estudio del coste económico/temporal que ha supuesto el mismo.

2

ESTADO DEL ARTE

Resumen. En este capítulo se introducen los avances tecnológicos en el ámbito de la tele-asistencia mediante sensores acústicos que se están llevando a cabo (o se han llevado a cabo en el pasado). Se habla tanto de avances en relación a la *feature extraction* como al *data minig* y *machine learning*. A partir del análisis de los resultados que han obtenido otros investigadores se pretende tener un punto de partida desde el cual plantear un escenario en el que se pueda implementar una plataforma de sensores acústicos para la monitorización y asistencia de personas parcialmente dependientes.

Actualmente, los avances en la investigación y desarrollo de tecnologías para la monitorización y asistencia para adultos en sus propias casas (de ahora en adelante nombradas como tecnologías para el AAL) están permitiendo a las personas parcialmente dependientes poder seguir teniendo un estilo de vida más autónomo sin poner en riesgo su seguridad. Concretamente, el despliegue de redes de sensores acústicos sin cables (de ahora en adelante nombradas *Wireless Acoustic Sensor Network* (WASN)) están permitiendo desarrollar estas estrategias de monitorización en entornos interiores de forma no invasiva. Por otra parte, las arquitecturas de *fog computing* también han contribuido en la reducción de coste de computación y almacenamiento de datos en la nube. Así pues, en este capítulo por una parte se va a explicar el estado del arte de las WASN y las arquitecturas de *fog computing* para las aplicaciones de monitorización de salud de personas parcialmente dependientes y, por otra parte, se explicarán las limitaciones del homeSound [21] , proyecto predecesor a la realización de este TFM.

2.1 PROYECTOS DE INVESTIGACIÓN EN EL ÁMBITO DE AAL

El concepto de casa inteligente es fundamental para la realización de proyectos en el ámbito de AAL. Habitualmente, se describe una casa inteligente o *smart home* como una casa normal y corriente con sensores encargados de recolectar datos o automatizar tareas cotidianas. En el ámbito de la asistencia ambiental en el hogar, con una casa inteligente se puede obtener tanto información del entorno del paciente como información relacionada con la vida del mismo (por ejemplo, el movimiento del paciente dentro de casa, patrones de comportamiento determinados, actividades, etc.)[19, 4]. Para llevar a cabo la recolección de datos se pueden usar sensores de distintos tipos, siendo comunes los sensores de captura de movimiento, RFID, cámaras, sensores de ultrasonidos o micrófonos [24]. A continuación se citan varios proyectos que se han llevado

a cabo en el ámbito de la asistencia en el hogar, muchos de ellos englobados dentro del proyecto *Assisted Living Joint Program* [25].

En primer lugar, el proyecto *Aware Home* [26] utiliza una gran variedad de sensores, desde sensores diseñados especialmente como *smart floors* (suelos inteligentes que pueden ubicar a una persona según sus pasos) hasta cámaras de vídeo o clásicos sensores de ultrasonidos. Estos sensores, junto a robots sociales, monitorizan y ayudan a adultos de avanzada edad.

Otro tema de interés en los últimos años en el campo de *AAL* es el análisis del comportamiento y la monitorización de actividad de los pacientes. La principal hipótesis es que observando la actividad doméstica de una persona se pueden llegar a detectar enfermedades en su etapa más precoz. Entre estos hábitos podrían destacar: cuántas veces se ducha el paciente, cuando entra o sale de la cama, qué uso hace del teléfono, cuándo se viste, sus hábitos a la hora de preparar comida, sus hábitos de entrar y salir de casa, cada cuánto toma medicación o cada cuánto hace tareas del hogar. Los cambios en los patrones de comportamiento podrían indicar que la salud del individuo ha empeorado.

En este ámbito destaca el proyecto *Project House*[27], que presenta un sistema para hacer un seguimiento de la actividad doméstica en la vivienda utilizando pequeños sensores que pueden ser instalados en casa rápidamente. El objetivo de este proyecto es evitar utilizar cámaras y micrófonos, que requerirían técnicas de procesado del señal para poder obtener datos significativos y, además, podrían parecer invasivos desde el punto de vista de los pacientes.

Por otra parte, también destaca el proyecto *Gloucester Smart House* [28], un sistema de tele-asistencia basado en la monitorización del estilo de vida de los pacientes. En este sistema, se mantienen los componentes clásicos de la teleasistencia (botón del pánico que aprieta el usuario cuando se encuentra en una situación en la que necesita ayuda como podría ser una caída), pero además se añaden pequeños sensores en casa del paciente que permiten saber por dónde se mueve mediante rayos infrarrojos, si las puertas se abren o se cierran, la temperatura de casa, etc. Recolectando éste tipo de información continuamente han diseñado un sistema que genera una alarma de forma automática cuando se detecta que algo no va bien y se avisa mediante mensajes de voz a los cuidadores de que el paciente necesita asistencia.

Otro proyecto muy interesante y original es el presentado por el *Elite care project* [20]. En este caso, los creadores del proyecto parten de la hipótesis que el sueño es un factor muy importante para determinar la salud de un paciente, puesto que un patrón irregular de sueño o una actividad excesiva durante el mismo podría ser síntoma de una enfermedad o una anomalía neurológica. Así pues, presentan un sistema que mediante sensores de peso situados en las esquinas de la cama del paciente, monitoriza la actividad nocturna de un paciente para detectar irregularidades en los patrones de sueño. Estos sensores tampoco son invasivos para los pacientes, ya que no sienten un nivel de intrusión muy elevado (como el que podrían notar con, por ejemplo, el uso de cámaras instaladas en sus dormitorios).

Finalmente, el proyecto *Ubiquitous Home Project* [18]. En este proyecto se ha creado una casa completamente equipada con diferentes sensores como micrófonos, cámaras, sensores de presión en el suelo, sensores de *Radio Frequency IDentification* (RFID) y sensores infrarrojos. Además, cuenta también con un robot con el que los usuarios interactúan mediante voz. Este robot puede hacer distintas funcionalidades para ayudar a los habitantes en su día a día (por ejemplo, los habitantes pueden pedirle que ponga la televisión y les muestre programas sugeridos, pueden pedirle ayuda para llevar a cabo una receta de cocina, etc.).

2.2 REDES DE SENSORES ACÚSTICOS *wireless* PARA LA TELE-ASISTENCIA

Una WASN puede definirse como una red de nodos de micrófonos sin cables distribuidos espacialmente en entornos interiores i/o exteriores. A la hora de diseñar una red de este tipo, es importante tener en cuenta: la escalabilidad de la misma, los retrasos que pueden sufrir las señales por la red, la sincronización entre los distintos nodos de la red y la decisión de dónde se realizan los cálculos de procesado (se podrían realizar de forma local o se podrían hacer de forma remota en entornos de *cloud*)[29].

La popularidad de estos sistemas en entornos de monitorización para sistemas de AAL está incrementando últimamente debido a que los usuarios perciben las redes de micrófonos como sensores prácticamente no invasivos. Además, son muy competitivos en términos de coste ya que los sensores de la red pueden ser *low-cost*, así como el coste computacional es mucho más bajo que si se trabajara con imágenes. Un ejemplo de estos sensores ha sido presentado en [30].

Una aplicación muy común para las redes de sensores acústicos es la teleasistencia, ya que permiten localizar a un individuo dentro de casa sin la necesidad de que haya otra persona supervisando. Esto se haría situando los sensores repartidos por la casa del individuo y un sistema central encargado de procesar las señales que van captando los micrófonos. Así, además de poder localizar dónde se encuentra el individuo, se podría llevar a cabo lo que se denomina comúnmente Detección de Actividad Acústica (*Acoustic Activity Detection Acoustic Activity Detection* (AAD))[29]. Para poder llevar a cabo esta detección de actividad, lo primero que habría que hacer es detectar eventos acústicos relevantes del ruido de fondo que pueda tener una casa (un ejemplo de evento relevante podría ser un cristal roto o un grito, que podría indicar que la persona está bajo una situación de peligro). En una primera instancia, esto podría hacerse determinando un nivel de energía como umbral, y definir que aquello que tenga más energía que el umbral sea catalogado como evento anómalo [6].

Al igual que en este proyecto, en [31] utilizan redes de sensores en casa para poder monitorizar a las personas mayores o parcialmente dependientes que quieren llevar una vida más autónoma viviendo en su propio hogar. Esto pre-

senta un gran reto en términos de detección de eventos acústicos; ya que es muy difícil distinguir cuándo un evento está afectando realmente a una persona o cuándo simplemente es un ruido de fondo sin importancia como podría haber en cualquier casa. Para intentar detectar con seguridad estos eventos, en [32] utilizan los coeficientes MFCC (coeficientes cepstrales basados en la escala Mel, explicados más adelante en esta memoria), el *Zero Crossing Rate* (ZCR) y la *Discrete Wavelet Transform* (DWT) —entre otros parámetros de *feature engineering*—, junto a una SVM o un *K-Nearest Neighbours* (k-NN) con 5 vecinos como herramientas de decisión de clasificación para lograr conseguir una F1 superior al 90 % de acierto. La puntuación F1 también será explicada más adelante en esta memoria.

Por otra parte, en [30] se presenta un sistema de detección de eventos basado en una plataforma de bajo coste que es capaz de grabar y procesar los sonidos en la misma casa del paciente. En el trabajo, presentan el porcentaje de acierto y el consumo de energía del sensor cuando adquiere y procesa los datos. Se puede ver que la adquisición y el procesamiento de los datos puede llevarse a cabo simultáneamente sin comprometer la frecuencia de muestreo deseada.

En [33], se explica un algoritmo basado en el análisis de vectores de características independientes para poder seguir el movimiento de elementos en entornos naturales. A pesar de que se utiliza para localizar coches, esta aproximación también podría ser válida para entornos interiores donde se tienen que localizar a personas.

Centrándonos más en la descripción de las aplicaciones, en [34] se describe un sistema de detección de caídas orientado específicamente para las personas mayores que viven —o pasan gran parte del tiempo— solas.

Otro proyecto del ámbito AAL es el proyecto CIRDO, [35]. En este caso, además de utilizar audio se procesan *streams* de vídeo a tiempo real. Para lograr hacerlo, hacen uso de una *General Purpose Graphics Processing Unit* (GPGPU). Para respetar la privacidad de los individuos, además, garantizan que no hay interacción humana en los datos recogidos en casa, estos se procesan de forma totalmente autónoma por lo que nunca nadie visualiza las imágenes que capturan las cámaras. Aún y así, las personas siguen sin estar cómodas al saber que tienen cámaras en casa. En este caso también tratan de detectar las caídas de las personas mayores.

En [7], se detalla un estudio preliminar para el reconocimiento de actividades (predefinidas) en la actividad diaria de las personas mayores utilizando una WASN de bajo consumo. En este caso, la red está compuesta tanto por micrófonos como por sensores de ultrasonidos.

Otro proyecto relacionado con el ámbito se describe en [16], cuyo objetivo principal es el de detectar enfermedades mentales (en sus fases primarias) en personas mayores que viven solas en casa. Al igual que en otros proyectos, en este caso se utilizan tanto sensores acústicos como cámaras de vídeo. Finalmente, en [36] se distinguen hasta 21 tipos distintos de sonidos que ocurren en la cocina para poder modelar el comportamiento de los habitantes de la casa.

2.3 ARQUITECTURAS DE FOG COMPUTING PARA PROYECTOS DE AAL

Dado que una red de sensores como puede ser una WASN en un entorno de AAL genera una cantidad de datos muy grande, es común que las infraestructuras clásicas de telecomunicaciones tengan problemas a la hora de procesar y transmitir toda la información asociada a un escenario [37, 38]. Así pues, el paradigma del *fog computing* ha nacido como alternativa para almacenar, procesar y analizar datos en entornos heterogéneos con conectividad limitada o con limitaciones en la capacidad de almacenamiento y capacidad de cómputo del *hardware* asociado como pueden ser entornos de Internet de las cosas *Internet of Things* (IoT), AAL, etc.[39, 40, 41]. Así pues, el paradigma del *fog computing* tiene como objetivo principal desacoplar el proceso de tratamiento de datos en dos (o más) fases. Esto permite (1) llevar a cabo procesos de procesado sencillo y concentración de datos cerca de donde se generan los datos (proceso que normalmente se denomina *edge computing*) [42] y (2) realizar la parte pesada del procesado en la nube (normalmente denominado *cloud computing*)[38, 41, 43]. El *fog computing* ha resultado, pues, en una poderosa herramienta para poder tener tanto las ventajas que ofrece un *cloud* como las ventajas de procesar los datos que requieren ser tratados a tiempo real en local (cerca del usuario final, lo que hace que haya menos latencia). En [44] explican cómo el tratamiento de datos de un sistema de *e-Health* se ha realizado a tiempo real en el ordenador de casa del paciente, mientras que los meta datos se han llevado a un *cloud* para ser posteriormente tratados.

A continuación se proceden a explicar arquitecturas de *fog computing* de proyectos relacionados con el ámbito del AAL. En primera instancia, en [40] se presenta una arquitectura de *fog data* para una aplicación de recolección de datos relacionados con la salud y adquiridos mediante sensores *wearables* (es decir, relojes inteligentes, brazaletes, gafas, etc.). En esta situación, la arquitectura permite reducir en un 99 % la cantidad de datos que se suben al *cloud*[39]. Por otra parte, en [45] se presenta una infraestructura a gran escala basada en *fog computing*. En este caso, son los *smart phones* de las personas que se conectan al *cloud* de *Amazon Web Services* y permiten detectar caídas en los pacientes supervivientes de apoplejía. En [39], de forma similar, los dispositivos *edge* permiten acelerar el procesado a tiempo real y el *cloud* se utiliza únicamente para guardar los meta datos de los pacientes así como su historial.

También es importante tener en cuenta que utilizar el paradigma de *fog computing* permite reducir los problemas de ciber-seguridad que típicamente se pueden encontrar en aplicaciones de *eHealth*, ya que se puede (1) evitar transferir la información sensible por redes de terceros (como Internet) [46] o (2) anonimizar los datos antes de enviarlos al *cloud*[21, 43].

2.4 EL PROYECTO HOMESOUND

El Trabajo de Fin de Máster presentado parte, principalmente, de una primera aproximación al problema de la monitorización de eventos acústicos para entornos domésticos. Esta primera aproximación es el proyecto *homeSound* [47, 21], cuyo objetivo era el de detectar hasta 14 tipos de eventos domésticos en los que se incluyen ladridos de perro, agua hirviendo, caídas de persona, etc. que se pueden agrupar en tres grandes grupos: (1) sonidos producidos por animales, (2) sonidos producidos por objetos y (3) sonidos producidos por personas. En este proyecto los eventos detectados debían ser reportados a un centro médico para ser analizados. Este proyecto se debía emplear mediante sensores acústicos no invasivos y haciendo uso de plataformas de bajo coste, y la parte experimental del mismo demostró conseguir un porcentaje de acierto cercano al 82 %.

Sin embargo, el sistema presentado presentaba ciertas limitaciones, de las cuales destacan:

1. **Cooperación de múltiples fuentes acústicas**[48].A pesar de que el sistema del *homeSound* esta diseñado para procesar múltiples *streams* de audio provenientes de diferentes micrófonos en paralelo, cada *stream* de audio se procesa de forma independiente. Así pues, si en un *stream* de audio se detecta un evento anómalo con un nivel de confianza más alto que un umbral, se envía al centro médico. En cambio, si en el audio no se detecta con suficiente confianza ningún evento anómalo, el audio se descarta. En el proyecto no se tuvo en cuenta que tal vez el nivel de confianza es bajo porque hay múltiples eventos sucediendo en un mismo momento, o tal vez no logra detectar porque el evento ocurre muy lejos de todos los micrófonos. Por lo tanto, en este proyecto se descarta un *stream* de audio sin comparar el resultado obtenido con el resultado que presentan los otros sensores de la red. Esto es un problema ya que si varios micrófonos han detectado un mismo evento con un bajo nivel de confianza en un mismo momento, probablemente ese evento sí que esté ocurriendo en realidad.

2. **Sistema de toma de decisiones no automatizado**. Las etiquetas que la plataforma reporta al centro médico deben ser manualmente analizadas por un equipo de expertos, lo que limita enormemente la escalabilidad del sistema (si hay muchos pacientes a monitorizar también habrá muchas personas que deban estar supervisando las etiquetas del sistema). Además, debido a esta limitación el sistema es también susceptible al error humano (por ejemplo, si solamente hay un experto supervisando y ocurren varios eventos de diferentes pacientes a la vez, el expero podría perderse algún evento). Sin embargo, es importante remarcar que la supervisión humana del sistema es normalmente necesaria en los proyectos de AAL, no es únicamente una limitación de este proyecto, ya que los sistemas automáticos son propensos a generar falsas alarmas [49].

3. **Diversidad en los eventos del *data set***. El *homeSound* se diseñó como un sistema de soporte de AAL de propósito genera, sin considerar ningún escenario de uso específico. Así, constaba de un sistema de aprendizaje automático (*Learning Classifier System Learning Classifier System* (LCS)) muy versátil capaz de detectar un espectro de eventos muy amplio. Estos eventos, eran tan diferentes que era fácil para el sistema diferenciarlos (por ejemplo, diferenciar un cristal rompiéndose de una impresora es más fácil que diferenciar entre dos personas hablando y una persona gritando). Esto hizo que el sistema obtuviera un porcentaje de acierto bastante optimista pero con algunas limitaciones (por ejemplo, los eventos de caídas de persona tenían un porcentaje de acierto cercano al 62 %, siendo un evento muy importante y representativo para la monitorización de pacientes).

4. ***Data set* de entrenamiento de propósito general**. El *data set* utilizado para entrenar el LCS se componía de ficheros de audio obtenidos de diversas fuentes y con características muy distintas: eventos con mucho ruido de fondo, ficheros de audio con múltiplos eventos solapados que se etiquetan como un solo evento, ficheros de audio grabados a distintas frecuencias de muestreo, etc. Este *data set* funcionó como primer prototipo, pero es de esperar que utilizando el mismo LCS pero mejorando el *data set* se pueda conseguir un mayor porcentaje de acierto.

3

CASO DE USO DE LA ARQUITECTURA: LA FUNDACIÓ AVE MARIA

Resumen. En este capítulo se presenta el escenario en el que se ha inspirado el proyecto: la *Fundació Ave Maria*. En él se explica dónde está situada geográficamente la fundación y cómo están distribuidos los edificios en los que se instalaría el prototipo propuesto.

En este capítulo se presenta el escenario y los requerimientos donde se pretende instalar la plataforma de AAL. Es importante hacer un estudio previo del escenario donde se pretende instalar la red para que así, a la hora de diseñar el modelo de AAL, se optimicen los diseños de manera que satisfagan a los usuarios de la manera más eficiente posible.

El escenario en el que se pretenden instalar los sensores es la llamada *Fundació Ave Maria*. Esta fundación es una institución privada sin ánimo de lucro cuya misión es la asistencia de personas con discapacidades intelectuales en su edad adulta, así como a sus familiares. Esta institución se formó en 1987 y está reconocida por la Generalitat de Cataluña.

En la actualidad la institución ayuda a más de 800 personas mediante distintas actividades, y cuenta con un equipo de personal de más de 65 personas y 110 000 horas de dedicación.

Además, la fundación también colabora con distintas universidades y centros de investigación para poder obtener productos (muchos de ellos tecnológicos) que ayuden a llevar a cabo su misión. Entre ellos, el centro destaca los siguientes productos: sillas que no vuelcan, sofás y butacas resistentes y de bajo mantenimiento, cucharas de silicona adaptadas,sistemas de identificación de personas, sistemas de identificación y clasificación de la ropa, control de accesos, detectores de presencia y detectores de caídas.

Para obtener más información de la fundación, se puede visitar su página web: `http://www.avemariafundacio.org`.

Como se puede ver en el mapa de la Figura 2, la fundación está situada en Sitges, comarca del Garraf de la provincia de Barcelona en Cataluña. Concretamente, la sede de la fundación está marcada en el mapa con un globo rojo.

Centrándonos en la topología de la fundación, se debe tener en cuenta que la misma cuenta con (1) una sede principal compuesta de tres edificios y (2) diversos pisos independientes en la ciudad de Sitges donde adultos parcialmente dependientes viven de la forma más autónoma que pueden. Así pues, para llevar a cabo el proyecto se debe tener en cuenta que hay una distinción entre los dos tipos de escenarios posibles. En la siguiente sección se explica en detalle cada uno de estos tipos de escenarios.

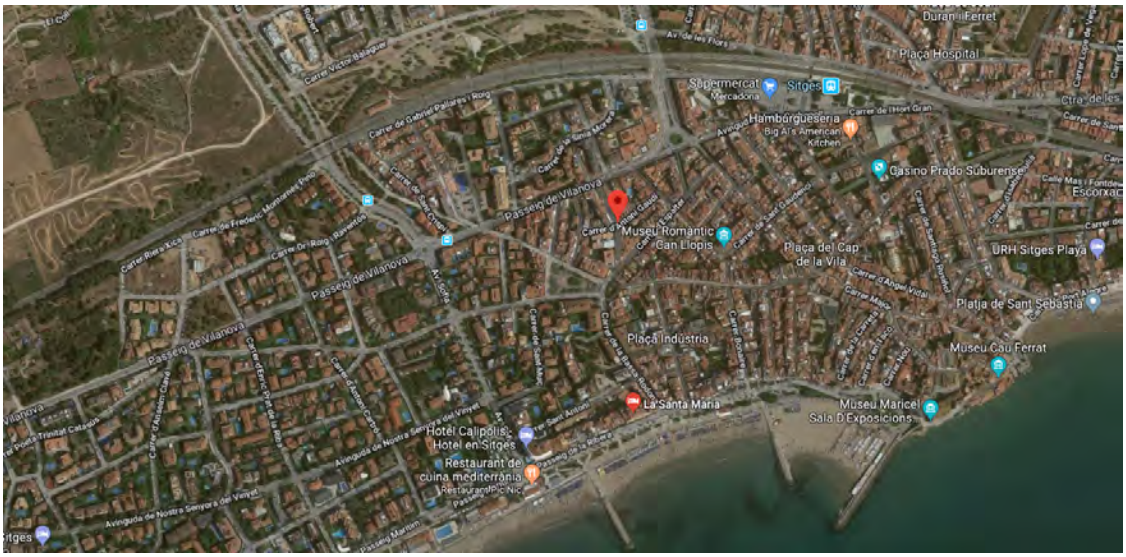

Figura 2: Mapa de Sitges.

3.1 TOPOLOGÍA DE LOS EDIFICIOS

Los servicios que se ofrecen en la Fundación Ave María se ven distribuidos en diferentes edificios según las necesidades de cada paciente. Concretamente, se pueden encontrar los siguientes tipos:

- **Campus residencial (RC)**. La fundación Ave María tiene un campus residencial para aquellos pacientes que necesitan atención constante. En estos edificios, los pacientes conviven 24 horas al día, 7 días a la semana, 365 días al año. Sin embargo, los pacientes están distribuidos en pequeñas casas donde son atendidos por profesionales constantemente. Así pues, el campus residencial es un espacio de más de 3 000 m^2 distribuidos en tres edificios principales y pequeñas casas donde viven unas 60 personas que no tienen la capacidad de vivir de forma autónoma y necesitan atención constante. En la Figura 3 se puede ver ampliado un mapa de cómo se encuentran distribuidos los edificios. Por otra parte, en la Figura 4 se puede ver cómo sería la estructura interna de una casa en las que viven las personas dependientes. Se puede observar que hay zonas comunes (como una sala de estar o una cocina), así como habitaciones individuales. Los conos verdes representan los sensores acústicos WAS (con sus diagramas de directividad) que se deberían poner para que hubiera cobertura total en el piso. Se puede observar que se intenta que todas las zonas queden redundadas por más de un sensor acústico, ya que así se intenta dar redundancia a la misma vez que se da más seguridad sobre los resultados de clasificación automática. Este punto quedará posteriormente explicado en los siguientes capítulos de la memoria.

- **Red de domicilios (NH)**. La fundación Ave María también da asistencia a una red de ocho casas distribuidas por el pueblo de Sitges. Estas casas son

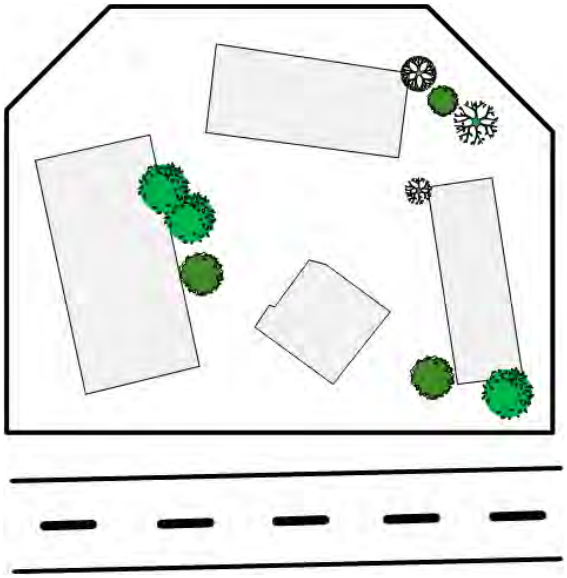

Figura 3: Mapa del campus residencial.

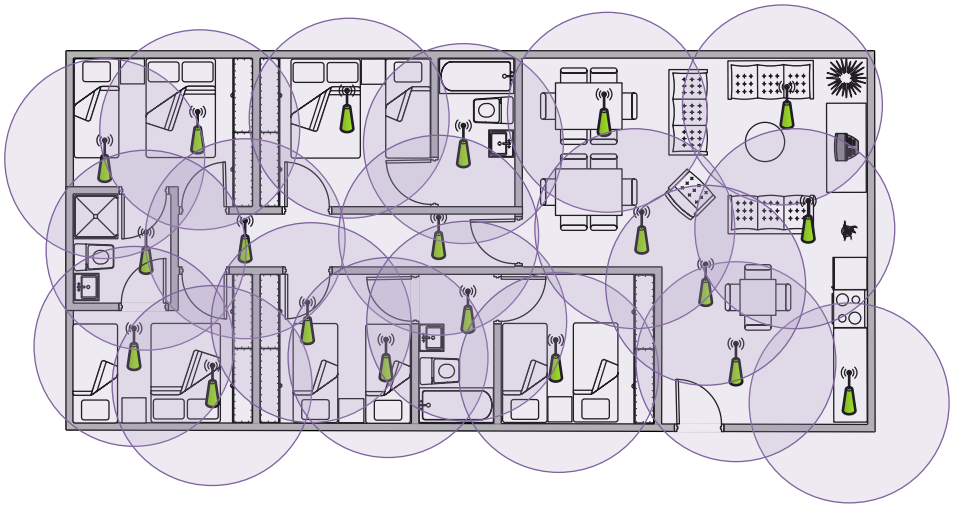

Figura 4: Interior de una casa en el campus residencial.

para adultos que tienen algún tipo de discapacidad pero que quieren (y tienen la capacidad) de vivir su vida de forma casi autónoma y necesitan menos servicios de la fundación. En este caso, los habitantes de los pisos únicamente necesitan los servicios de la fundación de forma intermitente y con un grado bajo de intensidad (es decir, no necesitan tanta ayuda como aquellos que viven en el campus residencial de la fundación). En la actualidad, estos pisos están dotados de domótica no invasiva como son los sensores de detección de presencia, por lo que la comunicación entre el centro y los pacientes es limitada e insuficiente. De ahí surge la necesidad de desplegar la WASN. Así pues, en la Figura 5 se puede ver cómo sería un edificio de la red de edificios, con el domicilio perteneciente a la fundación marcado en color gris. Por otra parte, en la Figura 6 se puede ver un mapa del interior de un piso de la red de domicilios. En este caso, se puede

apreciar que el mapa es el típico de un apartamento habitado por pocas personas. Se puede apreciar que el número de sensores es notablemente menor, ya que hay menos habitaciones; pero, al igual que en el caso de los edificios del campus residencial, se intenta que todas las áreas se vean cubieras por un mínimo de dos sensores WAS.

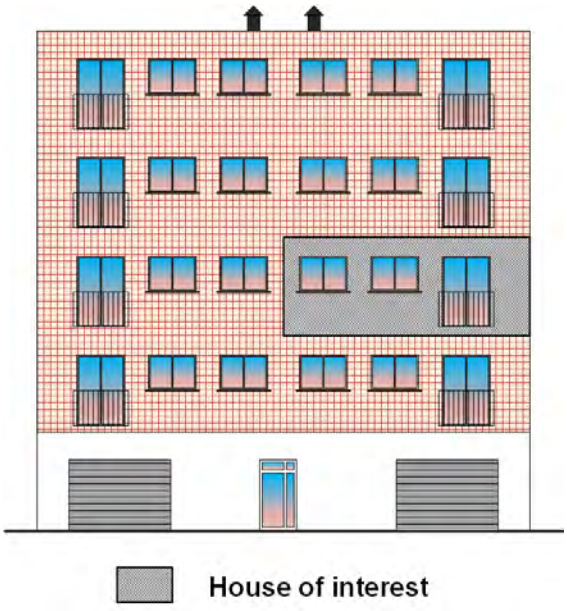

Figura 5: Estructura de uno de los edificios de la red de domicilios.

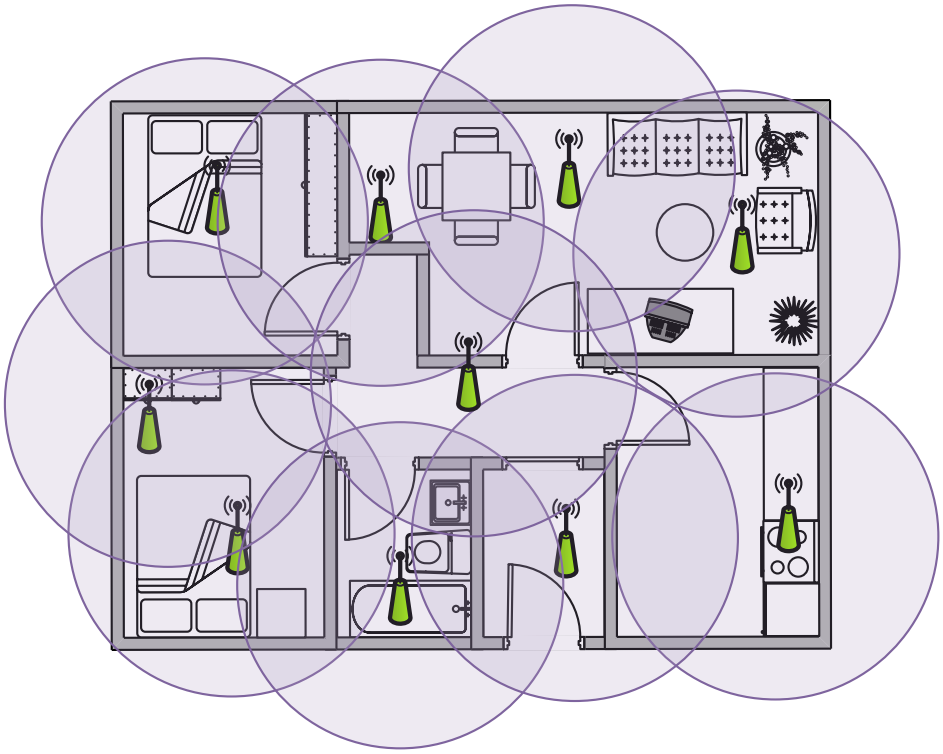

Figura 6: Interior de un piso de la red de domicilios.

En el caso de los edificios del campus residencial, el objetivo principal es el de monitorizar a los pacientes especialmente en las áreas privadas (a pesar de que en las áreas compartidas también hay sensores). Con ello se consigue que el personal de la fundación pueda minimizar la supervisión manual de los pacientes. Con la red funcionando, los trabajadores podrían dedicar la mayor parte de su tiempo en atender y cuidar a las personas que lo necesiten, reduciendo las tareas de supervisión especialmente en las zonas privadas donde los pacientes pueden querer tener más intimidad.

Por otra parte, en las casas o pisos privados, los servicios de AAL se instalan en la casa privada del usuario para poder monitorizar de forma remota a los habitantes y producir una alarma a los trabajadores de la fundación cuando los pacientes necesiten atención. En este caso, los sensores deben estar constantemente trabajando en las zonas "públicas"(zonas que el paciente comparte con sus familiares o el resto de habitantes del piso) o zonas "privadas"(la habitación personal del paciente).

Como se puede observar, las medidas de los escenarios son diversas; lo que constituye un reto a la hora de diseñar e implementar la herramienta de monitorización. Concretamente y resumiendo lo explicado, hay dos tipos de escenario posibles. El primero de ellos es un edificio grande de dos plantas con, al menos, 12 habitaciones diferentes (baño, cocina, comedor y dormitorios). El segundo de ellos, los pisos, tienen tamaño variable y pueden tener entre 3 y 6 habitaciones.

Por esta razón, en este trabajo —que es una prueba de concepto—, se va a asumir que el sistema se instalará únicamente en los comedores de las casas y la residencia para hacer los test. Esto permitirá que las pruebas sean más homogéneas y los resultados sean significativos para los distintos escenarios. Quedará como una línea de futuro estudiar el número de sensores óptimos que se deberían instalar en cada casa dependiendo del tamaño de la misma y su distribución.

Así pues, los requerimientos para hacer la arquitectura del sistema de la plataforma propuesta se listan a continuación:

1. **Monitorización a gran escala**. Las localizaciones de los domicilios de los pacientes pueden estar distribuidos a lo largo de una ciudad de tamaño medio, y la única forma de comunicarse con los edificios de la *Fundació Ave María* es mediante Internet, pues no hay redes de comunicación dedicadas entre ellas.

2. **Escalabilidad**. Un número arbitrario de casas (y pacientes) bajo supervisión y monitorización se pueden añadir o quitar de la red según lo vaya necesitando la fundación.

3. **Fiabilidad y tolerancia a fallos**. Todos los domicilios y edificios de la fundación deben ser constantemente monitorizadas para poder generar una señal de alarma a la fundación cada vez que se detecte que un paciente se encuentra en una situación de emergencia.

4. **Monitorización en escenarios heterogéneos**. Los sensores se implementarán tanto en los entornos domésticos como en el campus residencial. Así pues, el sistema debe ser lo suficientemente flexible como para poder dar servicio a casas de diferentes medidas, con número de sensores diferentes (típicamente proporcionales a las dimensiones de cada casa) y evitar que haya áreas del escenario que se encuentren sin cobertura.

Así pues, debidas las especificaciones comentadas, el reconocimiento de eventos acústicos [21] parece ser una técnica ideal para abordar el problema propuesto. Desplegando una red de sensores acústicos se pueden obtener resultados fiables minimizando los costes a la hora de detectar eventos y con un ancho de banda no muy elevado (se debe tener en cuenta que deberá ir por Internet, no habrá redes dedicadas).

4

EVENTOS ACÚSTICOS A CONSIDERAR

Resumen. En este capítulo se presenta la naturaleza de los eventos acústicos que se han escogido para poder hacer el seguimiento acústico de los pacientes de la *Fundació Ave Maria*.También se hace un estudio de la caracterización espectral de los sonidos escogidos y, finalmente, se detalla cuál ha sido el *data set* utilizado.

4.1 INTRODUCCIÓN

En primer lugar, se debe decidir qué eventos acústicos serán los significativos a la hora de generar las alarmas. En la siguiente sección se explica qué eventos acústicos se han tenido en cuenta en este trabajo.

4.2 NATURALEZA DE LOS EVENTOS ACÚSTICOS

En la actualidad, existen múltiplos eventos acústicos que podrían ser un indicativo que una persona se encuentra en una situación anómala o de emergencia. Sin embargo, hay que saber diferenciar cuándo se debe generar una alarma porque la persona necesita ayuda y cuándo no debe generarse una alarma porque la situación es normal para el paciente. Para la realización de este trabajo, las personas trabajadoras en la *Fundació Ave Maria* han propuesto cuatro líneas de acción que contemplan un total de hasta nueve diferentes tipos de sonido a analizar. Se cree que mediante el análisis temporal de estos eventos anómalos se podrá detectar cuándo estos eventos deben producir una alarma al centro.

Estos cuatro tipos de eventos (que engloban los nueve tipos de eventos a detectar) se presentan a continuación:

1. **Timbre de la puerta o teléfono sonando.** Si en casa de un paciente suena repetidamente el timbre y no se abre la puerta en un largo periodo de tiempo se podría considerar que el evento es suficientemente importante como para generar una señal de alarma al centro. Lo mismo pasaría con una llamada de teléfono. Podría darse el caso en que, por ejemplo, primero se llamara al timbre y posteriormente al teléfono y la persona no contestara a ninguno de los dos eventos. Esto podría ser un indicativo de que la persona no está en condición de contestar y necesita una ayuda personal del centro.

2. **Presencia de personas en casa a demás del paciente.** La presencia de más de una persona en casa del paciente podría ser un riesgo potencial. Si las personas que hay en el piso no están autorizadas para estar ahí, el

paciente podría encontrarse en una situación de intimidación que podría llegar a ser peligrosa. A pesar de que no sea una situación exclusivamente de riesgo, debería tenerse en cuenta y poderse detectar para poder generar una alarma preventiva a la fundación.

3. **Gritos.** Los gritos del paciente siempre se consideran una señal de alarma, ya que pueden ser debidos a muchos factores. Entre ellos, se podría destacar que el paciente está gritando porque no se encuentra bien, porque sufre de un ataque de ansiedad, porque se ha generado fuego en casa, por que han entrado a robar, etc. Así pues, de forma preventiva, los gritos siempre deberían generar una señal de alarma para que la fundación pueda cersiorarse de que todo esté bajo control o poder proporcionar la ayuda que el paciente necesite.

4. **Actividad en casa durante la noche.** Voces, la televisión encendida, música o cualquier otro signo de actividad anómala a determinadas horas del día también pueden ser un signo de alarma. Esto podría indicar que el paciente está despierto y activo durante la noche, lo que podría ser un signo de que el paciente está desorientado o tiene una emergencia en casa. Así pues, esto también debería generar una señal de alarma para que la fundación se pudiera asegurar de que el paciente está bien.

Teniendo en cuenta estos cuatro grupos, los eventos que se han escogido para clasificar de forma automática y así poder generar las alarmas son: picar a la puerta o *Door knocking*(este sonido es el que se realiza con el puño cuando se pica sobre una puerta de madera), timbre de puerta o *Door bell*, gritos o *Screaming*, personas hablando o *People talking*, silencio o *Silence*, golpe de puerta o *Door closing*, teléfono o *Telephone*, televisión o *Television* y cristal roto o *Glass breaking*.

Entonces, a la hora de poder diseñar un clasificador hará falta ver cómo de iguales (o diferentes) son estos sonidos en espectro a medida que varía el tiempo, pues esto será un factor clave a la hora de saber cómo se deben parametrizar.

4.3 CARACTERIZACIÓN ESPECTRAL DE LOS SONIDOS A IDENTIFICAR

Se procede a hacer el espectrograma de un sonido de cada clase para poder compararlos entre ellos. Un espectrograma no es más que la representación espectral de un señal ventaneado en el tiempo. Así pues, un espectrograma tiene tres dimensiones: el tiempo, la frecuencia y la amplitud de la distribución de energía. En este caso, se pintan en un gráfico de dos dimensiones donde el eje de las X corresponde a la variación temporal y el eje de las Y corresponde a la variación frecuencial. Para representar la variación en amplitud para ver la distribución de energía de una ventana se ha utilizado un eje de colores donde el naranja indica un nivel de energía máxima mientras que el color azul

representa un nivel de energía mínima. Así pues, para calcular un espectrograma lo único que hay que hacer es la transformada de *Fourier* de una ventana temporal de un número determinado de muestras que va corriendo por todo el audio e ir poniendo los resultados de forma consecutiva hasta conseguir el espectrograma completo. En este proyecto, se ha utilizado el software MATLAB ya que dispone de una función de espectrograma que permite crear la gráfica de forma sencilla. Se ha utilizado una ventana de tipo *Hanning* para hacer el ventaneado de la señal y se ha utilizado un factor de *overlapping* del 50 %. Esto significa que la ventana que va recorriendo la señal debe recorrer únicamente $\texttt{tamaño ventana}/2$ en cada paso, calcular la FFT y ponerla en la matriz de espectrograma resultante. En la Figura 7 se muestra un ejemplo de cómo sería una ventana corrediza o *sliding window*[50].

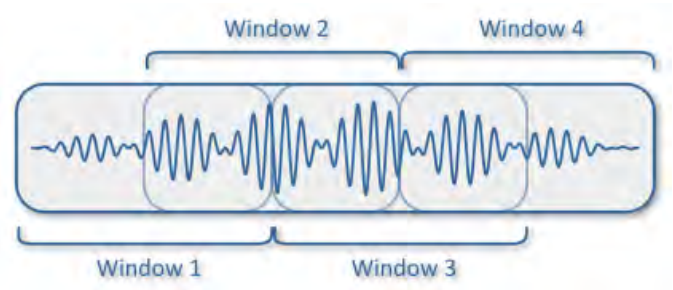

Figura 7: *Sliding window*.

En la figura 8 se muestra el espectrograma de las diferentes clases.

Se puede apreciar que a pesar de que el rango de frecuencia es bastante similar en todos los sonidos, hay sutiles diferencias que permiten distinguirlos. Por ejemplo, se puede observar que los sonidos de *People talking*, *Television* y *Scream* tienen distribuciones similares (como es de esperar, ya que todos provienen de la voz humana); pero la clase *Television* tiene componentes más fuertes en baja frecuencia mientras que la clase *Scream* tiene componentes más fuertes en las altas frecuencias y la clases *People talking* tiene remarcadas las frecuencias medias.

De forma similar, la clase *Door bell* tiene componentes fuertes en media y baja frecuencias, mientras que la clase *Silence* (que representa principalmente el sonido de ambiente que puede haber en cualquier domicilio) no tiene en absoluto componente en alta frecuencia.

Además de las diferencias en frecuencia, también se pueden observar cambios temporales en cada tipo de señal. Por ejemplo, observando las clases de *Door knocking* y *Telephone* se puede observar que mientras que las repeticiones entre golpes de puerta tienen un periodo de unos ~0.3 segundos, la melodía del teléfono tiene un periodo de unos ~0.02 segundos. Además, estas diferencias temporales también pueden ser ventajosas a la hora de distinguir entre las clases de *Door knocking* y *Door closing*, ya que aunque ambas tengan un patrón similar en frecuencia (es decir, transitorios muy bruscos al inicio del sonido y al final un decrecimiento más suave de frecuencias), el sonido de *Door closing* no

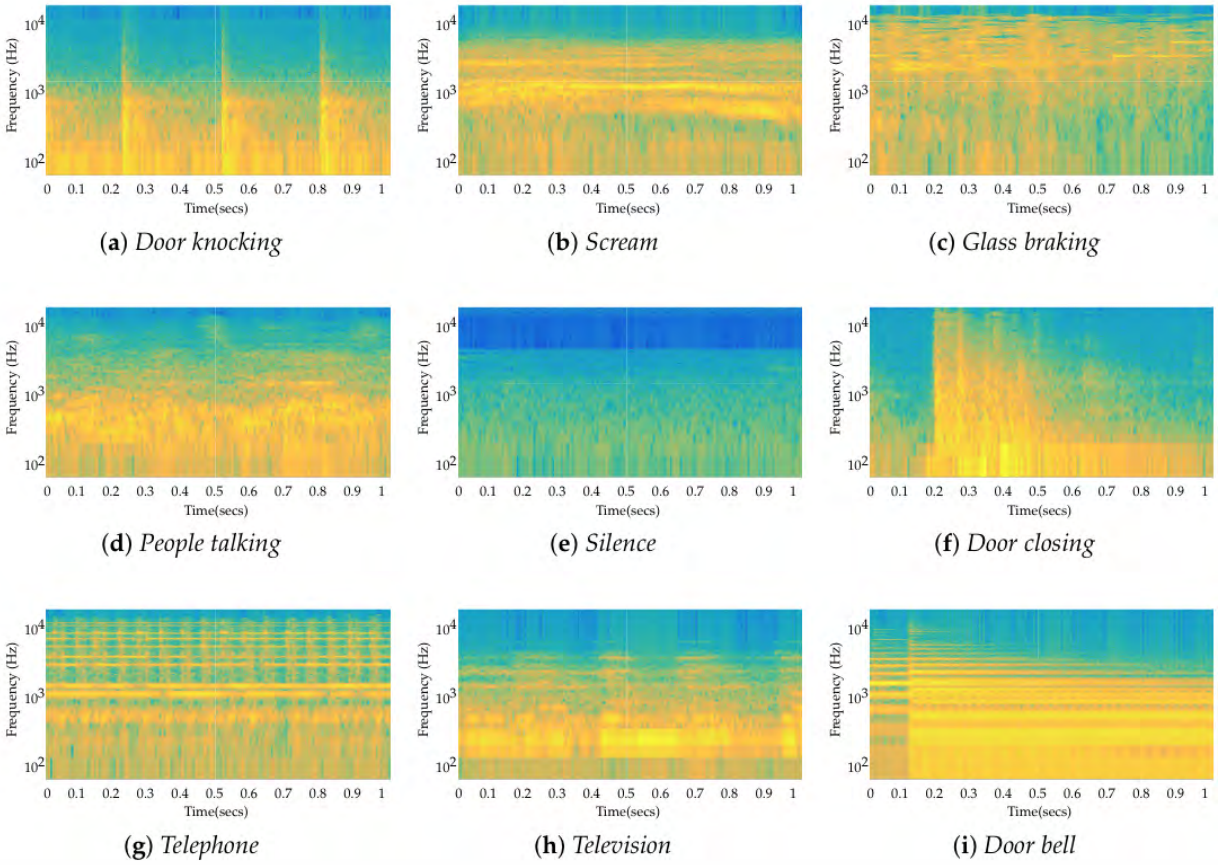

Figura 8: Ejemplos de espectrogramas de los sonidos a identificar.

tiene más que un golpe mientras que el sonido de *Door knocking* tiene múltiples repeticiones.

Finalmente, el sonido de *Glass breaking* tiene una variación espectrotemporal muy distinta a las demás y con una evolución muy peculiar, por lo que parece que se podría distinguir bien a priori.

Así pues, la conclusión a la que se puede llegar observando los espectrogramas, es que un sistema automático de clasificación para estos eventos debería tener en cuenta tanto las características frecuenciales de la señal como la evolución temporal de la misma, ya que ambas nos aportan características del señal que serán útiles para distinguirlas.

4.4 DATA SET UTILIZADO

Para la parte experimental del trabajo se ha utilizado un *data set* con sonidos de las nueve clases de sonidos a identificar. El hecho de que hay sonidos poco comunes (por ejemplo gritos o cristales rotos) ha hecho que no sea posible ir a la *Fundació Ave Maria* a obtener muestras de audio, ya que se tardarían años en poder recopilar un número de datos significativos para poder empezar a hacer experimentos.

Así pues, el *data set* utilizado se ha descargado de Internet. Además, al *data set* utilizado se le han aplicado técnicas de *data augmentation* para obtener más muestras en aquellas clases que no había muchas.

El término *data augmentation* hace referencia al aumento de datos en aquellos problemas en los que se disponen de muestras limitadas para ampliar un *data set*. Es decir, básicamente se copian y modifican los datos originales para obtener otros ficheros de audio (sintéticos) y así tener más datos.

Las técnicas de *data augmentation* no son únicamente válidas para señales acústicas, por lo que para terminar de entender el concepto se va a poner un ejemplo de aumento de datos en imágenes para posteriormente hacer una analogía al aumento de datos de señales acústicas. En la Figura 9 se puede ver un ejemplo de *data set* de imágenes aumentado.

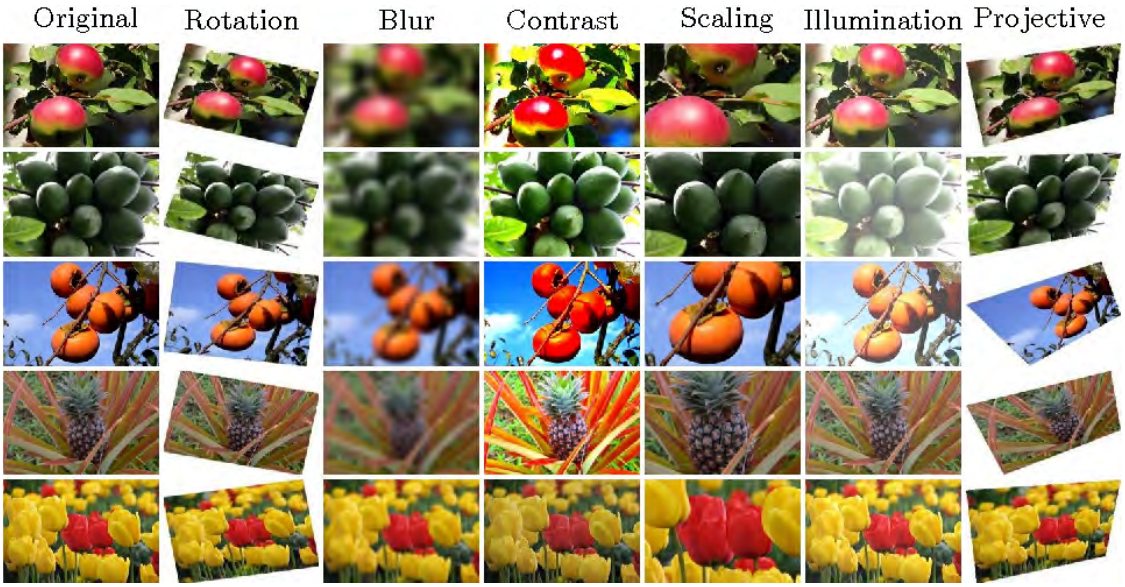

Figura 9: Ejemplo de *data augmentation* en un *data set* de imágenes [51].

Como se puede ver, el *data set* original consistía en las cinco imágenes de la primera columna, pero mediante la modificación de las mismas se ha conseguido un *data set* de treinta y cinco imágenes. En este caso, las transformaciones que se han aplicado han sido rotación, hacer la imagen un poco más borrosa, aumentar el contrasto, escalar la imagen para quedarse con una única parte de la misma, cambio en la iluminación y proyectar la imagen para que quede con una perspectiva diferente.

En el caso de señales de audio se puede hacer lo mismo pero con diferentes técnicas de transformación. Se puede modificar la señal tanto temporal como espectralmente. Estas técnicas pueden ser: añadir ruido Gaussiano a la señal de audio, *shiftar* parte de la señal de audio de interés, filtrar la señal para dar énfasis a altas frecuencias (o bajas frecuencias), hacer que el audio sea más rápido o más lento mediante interpolación o delmación, etc.

A pesar de que hay muchas técnicas para hacer *data augmentation*, se debe tener en cuenta que el número de datos aumentados que se pueden producir

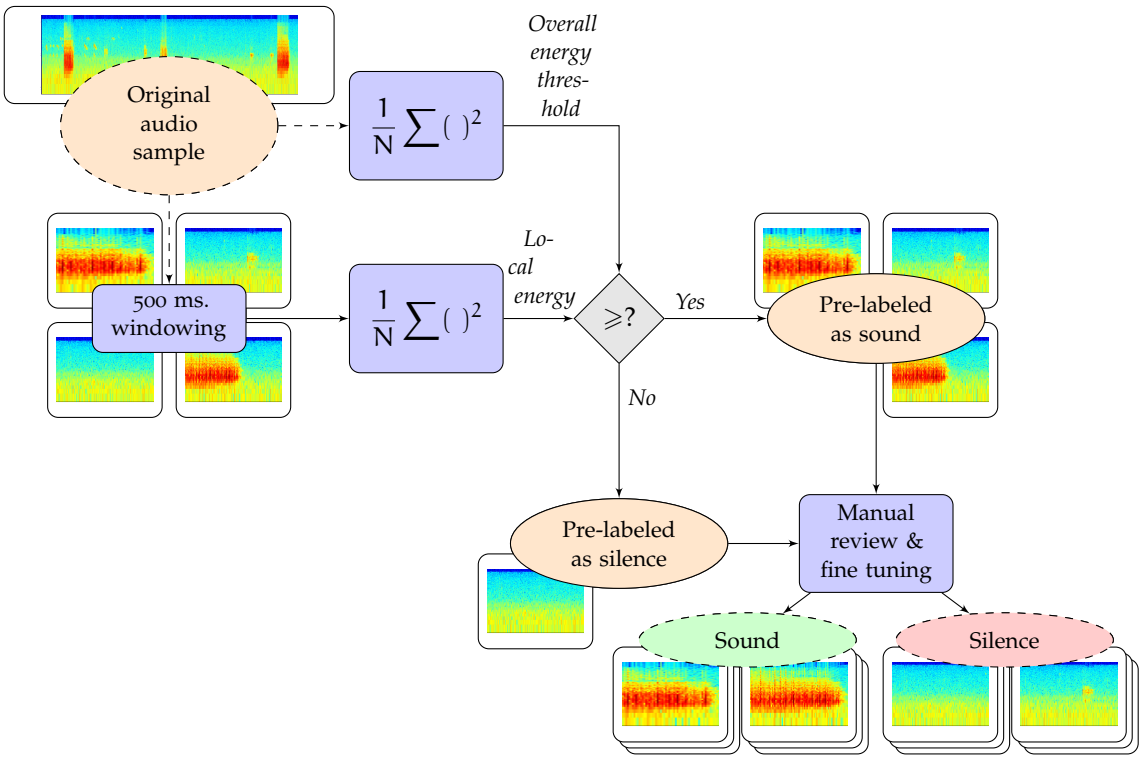

Figura 10: Proceso de segmentación y anotación del *data set*.

son limitados, pues si se abusa del aumento de datos se termina teniendo un *data set* que no representa fielmente a la realidad (se debe tener en cuenta que siempre se parte de los mismos ficheros de audio). Así pues, a pesar de que las técnicas de aumento de datos pueden ir bien para mejorar los porcentajes de acierto de un clasificador; es mejor —en la medida en que se pueda— partir de audios *raw* de nuevas fuentes de audio.

Finalmente, el *data set* obtenido ha sido de 7 116 segundos de los nueve sonidos de interés: *Door knocking, Screaming, People talking, Silence, Door closing, Telephone, Television, Door bell, Glass breaking*. Es importante volver a mencionar que varios de estos eventos són inusuales y difíciles de grabar (pues no se dan en el día a día), así que es de esperar que el *data set* utilizado no esté completamente balanceado. Todos los sonidos se han obtenido de *data sets* de propósito general de repositorios públicos diferentes, así que fue necesario procesarlos y refinarlos para poder terminar obteniendo un *data set* de calidad (que contuviera estrictamente los sonidos de interés). Las técnicas que se aplicaron para refinar y anotar el *data set* fueron las mismas que se utilizaron en [52].

Concretamente, para cada muestra de audio se llevaron a cabo los siguientes pasos, que también se pueden ver visualmente en el diagrama de la Figura 10.

1. Primero se calculó un umbral de energía para distinguir qué es un evento de interés o qué es ruido de fondo (se supone que los eventos de interés tienen una potencia sustencialmente mayor que el ruido de fondo). Este umbral de energía, concretamente, fue el valor medio del señal.

2. Seguidamente, se fragmenta el audio de interés en ventanas de 500 ms y, para cada ventana, se calcula su nivel de energía promedio.

3. A continuación se compara el nivel de energía global con el de la ventana. Si la energía de la ventana es menor que la global se clasifica la ventana como ruido de fondo; por el contrario, si la energía de la ventana es mayor que la global se clasifica la ventana como que hay un posible sonido de interés en ella.

4. Después, con todas las ventanas del audio pre-etiquetadas como posibles eventos o ruido de fondo, se hace una revisión manual ajustando el inicio y el fin de cada ruido y las partes donde no hay sonido se añaden a la clase de *Silence*.

Finalmente, el *data set* obtenido tiene las características mostradas en la Tabla 1. En la tabla se puede ver en detalle cuántos ficheros de audio hay de cada clase, la duración total de los mismos y un diagrama de tipo *boxplot* para que el lector pueda orientarse sobre la duración de los ficheros.

Event	File count	Total length (sec.)	Duration distribution (sec.)
Door knocking	1 821	499	
Screaming	134	177	
People talking	26	662	
Silence	86	2 647	
Door closing	104	82	
Telephone	397	1 270	
Television	28	357	
Door bell	375	949	
Glass breaking	443	474	
Total	*3 414*	*7 116*	1 31 62

Cuadro 1: Características del *data set* utilizado.

Como era de esperar, el *data set* final está desbalanceado (lo que significa que hay muchas más muestras de una clase que de otra. En este momento se podrían llevar a cabo tres alternativas:

1. Balancear el *data set* eliminando las muestras de las clases que tienen más muestras (lo que comúnmente se denomina *random subsampling*). Esto reduciría la entropía del dataset (se pasaría, por ejemplo, de una clase de 2 647 segundos a una clase de 82 segundos), lo que significa que el clasificador tendría menor capacidad de generalización.

2. Balancear el *data set* añadiendo más muestras sintéticas de audio (es decir, hacer más *data augmentation*). Eso no es deseable, ya que aunque se podría incrementar la *accuracy* del clasificador se correría el riesgo de hacer las clases menos distinguibles entre ellas. Además, las muestras de las clases donde hay pocos segundos de audio ya se han obtenido mediante el uso de estas técnicas.

3. Dejar el *data set* sin balancear y terminar de refinarlo cuando el sistema ya esté implementado en la *Fundació Ave María* y se puedan coger muestras reales del entorno si se ve que hay clases en las que el clasificador está obteniendo *accuracy* baja.

Como el propósito de este trabajo es hacer una prueba de concepto y presentar las posibilidades del sistema obteniendo un *worst case scenario*, se ha decidido seleccionar la tercera opción. Así pues, el *data set* se ha dejado desbalanceado sabiendo que puede afectar a la capacidad de generalización del clasificador (algunas clases seguramente van a clasificarse mejor que otras).

4.5 ANÁLISIS DE LA DISTRIBUCIÓN ESTADÍSTICA DE LOS DATOS

Para obtener una primera visión cualitativa sobre la separabilidad entre clases del *data set*, se ha hecho un *t-Distributed Stochastic Neighbor Embedding* (t-SNE) [53]. El t-SNE es una herramienta que permite visualizar datos con un número elevado de dimensiones a partir de posicionar cada punto de datos en un espacio de dos (o a veces tres) dimensiones. Así pues, si el t-SNE muestra los datos separados se puede predecir que será un problema de clasificación fácil. Por otro lado, si en el t-SNE se muestran los datos sobrepuestos unos con otros se puede predecir que será un problema de clasificación difícil.

Concretamente, el t-SNE nace de otro método menos eficiente en coste llamado *Stochastic Neighbor Embedding* (SNE), que intenta reducir la dimensionalidad del objeto a la vez que preserva la estructura del objeto respecto a sus vecinos. Para calcular la reducción de dimensiones se utilizan la entropía y las divergencias de la probabilidad. Lo que hace es para cada valor de entrada (de altas dimensiones) calcular una distribución gaussiana centrada en el propio valor con la finalidad de utilizar la densidad de la gaussiana para definir una distribución de probabilidad de todos los vecinos. Resumiendo, el objetivo principal es hacer que la distribución de probabilidad aproximada replique, en la medida de lo posible, el parentesco de los objetos en el espacio de bajas dimensiones [54].

En la Figura 11 se puede ver el t-SNE resultante. Hay varios factores que se pueden apreciar de esta figura.

1. Hay más de un grupo o *cluster* por cada una de las clases.

2. Las clases se sobreponen entre ellas moderadamente, por lo que parece que a priori no va a ser un problema de clasificación fácil.

3. Hay algunas muestras que se encuentran fuera de los *clusters* de su clase.

Así pues, habrá que buscar una herramienta robusta para poder hacer la clasificación.

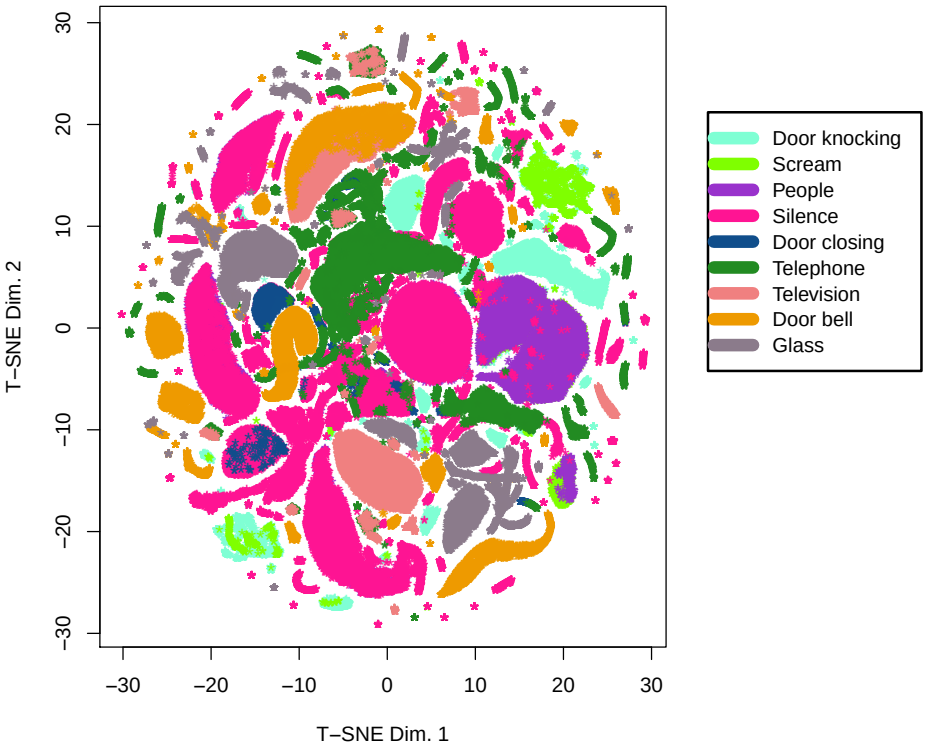

Figura 11: t-SNE [55] del *data set*.

5 CLASIFICACIÓN DE EVENTOS ACÚSTICOS

Resumen. En este capítulo se presenta el sistema de clasificación automática para solventar el problema de la detección y clasificación de eventos acústicos. Para hacerlo, se empieza explicando el proceso de extracción de características de los ficheros de audio, el sistema clasificador empleado y los resultados que se han obtenido tras realizar un estudio exhaustivo.

5.1 INTRODUCCIÓN

Para llevar a cabo la clasificación se han diseñado dos clasificadores. El primero de ellos será a nivel individual sobre los flujos de audio que vayan captando cada sensor (WAS). Este clasificador va a decir a qué clase de las nueve posibles pertenece el ruido que está captando a tiempo real. Así pues, a esta etapa de clasificación se le llamará *Real-time Early Event Detection Layer*. Por otra parte, habrá una segunda etapa de clasificación donde se comparen las etiquetas obtenidas en la *Real-time Early Event Detection Layer* de cada sensor, ya que como se ha comentado anteriormente cada área de la casa se encuentra dentro de un mínimo de dos sensores. Así, si el clasificador de primer nivel comete errores o se ve que hay incongruencias con las etiquetas generadas, el segundo clasificador va a poder corregirlas. A la capa que contiene el segundo clasificador se le llamará *High Level Event Analysis Layer*. La arquitectura (o *Hardware*) en la que se encuentran estos clasificadores se explicará en posteriores capítulos de esta memoria.

5.2 EXTRACCIÓN DE CARACTERÍSTICAS DE LOS FICHEROS DE AUDIO

El primer paso para poder llevar a cabo una clasificación eficiente es la extracción de características de los *streams* de audio, ya que de otro modo la cantidad de datos a procesar sería demasiado grande.

En este trabajo se han escogido los parámetros MFCC para parametrizar las señales de audio. Así pues, se procede a explicar qué es un MFCC y cómo se obtiene.

Lo primero que se debe tener en cuenta es que el comportamiento del oído humano no es lineal en frecuencia, por lo que algunas bandas frecuenciales se perciben con más intensidad que otras aún y recibiendo el mismo nivel de decibelios. Así pues, para medir la tonalidad, Stevens, Volkman y Newmann definieron en 1937 la escala *Mel* para poder cuantificar el nivel que perciben los humanos de forma perceptual. Resumiendo la escala, lo más significativo

es que el oído humano percibe mejor a medias frecuencias que a altas/bajas frecuencias. Aún y ser una escala perceptual (la escala se ha creado a partir de hacer experimentos con voluntarios), se ha definido una aproximación numérica a la escala que corresponde a la siguiente ecuación:

$$mel = 2595 log_{10}(1 + \frac{f}{700})$$

Donde f es la frecuencia de la señal que se está escuchando. A partir de esta escala, es habitual que en problemas de clasificación acústica se caractericen los ficheros de audio de forma perceptual (especialmente en los ámbitos de procesado de habla).

Por otra parte, los coeficientes cepstrales se extraen a partir de la información frecuencial de una señal (su espectro). Concretamente, para extraer los coeficientes cepstrales de una señal se debe hacer el siguiente procedimiento:

$$C(n) = F^{-1}\{\log|F\{x(t)\}|\}$$

Lo que básicamente se traduce en hacer la transformada inversa de Fourier del espectro de la señal. Así pues, los pasos a seguir serían:

1. Ventaneo de la señal (la ventana dependerá del tipo de problema).
2. Transformada de Fourier de la señal.
3. Elevar el resultado de la transformada de Fourier al cuadrado (opcional, si se quiere trabajar con la energía del señal en lugar de su amplitud).
4. Hacer el valor absoluto del resultado del punto anterior.
5. Aplicar un banco de filtros si conviene (como es el caso del cálculo de los MFCC, ya que se ha comentado que se quieren obtener los coeficientes re-escalados según la escala Mel).
6. Hacer el logaritmo del resultado del punto anterior.
7. Calcular la DCT y quedarse únicamente con los M primeros coeficientes, ya que serán los más significativos. El valor de M es típicamente de 13, ya que se considera que con este número de coeficientes el señal ya está suficientemente caracterizado. Para problemas más complejos se pueden usar más coeficientes, pero en el caso del problema de éste trabajo se utilizan únicamente 13 ya que hay restricciones de ancho de banda.

En este trabajo, para poder generar el banco de filtros Mel, se ha partido de la *Auditory toolbox version 2*[56].

Concretamente, el banco de filtros utilizado se muestra en la Figura 12.Además, en la Figura 13 se muestra un *zoom* de los filtros lineales para que se vea que todos tienen la misma amplitud y ocupan el mismo valor de frecuencias.

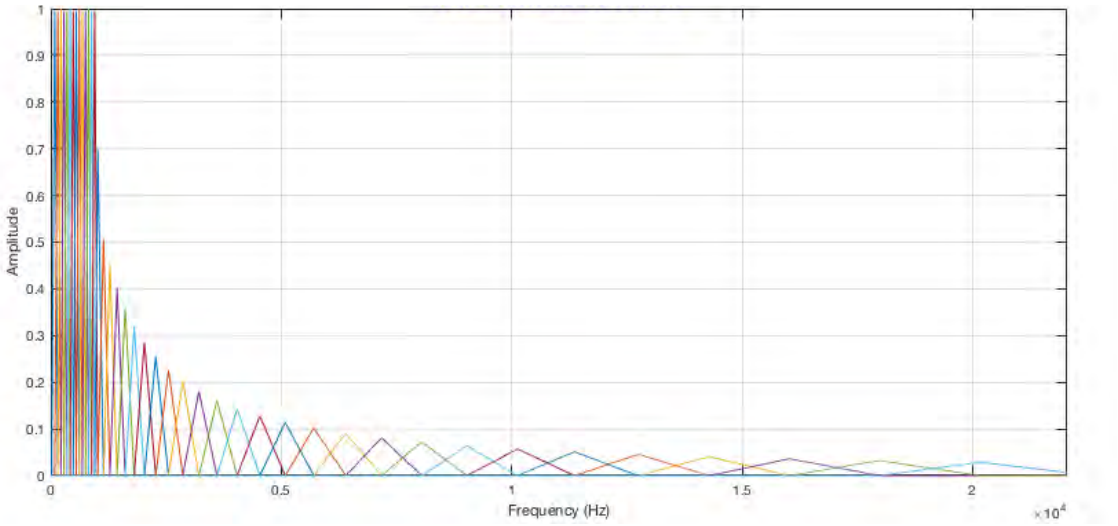

Figura 12: Filtros MFCC aplicados.

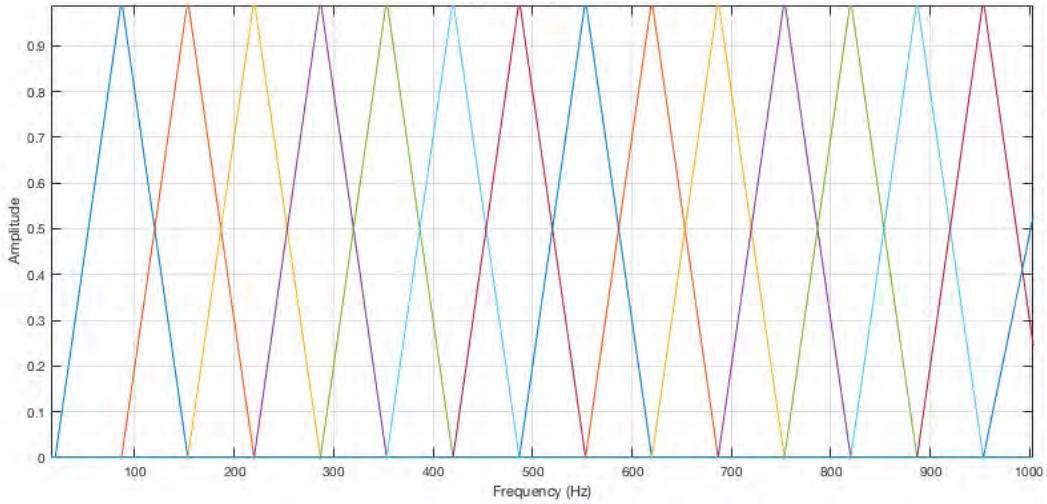

Figura 13: Filtros MFCC aplicados con *zoom* en los 1000 primeros *Hz*.

Como se puede apreciar, los primeros filtros (que son los de frecuencias bajas) son lineales. Esto es debido a que en la escala Mel las bajas frecuencias también tienen un comportamiento lineal. Concretamente, los filtros lineales corresponden a 14 filtros espaciados 66,6 Hz. Además, estos filtros que están equi-espaciados en frecuencia tienen la misma amplitud. A partir de ahí los siguientes filtros empiezan a ser logarítmicos y su amplitud decrece de forma que todos los filtros tengan siempre la misma área (es decir, que como más frecuencias abarque un filtro menor amplitud va a tener, se aplica la técnica de normalización de energía). Concretamente hay 27 filtros logarítmicos. Así pues, haciendo la suma entre los filtros lineales y los logarítmicos se obtiene un total de 41 filtros. Es importante también tener en cuenta que todos los filtros tienen un solapamiento del 50 % para que no haya ninguna frecuencia que no se encuentre cubierta por un filtro como mínimo.

Además de los MFCC, en este trabajo se han hecho pruebas con otros coeficientes cepstrales llamados *GammaTone Cepstral Coefficients* (GTCC), aunque finalmente se optó por utilizar únicamente los MFCC. Los GTCC se calculan del mismo modo que los MFCC pero con la diferencia que en lugar de ser filtros triangulares son filtros Gaussianos. Además, en el cálculo de los GTCC se optó por no utilizar normalización de energía, por lo que todos los filtros tenían el mismo valor de amplitud. Concretamente, se hicieron las siguientes pruebas para obtener la prueba de concepto:

1. Parametrizar con MFCC de amplitud.
2. Parametrizar con MFCC de energía.
3. Parametrizar con GTCC de amplitud.
4. Parametrizar con GTCC de energía.
5. Parametrizar con un vector compuesto de MFCC y GTCC concatenados.

En todos los casos los resultados fueron parecidos. Concretamente la opción que obtenía mejores resultados era la número 5, pero se descartó debido a que utilizar vectores de MFCC y GTCC suponía utilizar más ancho de banda (en lugar de utilizar un vector de 13 componentes para parametrizar una ventana se utilizaba un vector de 26 componentes ya que había 13 de MFCC y 13 de GTCC).

5.3 CLASIFICACIÓN EN LA *real-time early event detection layer*

Una vez obtenidos los parámetros de cada ventana de cada fichero de audio, se procedió a realizar el *machine learning* que permite que el sistema reconozca los sonidos de forma automática.

La primera decisión que se tuvo que tomar fue la de saber qué tipo de clasificador era más adecuado para este tipo de problema. Teniendo en cuenta que se quiere que la clasificación sea a tiempo real, no es adecuado utilizar técnicas que sean muy costosas en la etapa de *test* del clasificador, así que de buenas a primeras se descartó utilizar clasificación jerárquica o clasificadores del tipo k-NN, que son muy sencillos de implementar pero (1) requieren mucha memoria para almacenar el *data set* completo parametrizado para poder hacer comparaciones vector a vector y (2) computacionalmente caros a la hora de realizar la clasificación.

Así pues, los dos algoritmos que quedaron como finalistas para poder llevar a cabo la clasificación fueron o utilizar una SVM o utilizar una ANN. Para tomar la decisión se tuvo en cuenta [57]. A continuación se hace una breve explicación de cada método de clasificación y se procede a explicar cuál se ha escogido para este problema.

5.3.1 *La Support Vector Machine*

Una SVM o *Support Vector Machine* es un método de clasificación que se caracteriza por representar el data set mediante un conjunto de *hiperplanos*.

Un *hiperplano* consiste de un sub-espacio de una dimensión menor al espacio en el que se encuentra. Así pues, en espacios tres dimensiones el *hiperplano* es un plano corriente, en espacios de dos dimensiones un *hiperplano* se corresponde con una línea recta y así sucesivamente si se modifica el número de dimensiones del sub-espacio. Esta caracterización se hace en la etapa de *train* del clasificador, por lo que a pesar de que puede llegar a ser una fase lenta se realiza únicamente una vez y no influye en el tiempo de clasificación. Concretamente, estos *hiperplanos* separarán cada clase de forma que la distancia entre clases sea máxima.

Así pues, una vez caracterizado el espacio mediante *hiperplanos*, para clasificar una nueva instancia simplemente se deberá proyectar el dato a reconocer sobre el espacio de *hiperplanos* creado para poder ver a qué clase pertenece. Dependiendo de en qué lado del *hiperplano* se encuentre se podrá decir que el vector pertenece a una clase u a otra.

El nombre de SVM viene dado debido a que no todas las muestras de entrenamiento se utilizan para crear los *hiperplanos*, sino que éstos vienen caracterizados por muestras del data set que cumplen con ciertas características. Concretamente, la condición para que un vector se pueda considerar vector de soporte es que la distancia entre el plano que contiene los vectores de soporte de una clase y el plano que contiene los vectores de soporte de otra clase sea máxima y esté vacía de muestras (es decir, que el área entre los dos planos sea la mayor posible sin que dentro se encuentren vectores pertenecientes a ninguna clase). Ésta distancia se denomina margen, por lo que es imprescindible que el margen sea máximo para encontrar el *hiperplano* solución.En la Figura 14 se puede ver la representación de una SVM. Una clase serían las estrellas y la otra los círculos. Si se proyectada el triángulo y se viera que está a la derecha del *hiperplano* (como es el caso del ejemplo), se etiquetaría el triángulo como perteneciente a la clase círculo.

Por otra parte, como es fácilmente deducible, existen problemas en los que no es fácil hacer una separación lineal entre clases. Para solucionar estos problemas existe la técnica del *Kernel trick*; técnica en la que se transforma el espacio de características en el que se encuentran las muestras a un espacio de dimensiones superiores en el que sean linealmente separables. El único problema que conlleva éste "truco.es que debido a las matemáticas que hay detrás de las SVM el coste computacional al utilizar un kernel no lineal aumenta significativamente.

En el caso de problemas multiclase (como es el caso de este problema, donde hay concretamente nueve posibles clases), hay dos estrategias para llevar a cabo una clasificación mediante la SVM.

- *One vs. One*: En este método se aplica un clasificador binario para cada combinación posible de dos clases (como si el resto de clases no existieran,

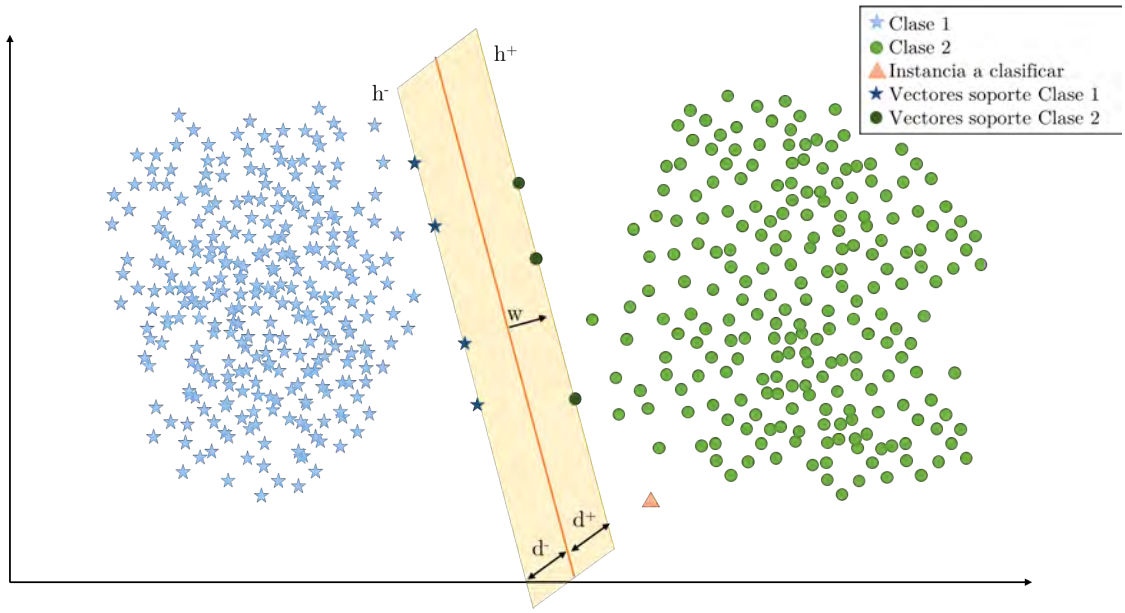

Figura 14: Ejemplo de SVM.

no se tienen en cuenta a la hora de crear el *hiperplano* que separa las clases y se proyecta la muestra a clasificar para ver a qué clase de las dos es más probable que pertenezca. También se almacena el valor de distancia entre la muestra y el *hiperplano*). Se puede deducir entonces que se van a tener que emplear $n(n-1)/2$ clasificadores binarios, siendo n el número de clases (en este caso 9). Tras aplicar los distintos clasificadores, se van a tener múltiples resultados. Para saber qué resultado es el ganador, se va a hacer un recuento de todos los resultados y va a ganar el de mayor peso (recordemos que se guarda el valor de distancia entre la muestra y el *hiperplano*).

- *One vs. All*: En este caso, se entrena el clasificador con n clasificadores distintos (siendo n el número de clases totales, en este caso 9). Para cada clasificador, se separa una clase del resto por un *hiperplano* se realiza el proceso de clasificación. Es decir, que una clase queda aislada y la otras quedan agrupadas como si fuera una clase individual. En este caso, se mira también en qué lado del *hiperplano* queda la muestra y la distancia entre el *hiperplano* y la clase a la que supuestamente pertenece. Al final, la clase ganadora es la que tiene un valor de distancia superior.

5.3.2 *La Red Neuronal Artificial*

Las ANN o *Artifical Neural Network* intentan emular el comportamiento del cerebro humano creando una gran red de "neuronas artificiales"que permitan realizar tareas de aprendizaje automático y, en consecuencia, se pueden utilizar para problemas de clasificación.

Concretamente, estas neuronas artificiales consisten en elementos con la estructura mostrada en la figura 15:

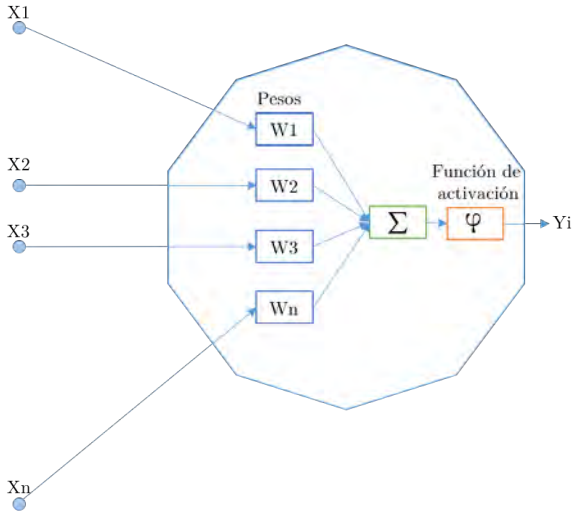

Figura 15: Ejemplo de neurona en una ANN.

Como se puede apreciar en la Figura 15, la neurona tiene múltiples entradas y cada una se multiplica por un peso concreto Wx (más adelante se explica cómo se calcula este peso); posteriormente se hace la suma de todas las entradas multiplicadas por sus pesos y el resultado de la suma se pasa por una *Función de activación*.

La función de activación de la neurona es un parámetro que se selecciona cuando se está creando la red neuronal. Concretamente, las funciones de activación más comunes son las siguientes (aunque se puede utilizar cualquier función que sea derivable como función de activación):

1. **Función logística sigmoide**: Mantiene el valor de salida entre 0 y 1 a partir de cualquier rango de entrada:

$$f(x) = \frac{1}{1 + e^{-x}}$$

2. **Función tangente hiperbólica**: Mantiene el valor de salida entre -1 y 1 a partir de cualquier rango de entrada:

$$f(x) = \tanh(x)$$

3. **Función *Rectified Linear Unit* (ReLU)**: Utilizada especialmente para la visión por computador. Si el valor de entrada es positivo se mantiene, pero si es negativo el valor pasa a valer 0:

$$f(x) = \max(0, x)$$

Así pues, una red neuronal no será más que capas de varias neuronas unidas, ya que la salida de una neurona puede ser la entrada de otra neurona. Cuando hay muchas capas de neuronas, a la red se le suele llamar *perceptrón multicapa*.

En un perceptrón multicapa hay tres tipos de capas:

- **Capa de entrada**: es la capa del inicio de la red de neuronas. En ella se conectan los datos de entrada que se quieren clasificar, por lo que hay tantos puntos como componentes tenga el vector de entrada. En el caso de este problema, como los vectores de entrada son MFCC de 13 componentes, la capa de entrada tendrá que tener obligatoriamente 13 componentes. En esta capa aún no hay ni funciones de activación ni pesos.

- **Capas ocultas**: Este tipo de capas contienen las neuronas que van a aplicar los pesos y las funciones de activación. Son la clave fundamental para que el problema de clasificación funcione. Hay que diseñarla de forma que el valor de *accuracy* obtenido sea máximo teniendo en cuenta que no se quiere hacer una *deep net* en este problema (principalmente debido a que el data set del que se dispone no tiene suficientes datos y como se quiere hacer clasificación a tiempo real también conviene que la red no sea muy grande para que no haya que hacer tantas multiplicaciones).

- **Capa de salida**: Al igual que con la capa de entrada, solo hay una única capa de salida. Esta capa tiene tantas neuronas como posibles clases tenga el problema. En el caso del problema de este trabajo, hay nueve neuronas en la capa de salida, ya que hay nueve clases posibles. En cada una de las neuronas se muestra el porcentaje de probabilidad de que la muestra que ha entrado pertenezca a la clase a la que representa (por ejemplo, si se obtuviera como resultado «1, 0, 0, 0, 0, 0, 0, 0, 0», significaría que con un 100 % de probabilidad la muestra de entrada pertenece a la clase *Door knocking*). Para obtener este valor de probabilidad, es importante que la función de activación de la última capa se a la función *Softmax*. Esta función es parecida a la función sigmoide, con la diferencia de que calcula el valor de probabilidad de que la muestra pertenezca a cada clase, por lo que al final la suma de los valores de las salidas de cada neurona será siempre igual a 1 [58].

Ahora se va a explicar cómo se entrena una red neuronal, que básicamente consiste en definir los pesos de las diferentes neuronas. Esto se hace con un algoritmo denominado *back propagation*[59]. Así pues, para entrenar la red lo que se hace es introducir en la capa de entrada los parámetros de los datos (en el caso de este proyecto serían los 13 MFCC. En este momento, automáticamente se asignan pesos de forma aleatoria en las neuronas, por lo que al final se van a obtener ciertos valores de salida. Estos calores de salida serán (*a*) correctos si el resultado coincide con el valor deseado en cada neurona, o (*b*) incorrectos si el resultado no es el esperado. Entonces, a partir de estos valores, se calcula el error del clasificador (habitualmente se usa el error cuadrático

medio). Para ajustar los pesos de la capa oculta del clasificador se calcula el gradiente del error de las capas superiores y se modifican los pesos en dirección contraria (con el fin de intentar disminuir el error). Hay que tener en cuenta que se realiza el mismo proceso iterativo distintas veces hasta que el error sea lo suficientemente pequeño (es un parámetro que puede ajustar el usuario) o se haya llegado a un número determinado de iteraciones. Entonces, los pesos que se asignarán finalmente a la red neuronal serán aquellos que hayan proporcionado ese error mínimo. Concretamente, en este proyecto se ha utilizado un método estocástico optimizado —llamado *ADAM* — para calcular el gradiente y resolver el problema de la asignación de pesos a cada neurona [60].

Se puede ver que en la fase de entrenamiento el proceso va a ser lento, pues es posible que se deban hacer distintas iteraciones hasta encontrar el valor de pesos óptimo. A pesar de esto, en la etapa de clasificación el proceso será rápido ya que únicamente habrá que hacer multiplicaciones y sumas (como se puede apreciar en la Figura 15.

5.3.3 *Clasificador utilizado*

Finalmente, se ha optado por implementar una ANN a partir de la librería de *scikit-learn* en python. Concretamente, se ha utilizado la librería *MLPClassifier* [61] para crear la red neuronal.

Esta decisión se ha tomado debido a que como se explica en [62], cuando las clases del data set se encuentran desbalanceadas es común que con la SVM haya más error que con la ANN. Además, también se compararon los resultados de *accuracy* obtenidos con la ANN y con la SVM y se vio que con la ANN los resultados eran mejores.

Tras hacer un *grid search* con los distintos parámetros de la ANN, se optó por finalmente utilizar la red que proporcionaba mejores resultados. El diseño de la misma puede verse en la Figura 16.

Concretamente, esta red neuronal tiene una capa de entrada con 13 neuronas por donde entran los 13 coeficientes MFCC de cada ventana, cinco capas ocultas con 100, 70, 50, 30 y 10 neuronas respectivamente y una capa de salida con nueve neuronas (una para cada clase posible). La función de activación de todas las capas exceptuando la última —que se ha implementado con una *Softmax* para poder saber las probabilidades de cada clase— ha sido la ReLU.

Otro problema que puede surgir en problemas de clasificación es el del *overfitting*, que ocurre cuando el clasificador se entrena de forma que está demasiado particularizada para las muestras de entrenamiento; por lo que cuando se hace el *test* del sistema con otras muestras los resultados obtenidos no son muy buenos.

Para intentar datos estadísticamente significativos y evitar el overfitting, se ha utilizado la técnica del *10-fold cross validation*[63, 21], que es un modo particular de hacer una *cross fold validation*. Esto significa que se hacen 10 pruebas distintas de datos en los que un porcentaje de los mismos se utilizan para en-

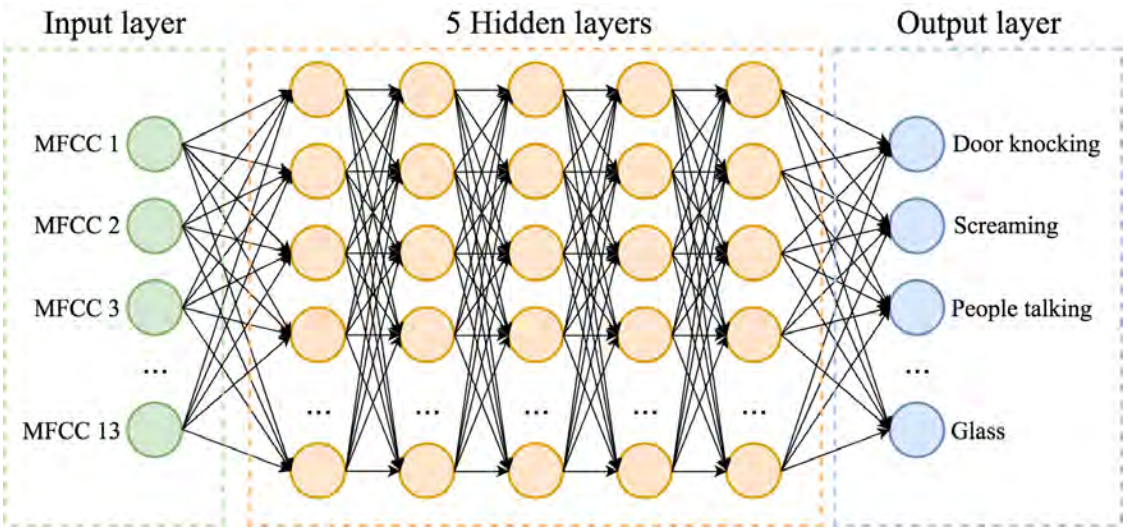

Figura 16: Esquema de la red neuronal utilizada.

trenar el sistema y el resto se utiliza para testear el sistema. Concretamente, en este proyecto se han hecho 10 iteraciones (o *folds*) de datos en los que un 70 % aleatorio de datos se utilizan para *train* y un 30 % de datos se utilizan para *test*. Una vez se obtiene el porcentaje de acierto de cada iteración, se hace el promedio de los 10 resultados y se obtiene lo que llamamos *accuracy* del sistema. En la Figura 17 se puede ver un ejemplo de como funciona una validación cruzada con 10 elementos. Además, para que hubiera aún más precisión, el *10-fold cross validation* se ha llevado a cabo 10 000 veces.

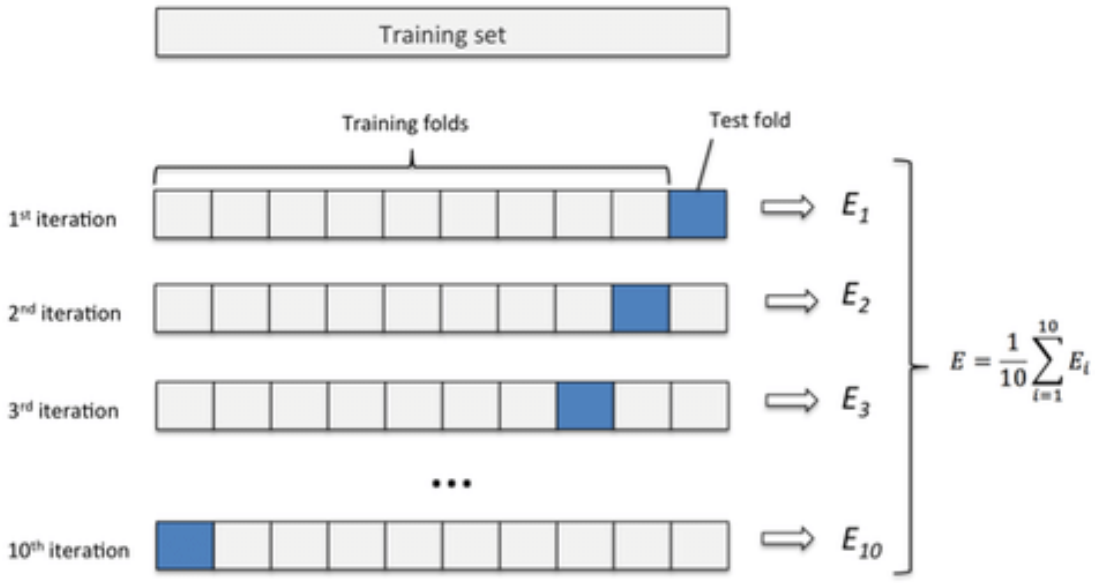

Figura 17: Ejemplo de *10-fold cross validation* [64].

5.3.4 *Resultados*

La *accuracy* general obtenida ha sido del 85.4 %, y la F1 ha sido de 71 %. Cabe recordar que no hay el mismo número de muestras en cada clase, por lo que para hacer los cálculos finales se ha tenido que ponderar (ya que las clases que más peso cuentan son las que tienen más muestras). Para poder analizar mejor los resultados a continuación se presenta la matriz de confusión del sistema. Cabe recordar que una matriz de confusión es un elemento que permite ver cómo está clasificando el sistema los datos. En la primera fila y la primera columna del clasificador se encuentran las posibles clases. Entonces, las filas representan la etiqueta real del audio a clasificar y las columnas representan la etiqueta que el clasificador le ha asignado al dato en concreto. Entonces, la diagonal de la matriz — posiciones (1,1),(2,2),(3,3)...— representan aquellos datos que la matriz ha clasificado correctamente. Por ejemplo, si se posicionara un dato en la posición (1,9); significaría que su clase real sería la de la primera posición de la primera fila pero el clasificador lo ha clasificado con la etiqueta de la primera posición de la novena columna.

En el Cuadro 2 se muestra la matriz de confusión.

Observando los resultados de clasificación por separado, se puede ver un patrón en los errores que comete el clasificador.

Se puede ver que el error de clasificación más común es entre las clases *Door knocking* y *Door closing*. Esta situación no es extraña, ya que ambos eventos suenan prácticamente igual al oído humano (si se escuchan los ficheros de audio del data set). Concretamente, la única diferencia entre ambos eventos es que cuando se cierra una puerta se oye un único golpe, mientras que cuando se llama a una puerta se oyen varios golpes seguidos. Esto puede comprobarse si se revisan los espectrogramas *a* y *f* de la Figura 8. Este error podría solventarse fácilmente si no se considerara únicamente la información espectral de la señal (los MFCC), sino que también se tuviera en cuenta la evolución temporal de la misma. Si más de dos fragmentos seguidos se etiquetan como *Door closing* en menos de un segundo, es muy probable que sea un error de clasificación y que realmente en evento acústico sea un *Door knocking*.

El segundo error más común de clasificación se da entre los eventos *Door bell* y *Telephone*. Esto es debido a que en algunos ficheros de audio en estas clases los sonidos son muy similares, ya que algunos tonos de teléfono pueden parecer timbres de casa y vice versa. En este caso, una visión temporal de la señal también podría ayudar ya que mientras que los tonos de teléfono tienen repeticiones periódicas, cuando una persona llama al timbre lo hace de forma irregular. Además, tanto la clase de *Telephone* como la de *Door bell* se confunden esporádicamente con la clase de *Glass*. Este error tampoco es crítico, pues el sonido de un cristal rompiéndose suele ser del orden de 1 segundo mientras que el sonido de un teléfono sonando o un timbre suelen durar más. Esto puede verse en el *box plot* del Cuadro 1, en el que vemos que tanto la clase de *Telephone* como la de *Door bell* tienen una duración promedio de más o menos 2 y 1.8

Predicted Class \ Actual Class	*Door knocking*	*Screaming*	*People talking*	*Silence*	*Door closing*	*Telephone*	*Television*	*Door bell*	*Glass*
Door knocking	93.21 %	0.01 %	0.58 %	0.10 %	2.03 %	0.01 %	3.39 %	0.19 %	0.48 %
Screaming	4.63 %	79.01 %	2.23 %	0.12 %	1.32 %	6.57 %	2.41 %	2.02 %	1.69 %
People talking	0.72 %	5.43 %	91.67 %	0.23 %	1.04 %	0.23 %	0.03 %	0.28 %	0.37 %
Silence	1.87 %	0.46 %	3.85 %	69.23 %	0.19 %	0.21 %	0.40 %	0.71 %	23.08 %
Door closing	85.71 %	2.98 %	4.23 %	0.49 %	4.17 %	0.08 %	0.68 %	0.19 %	1.47 %
Telephone	4.13 %	2.33 %	0.04 %	0.02 %	1.94 %	80.83 %	0.39 %	4.49 %	5.83 %
Television	0.23 %	0.12 %	4.88 %	0.06 %	0.19 %	0.16 %	94.12 %	0.20 %	0.04 %
Door bell	0.77 %	0.68 %	0.11 %	0.34 %	0.21 %	8.42 %	1.05 %	67.37 %	21.05 %
Glass	1.82 %	0.88 %	0.02 %	0.02 %	0.28 %	0.65 %	0.06 %	0.38 %	95.86 %

Cuadro 2: Matriz de confusión del sistema en la etapa de clasificación de tiempo real.

segundos respectivamente; mientras que la clase de *Glass* tiene una duración promedio de 0.8 segundos.

Finalmente, cabe destacar que la probabilida de perder un ecento acústico (es decir, clasificar un evento acústico real como *Silence*) es considerablemente baja (menor de un 0.5 %) como puede verse en la columna de *Silence* de la matriz de confusión del Cuadro 2. Esto garantiza una gran fiabilidad en esta etapa de detección de acústicos a tiempo real (a pesar de que la *accuracy* a la hora de distinguir qué evento es sea moderado. Así pues, el siguiente experimento que se debe llevar a cabo es el de detectar y corregir los errores mencionados en la *High Level Event Analysis Layer*. Pero antes, se quiere dar un análisis el profundidad de los resultados adquiridos en esta etapa.

5.3.5 *Análisis de resultados*

A continuación se enumeran los distintos parámetros que se han calculado para analizar los resultados de esta etapa de clasificación, que se pueden ver en el Cuadro 3.

1. *Sensitivity* o *Recall*: Es el número de positivos que el clasificador ha clasificado como positivos. Por lo tanto, es el número de positivos que ha acertado el clasificador. Concretamente, se calcula como:

$$\text{Sensitivity} = \frac{\text{Positivos verdaderos}}{\text{Positivos verdaderos} + \text{Falsos negativos}}$$

   Así pues, que la *sensitivity* sea alta significa que la mayoría de eventos clasificados serán clasificados de forma correcta. Observando la matriz de confusión, se puede ver que la mayoría de clases tienen este parámetro muy cercano al 1, únicamente tienen este parámetro por debajo del 0.7 las clases de *Silence* y *Door bell*.El caso de *Door bell* ya se ha comentado, pero en el caso de *Silence* se considera que es mejor que eventos de silencio se clasifiquen como posibles eventos anómalos a que se pasen por alto.

2. *FPR* o *False Positive Rate*: Este parámetro indica el número de falsos positivos (es decir aquellas muestras de eventos negativos que se han clasificado como eventos reales) respecto al número de eventos negativos. Se calcula como:

$$\text{FPR} = \frac{\text{Falsos positivos}}{\text{Falsos positivos} + \text{Negativos verdaderos}}$$

   En este caso, importa mucho que la clase de *Silence* tenga un parámetro *FPR* bajo, ya que no se desea que eventos de otras clases sean clasificadas como silencio (falsos positivos).

	Sensitivity	FPR	Precision	Specificity	F-Measure	MCC	AUC	PRC Area
Door knocking	0.9321	0.0305	0.6973	0.9695	0.7978	0.7899	0.9508	0.6174
Screaming	0.7901	0.0132	0.6051	0.9868	0.6853	0.6826	0.8885	0.5925
People talking	0.9167	0.0203	0.8221	0.9797	0.8668	0.8539	0.9482	0.5473
Silence	0.6923	0.0014	0.9965	0.9986	0.817	0.7628	0.8454	0.3479
Door closing	0.0417	0.0075	0.0606	0.9925	0.0494	0.0411	0.5171	0.4906
Telephone	0.8083	0.0175	0.9093	0.9825	0.8558	0.8288	0.8954	0.4495
Television	0.9412	0.0071	0.8756	0.9929	0.9072	0.9028	0.9671	0.5328
Door bell	0.6737	0.0138	0.8828	0.9862	0.7642	0.7421	0.8300	0.3954
Glass	0.9589	0.1346	0.3370	0.8654	0.4987	0.5244	0.9122	0.8109

Cuadro 3: Detalle de resultados obtenidos en la capa de tiempo real.

3. *Precision*: Este parámetro está estrictamente ligado con la varianza de los resultados. Concretamente, es la inversa de la varianza y se calcula como:

$$Precision = \frac{\text{Positivos verdaderos}}{\text{positivos verdaderos} + \text{falsos positivos}}$$

   Indica el número de eventos catalogados como positivos que realmente son de esa clase.

4. *Specificity* o *True Negative Rate*: Indica la proporción de negativos que realmente se clasifican como tal. Concretamente, se calcula como:

$$Specificity = \frac{\text{Negativos verdaderos}}{\text{Negativos verdaderos} + \text{Falsos positivos}}$$

   Puede verse que los valores son bastante altos exceptuando la clase de *Glass*.

5. *F-Measure* o *F1-Measure*: Es un parámetro para medir la *accuracy* del sistema. Para calcular este parámetro se tiene en cuenta la *Sensitivity* y la *Precision*. Concretamente, se calcula como:

$$F\text{-}Measure = 2 \times \frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}}$$

6. *MCC* o *Matthews Correlation Coefficient*: Devuelve un valor entre -1 y 1, donde +1 significa una predicción perfecta; 0 una predicción que aleatoria y -1 indica una predicción contraria a la que ha hecho el clasificador. Su cálculo es más complicado que los anteriores. Para que la fórmula sea más entendible se escribirá TP como positivos verdaderos, TN para los negativos verdaderos, FP para los falsos positivos y FN para los falsos negativos.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

7. *AUC* o *Area Under the Curve*: es el área de debajo de la curva ROC. Cabe recordar que la curva ROC es la gráfica que representa la *sensitivity* del sistema respecto a la *specificity*.

   Con la *AUC*, hay un estándar de valores que indican si el test ha sido bueno o no.

   - (0.5, 0.6) Test malo.
   - (0.6, 0.75) Test regular.
   - (0.75, 0.9) Test bueno.
   - (0.9, 0.97) Test muy bueno.

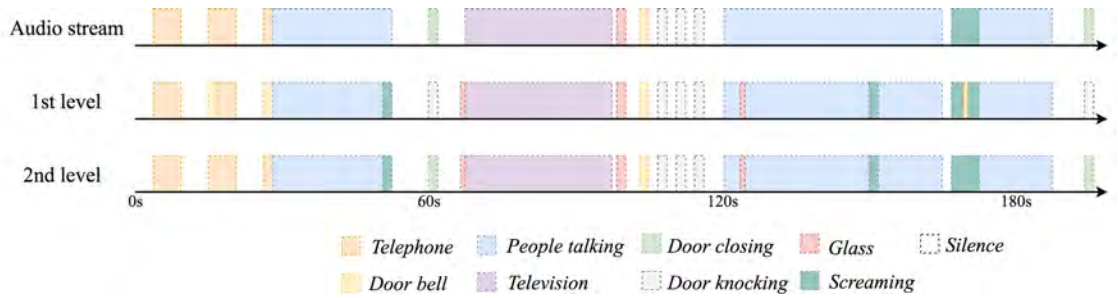

Figura 18: Comparación entre los resultados obtenidos en cada capa del clasificador.

- (0.97, 1) Test excelente.

En este caso, únicamente la clase de *Door closing* es la única que tiene un test malo.

8. *PRC Area*: En este caso, lo que se intenta calcular es la *precision* respecto a la *sensitivity*. Las curvas *PRC* suelen dar más información que las curvas *ROC* en data sets desbalanceados, como sería el caso de este problema.

Tras haber analizado los resultados de esta capa de clasificación en tiempo real (la que analiza únicamente el *stream* de audio de cada sensor individualmente), se procede a ver la siguiente capa del clasificador.

5.4 CLASIFICACIÓN EN LA *high level event analysis layer*

Como se hace comúnmente en este tipos de trabajos [65, 66, 67], para mostrar el funcionamiento del sistema de forma ilustrativa, se ha creado un audio sintético de datos que contiene todos los posibles eventos del data set. Se pretende que este audio emule el comportamiento de un paciente de la fundación desde que se levanta hasta que se va a dormir pero reduciéndolo a un tiempo más corto (únicamente queremos ver cómo se comporta el clasificador). Así pues, el fichero de audio sintético tiene una duración total de 200 segundos. Las muestras se han seleccionado cuidadosamente del data set explicado en el Cuadro 1 intentando estresar al clasificador para saber cuáles son sus fortalezas y sus debilidades.

Explicando el funcionamiento del sistema, los pasos a seguir són primero obtener los MFCC de cada fragmento acústico en la capa de sensórica, se envían al clasificador con la ANN y se envía el resultado de la red neuronal a la *High Level Event Analysis Layer*.

En la Figura 18 se muestran los resultados obtenidos en ambas capas del clasificador.

En la figura, se puede observar que, primero, alguien llama a casa del paciente (es decir, un evento del tipo *Telephone*), y el teléfono suena tres veces antes de que alguien conteste en casa del paciente. Se puede observar la clara periodicidad del evento del teléfono. Así, aunque el primer nivel del clasificador confunda algunas partes del teléfono con el evento *Door bell*— analizando con

la red neuronal únicamente los MFCC de una ventana de 100 ms—, el segundo nivel del clasificador puede identificar que realmente la clase final es un *Telephone* a base de considerar el patrón temporal de la señal.

Lo siguiente que ocurre es que el paciente coge el teléfono y habla (es decir, un evento del tipo *People talking*) durante unos segundos con el interlocutor. En este caso, el primer nivel del clasificador confunde parte de la conversación por teléfono con un evento del tipo *Scream*, y el segundo nivel del clasificador no es capaz de corregirlo. Aún y así, como el paciente sigue hablando por teléfono, este evento de tipo *Scream* no generaría una alarma; ya que no es muy importante que haya un grito si el paciente está hablando con otra persona.

Unos segundos después de la llamada, se cierra una puerta (evento del tipo *Door closing*). El primer nivel del clasificador confunde este evento con un evento del tipo *Door knocking*, pero el segundo nivel del clasificador puede corregirlo viendo que no hay más fragmentos etiquetados con esta etiqueta en un periodo de tiempo corto.

El siguiente evento que ocurre es que el paciente enciende la televisión (evento *Television*) y se detecta un evento del tipo *Glass breaking*. Este evento de cristal roto no es real (en realidad debería haberse detectado la clase de *Silencio*), pero ninguno de los niveles del clasificador puede detectarlo y corregirlo. Sin embargo, el nivel de confianza del clasificador (es decir, la probabilidad de que el evento pertenezca a la clase que sale de la red neuronal) es lo suficientemente baja (53 %) como para hacernos pensar que con una fuente acústica adyacente—debemos recordar que todas las áreas están cubiertas por más de un WAS— podría hacer que el segundo nivel de clasificación pudiera corregir el evento.

A continuación un evento del tipo *Scream* seguido de una secuencia de eventos *Door bell* y *Door knocking* se detectan de forma correcta en la primera capa del clasificador y después de confirman en la segunda capa del clasificador.

En el segundo 120, una conversación (evento del tipo *People talking*) empieza y produce dos eventos fantasmas (eventos que no tienen nada que ver con el evento que realmente está sonando). Concretamente, los eventos fantasmas que se detectan son de *Glass breaking* y *Screaming*. El segundo nivel de clasificación no es capaz de corregirlo. Sin embargo, no saltan alarmas de emergencia ya que no se percibe un cambio abrupto en el contexto del escenario (por lo tanto, el sistema asume que todo está bajo control).

Seguidamente, tras unos segundos de *Silencio*, empieza una conversación con un grito (*Scream*) parcialmente reconocido como un *Door bell*; pero la segunda capa del clasificador puede corregirlo. Finalmente, se cierra una puerta (*Door closing*) que se reconoce como un *Door knocking*, pero el segundo nivel del clasificador puede corregirlo viendo la evolución temporal del evento.

Finalmente, se puede afirmar que tras añadir esta segunda capa del clasificador la *accuracy* general del sistema aumenta a 93.27 % y el índice de *F1-measure* se incrementa hasta un 88.14 %. Nótese que aunque en la figura únicamente muestra un vector entregado a la segunda capa del clasificador (correspondien-

te a un único WAS), en realidad se entregarían los vectores correspondientes a todos los WAS del piso en cuestión, lo que incrementaría tanto la *accuracy* como la *F1-measure* del sistema.

6 ARQUITECTURA DEL SISTEMA

Resumen. En este capítulo se presenta la arquitectura de dos capas propuesta para el sistema. Además, también se relaciona la arquitectura con el sistema clasificador y se hace un estudio de la escalabilidad del mismo.

6.1 INTRODUCCIÓN

En este capítulo se detalla la arquitectura de sistema propuesta para instalar en la *Fundació Ave Maria*, teniendo en cuenta que parte del sistema se instalará en los pisos individuales y parte del sistema se instalará en la sede central de la fundación.

En esta arquitectura se presentará una capa de *fog computing*, se presentarán los sensores a instalar y donde se implementará el sistema automático de clasificación.

6.2 ARQUITECTURA DE SISTEMA

Partiendo de la arquitectura propuesta en [21] para escenarios *indoor*, y queriendo extender el escenario a las áreas residenciales de la *Fundació Ave Maria*, se requiere que la arquitectura:

1. Soporte el proceso múltiples *streams* de audio simultáneamente.
2. Mejore la etapa de extracción de características y procesado de señal para evitar cuellos de botella en la red [68], donde una forma de hacerlo es encontrar un punto medio entre el procesamiento local de datos y la cantidad de datos a procesar [69].
3. Mejore la etapa de clasificación de eventos acústicos para maximizar la fiabilidad de los resultados del clasificador con una mínima interacción humana [70], ya que en el caso de áreas tan grandes como las de la *Fundació Ave María* se necesitarían demasiados recursos si se tuviera que atender a los pacientes y también estar constantemente comprobando el sistema de detección automático.

Actualmente, las redes WiFi ofrecen un buen rendimiento teniendo en cuenta sus costes de infraestructura, consumo y ancho de banda disponible [71], especialmente en el contexto de las soluciones de AAL. Así pues, es habitual que se utilicen este tipo de redes para conectar la capa de sensores (por donde se adquieren los datos) con la capa de análisis/almacenjae/procesamiento o de computación en las aplicaciones de AAL [72].

Normalmente [43, 44, 40, 46, 45, 73, 74], la capa de computación no se encuentra físicamente cerca de donde haita el paciente (es decir, se utiliza una arquitectura de *cloud computing*), por lo que Internet se utiliza para unir los dos entornos— lo que limita el ancho de banda disponible, pone al sistema en riesgo a cyberataques [75] y añade un retraso significativo al proceso de detección de eventos.

Cabe destacar que los efectos negativos de estos problemas (ancho de banda limitado, cyberseguridad y retraso) crecen cuando el número de pacientes y/o el área a monitorizar incrementan, ya que los datos incrementan de forma proporcional.

Así pues, para satisfacer los requerimientos mencionados, se presenta la arquitectura distribuida de *fog-computing*[44] de la Figura 19 para (1) recolectar datos a partir de múltiplos sensore, (2) implementar la *Real-time Early Event Detection Layer* y (3) refinar el sistema de deteccion automática de eventos mediante una *High Level Event Analysis Layer* .

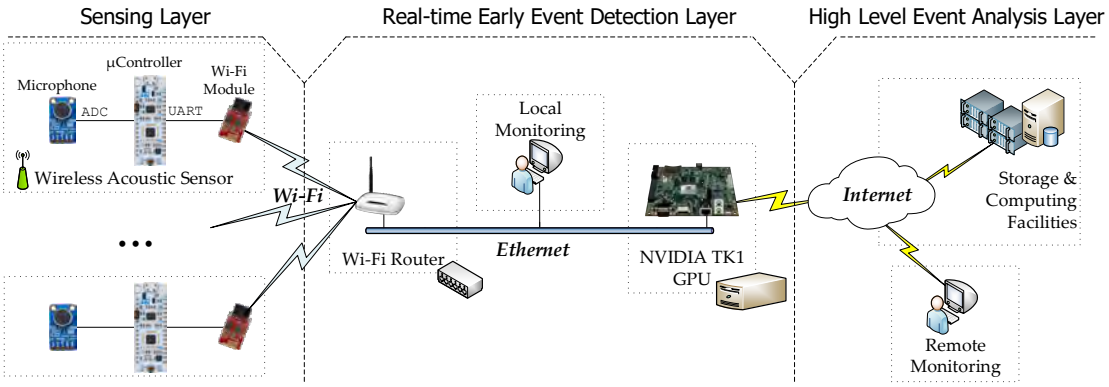

Figura 19: Topología de la red propuesta.

Así pues, las capas de la arquitectura son las siguientes:

1. **Sensing Layer**. Se compone de todos los sensores acústicos que se implantan en el área donde los pacientes son monitorizados. Así pues, cada sensor debe llevar a cabo las siguientes tareas:

   - Muestrear los *streams* de audio a 44.1 ksps.
   - Extraer las características de las muestras de audio (es decir, construir el vector de características). Esto se hace para evitar tener que mandar los *streams* de audio enteros, lo que haría que la red se saturase.
   - Mandar los vectores de características a un *wireless hub*.

   Esto se llevará a cabo mediante *hardware* de bajo coste —de aproximadamente 20€—: un micrófono electret con un amplificador MAX9814 ensamblados en una pequeña placa, una plataforma de desarrollo de 32 Núcleos con un µControlador STM32L432KC ARM cortex-M y un módulo WiFi basado (ESP8266).

2. **Real-time Early Event Detection Layer**.

   Cada vector de características generado en la capa de sensórica se analiza en una GPGPU NVIDIA Jetson TK1 [76]. El motivo de elegir este elemento como parte de la arquitectura es su capacidad de analizar *streams* de datos en paralelo [21]. Así pues, la GPGPU contendrá el modelo de una ANN entrenada que, para cada vector de características, proporcionará un vector de 9 componentes que se mandará al *High Level Event Analysis Layer*.

   Cada componente del vector de salida corresponderá a la probabilidad de que la ventana corresponda a un evento determinado (*Door knocking*, *Scream*, *People talking*, *Silence*, *Door closing*, *Telephone*, *Door bell* y *Glass breaking*). A este vector también se le conoce como vector de confidencia de clasificación.

   Esto puede verse como una etapa preliminar de qué evento está sucediendo (mirando la componente con valor de probabilidad más alta del vector), ya que no se tiene en cuenta el dominio temporal de los eventos (cabe recordar que se ha dicho que, por ejemplo, múltiples vectores consecutivos etiquetados como *Door closing* pueden indicar en realidad un evento de *Door knocking*).

   Es importante entender que los resultados de esta capa no son del todo fiables, así que no debería tratarse una única etiqueta como resultado final sin tener en cuenta la evolución temporal; sin embargo, utilizar estas decisiones preliminares podría ser útil para aquellos eventos que requieren asistencia inmediata (como *Scream* o *Glass breaking*) teniendo en cuenta que todas las áreas están normalmente cubiertas por más de un WAS y, por lo tanto se pueden analizar concurrentemente múltiples *streams* para ver si un evento se ha detectado mediante diferentes sensores adyacentes.

3. **High Level Event Analysis Layer**. Esta segunda capa de clasificación de eventos tiene dos propósitos principales. Por una parte, el sistema analiza los eventos acústicos teniendo en cuenta su contexto (es decir, teniendo en cuenta los eventos acústicos que han ocurrido en las ventanas acústicas anteriores) para filtrar eventos que ocurren en una única ventana y por lo tanto son propensos a ser eventos fantasmas (por ejemplo un evento de *People talking* que ocurra únicamente durante una ventana). Por otra parte, también tiene en cuenta los eventos acústicos de sensores adyacentes. En este sentido, aquellos eventos clasificados con baja probabilidad pero identificados en diferentes WAS, pueden ganar relevancia en este nivel de clasificación. Para poder llevarse a cabo, estas ventanas recolectadas en la etapa de *Real-time Early Event Detection Layer* se concatenan y comparan con distintas normas o casos guardados en memoria [77] (es decir, se implementa un CBR). Finalmente, esta capa genera alarmas definidas por el usuario cuando se detectan ciertos eventos acústicos. Estas alarmas las definen los expertos según las necesidades del momento (por ejemplo, se

puede generar una alarma si la televisión se enciende entre las 3 y las 6 de la mañana). Este CBR se implementará en un *cloud*.

Resumiendo, el proceso y la computación de los datos se ha dividido como se hace en los sistemas clásicos de computación de las arquitecturas de *fog computing*: los WAS extraen las características de cada ventana de los *streams* de audio, posteriormente se hace una clasificación automática de eventos acústicos en la GPGPU y, finalmente, las etiquetas se entran en un CBR para obtener el resultado final del clasificador en una infraestructura *cloud*.

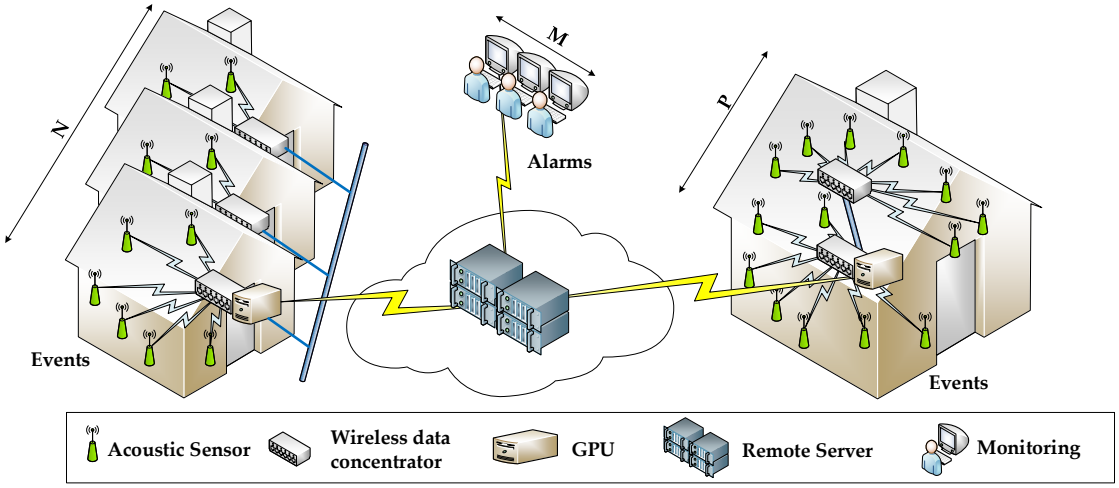

Figura 20: Arquitectura propuesta particularizada para la *Fundació Ave Maria*.

Como se muestra en la Figura 20, la arquitectura propuesta encaja con la *Fundació Ave Maria*. En cada edificio se debe poner un *wireless hub* para proporcionar conectividad WiFi. Además, para la zona de residencia de la fundación se pueden poner múltiples *wireless hubs* interconectados mediante *Ethernet* con la finalidad de expandir el área de cobertura del WiFi.

Cada WAS utiliza su red WiFi asociada para mandar los datos hacia la GPGPU. Subsecuentemente, cada *Graphics Processing Unit* (GPU) envía la salida de la capa de detección de eventos en tiempo real al servidor remoto que se encuentra en el *cloud* por Internet.

Además, los cuidadores pueden monitorizar las alarmas y el estado de los pacientes conectándose a los servidores remotos o a las GPGPU locales.

En este sentido, el sistema podrá escalar como se comenta a continuación:

1. El número de edificios adyacentes N podrá escalar fácilmente siempre que haya suficiente ancho de banda disponible en los enlaces *Ethernet*. Cabe recordar que por *Ethernet* no se mandan los *streams* de audio enteros, sino que únicamente se mandan los vectores de características que contienen cada uno los 13 MFCC de cada ventana de 100 ms.

2. El número de casas P (es decir, número de GPGPUs que entregan los datos a los servidores remotos) también pueden crecer dependiendo de la

capacidad de la infraestructura *cloud*, que a su misma vez puede crecer dependiendo del número de P conectadas.

3. El número de usuarios M monitorizando los pacientes también podría crecer siempre que el *cloud* y la conexión a internet puedan soportar más conexiones. Cabe recordar que, en esta etapa de *cloud*, en la red únicamente habrá las etiquetas finales de eventos y los *time-stamps* de cada ventana.

6.3 RELACIÓN ENTRE LA CLASIFICACIÓN Y LA ARQUITECTURA

Esta sección explica el ciclo de vida de los datas cuando pasan por la arquitectura.

El algoritmo propuesto consiste de tres diferentes fases que pueden asociarse a las tres capas de la arquitectura propuesta (ver Figura 19). Como se puede ver en la Figura 21, estas fases son las siguientes: extracción de características (o *feature extraction*), clasificación a tiempo real (o *real-time classification*) y generación de alarmas (o alarm triggering).

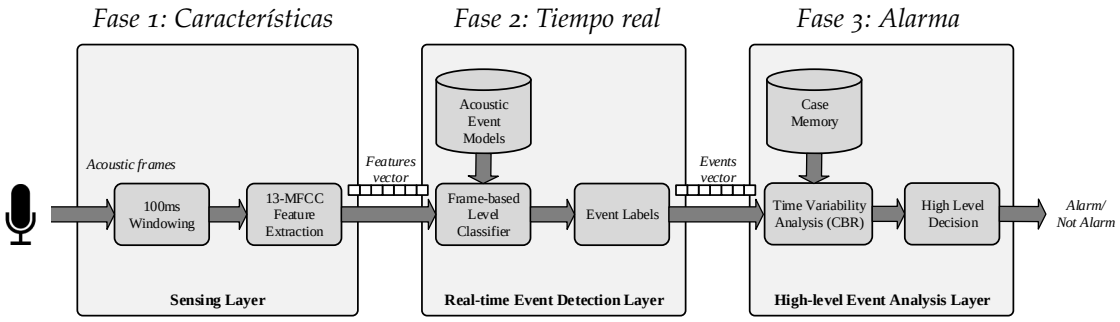

Figura 21: Diagrama de bloques del sistema propuesto.

A continuación se describe cada bloque:

Fase 1. Extracción de características. Se lleva a cabo en la *Sensing layer* de la Figura 19, y consiste en un proceso de procesado de la señal para obtener las características principales de la señal. En este caso, los coeficientes son los 13 MFCC. Una vez se obtienen estos coeficientes cada 50 ms (cabe recordar que son ventanas de 100 ms con un *overlap* del 50 %), se entregal a la capa de clasificación a tiempo real.

Fase 2. Clasificación a tiempo real. Aquí se encuentra la ANN de cinco capas que ya se ha explicado. Así pues, en esta capa se cogen los 13 MFCC de la capa de sensores y se comparan con el modelo de la ANN, de la cual sale un vector de 9 componentes con la probabilidad de cada evento acústico.

Fase 3. Generación de alarma. Se lleva a cabo en la *High-level Event Analysis Layer* y se implementa con un CBR [77]. En esta etapa se lleva a cabo la decisión de si se genera una alarma o no. Más específicamente, los vectores

de 9 componentes de almacenan en registros de 900 *buckets* (para poder tener una visión temporal de los eventos que suceden). Además, para poder tener también una visión de varios sensores adyacentes, los *buffers* asociados a los *streams* de datos de WASs que se encuentran físicamente cerca se promedian—también se puede hacer un promedio ponderado según la distancia entre sensores. Una vez está el *buffer* lleno, se compara el *buffer* contra toda la memoria de casos—que se ha llenado sintéticamente— para poder obtener otro vector de 9 componentes binario en el que se dice si el evento ha ocurrido o no.

Cuando el vector de componentes binario, se analiza con las reglas que han creado los cuidadores teniendo en cuenta el momento del día en el que suceden los eventos. En el caso en el que se genere una alarma, también se almacena por si hubiera repeticiones de eventos significativos (por ejemplo, si se da un evento de *Screaming* cada cinco minutos).

Este proceso puede hacerse en paralelo para cada *stream* de datos asociado a los WASs de la *Fundació Ave Maria*. Así pues, las fases 1 y 2 re-calculan sus salida para cada *frame* de 100 ms. La fase 3 también recomputa su salida cada para cada *frame*, pero solo da una salida cada 10 s.

6.4 ESCALABILIDAD DE LA PLATAFORMA PROPUESTA

La escalabilidad es uno de los mayores retos que existe en el diseño de plataformas AAL para soportar los servicios de tele-asistencia en escenarios de grande escala. Existen diferentes aproximaciones que integran múltiples *streams* de gran volumen (por ejemplo, vídeos), pero requieren infraestructuras de comunicaciones de alto coste —y, normalmente, ad hoc— [78]; lo que podría limitar el rendimiento del sistema cuando múltiples *streams* de datos tienen que analizarse de forma concurrente.

Así pues, la arquitectura de *fog computing* propuesta se ha diseñado con dos objetivos principales: (1) reducir la cantidad de datos a transferir (que la computación de los MFCC se mueva a la capa de sensórica es más barato que mover todos los *streams* de datos [42]) y (2) sacar partido a las infraestructiras existentes en los escenarios (el WiFi i las conexiones a Internet) para reducir el coste de despliegue.

Primero, y más importante, la cantidad de datos en *raw* que se deben transferir a la capa de sensórica para cada WAS y el *router* WiFi se puede calcular como:

$$\begin{aligned} \text{Bytes trans } WAS &= \frac{13 \text{ MFCCs}}{WAS * \text{Ventana}} * \frac{1 \text{ Ventana}}{100 \text{ ms.} * 0{,}5} * \frac{8 \text{ Bytes}}{1 \text{ MFCC}} \\ &= 2\,080 \frac{\text{Bytes}}{\text{segundo} * WAS} \end{aligned} \tag{1}$$

Sin embargo, todos estos datos tienen que estar encapsulados; por lo que se utiliza la cabecera *User Datagram Protocol* (UDP) con los mínimos campos posibles sobre *Ethernet*. El total de bytes de cabecera será, pues, de 52 *Bytes* desglosados en 24 *Bytes* de *Ethernet*, 20 *Bytes* de la trama *Internet Protocol version 4* (IPv4) y 8 *Bytes* de la trama UDP.

Así pues, el total de *Bytes* enviado (teniendo en cuenta que cada ventana de 100 ms se envía cuando se ha parametrizado, por lo que se envía cada 50 ms) es:

$$\begin{aligned} \text{Bytes reales WAS} &= \left(52\ B + \frac{13\ \text{MFCCs}}{WAS * \text{Ventana}} \frac{8\ \text{Bytes}}{1\ \text{MFCC}}\right) * \frac{1\ \text{Ventana}}{100\ \text{ms.} * 0{,}5} \\ &= 3\ 120\ \frac{\text{Bytes}}{\text{segundo} * WAS} \end{aligned} \tag{2}$$

Aunque este número no es muy grande comparado con el ancho de banda típico que hay en las redes WiFi domesticas (unos 54 *Mbps*) —y aunque a priori parezca que se pueden conectar miles de WAS a la red— se debe tener en cuenta que:

1. La mayoría de *routers* domésticos WiFi no aceptan más de unos 255 dispositivos conectados simultáneamente.

2. El hecho de estar mandando tramas cada poco tiempo (en este caso cada 50 ms) reduce el rendimiento de la red WiFi considerablemente [79], por lo que de los 54 *Mbps* finalmente se limitan a unos 21.6 *Mbps*.

3. El retraso incrementa con el número de WAS conectados a la red WiFi. Sin embargo, los análisis que se han hecho han revelado que se podrían conectar de forma segura hasta unos 60 WASs de forma simultánea sin percibir una degradación significativa del sistema. Por lo tanto, habría un tráfico constante de datos de, aproximadamente, 1.43 *Mbps* por cada *access point*.

Por otra parte, también se han calculado el número de *access points* que se pueden añadir a la red teniendo en cuenta el ancho de banda disponible por la red cableada *Ethernet* que conecta la GPGPU y el *router*. Se puede asumir que la red *Ethernet* tendrá un ancho de banda típico y suficiente de entre 100*Mbps* y 1*Gbps*. El cálculo es el siguiente:

$$\begin{aligned} \text{Bytes a transferir}_{\text{GPU}} &= \frac{9\ \text{etiquetas}}{WAS * \text{ventana}} * \frac{1\ \text{ventana}}{100\ \text{ms.} * 0{,}5} * \frac{4\ \text{Bytes}}{1\ \text{etiqueta}} \\ &= 720\ \frac{\text{Bytes}}{\text{segundo} * WAS} \end{aligned} \tag{3}$$

En este caso, hay que volver a tener en cuenta las cabeceras de UDP sobre *Ethernet*, por lo que los datos a transferir serían:

$$\begin{aligned} \text{Bytes reales}_{\text{GPU}} &= \left(52Bytes + \frac{9 \text{ etiquetas}}{WAS * ventana} * \frac{4 \text{ Bytes}}{1 \text{ etiqueta}}\right) * \frac{1 \text{ ventana}}{100 \text{ ms.} * 0{,}5} \\ &= 1\,760 \frac{\text{Bytes}}{\text{segundo} * WAS} \end{aligned} \tag{4}$$

También se debería tener en cuenta que el número de WASs estará limitado por el ancho de banda disponible en la conexión a Internet hacia el *cloud*. Por ejemplo, si se escoge un *Asymmetric Digital Subscriber Line* (ADSL) tradicional de 2*Mbps* y unos 100*Mbps* para la *Local Area Network* (LAN), se podrían soportar teóricamente unos 449 WAS aproximadamente, lo que requeriría un total de 40 *access point* WiFi.

En cuanto al coste computacional de la GPGPU, el coste aproximado de la ANN es de unos 14.2 *KFLOPS* por cada WAS, que es menor que la capacidad de computo de la GPGPU NVIDIA Tegra TX1 propuesta (que tiene una capacidad de 13.6 *GFLOPS* por *core* según [80]).

Así que se puede ver que el factor más limitante de la arquitectura propuesta en cuanto a escalabilidad será la conexión a Internet. Sin embargo, este hecho podría solventarse añadiendo más conexiones a Internet en los escenarios cuando fuera necesario.

6.5 TOLERANCIA A FALLOS

También es importante tener en cuenta que el sistema a veces puede fallar, por lo que es importante analizar la tolerancia a fallos del mismo.

1. Fallos en la *sensing layer*: esto implicaría que un WAS deja de funcionar o pierde su conexión con el *router* WiFi. Este fallo no sería crítico gracias a la redundancia de sensores que hay (un área está cubierta por más de un sensor). Sin embargo, si el *router* WiFi dejara de funcionar, todos los datos asociados a los WAS conectados se perderían. En esta situación, como la GPGPU detectaría que no se están recibiendo *streams* y generaría una alarma. Una alternativa sería que los WASs utilizaran una conexión alternativa mediante un teléfono móvil estándar para poder conectarse a la GPGPU mediante Internet; ya que la cantidad de datos a enviar tampoco sería muy elevada (como se muestra en la ecuación 2).

2. Fallos en la GPGPU: En este caso, los fallos podrían solventarse a base de redirigir los *streams* de datos hacia otra GPGPU por el *Ethernet* local (siempre que haya más GPGPUs en la misma LAN) o por Internet.

3. Fallos en la conexión a Internet: Esto podría hacer que un edificio entero quedara aislado. sin embargo, en este caso también se podrñia utilizar

un teléfono móvil para enviar los datos por internet (cabe recordar que el número de *Bytes* a transferir no es muy elevado tal y como se puede ver en la ecuación 3). Por otra parte, los datos de la capa de tiempo real siempre estarían disponibles para los cuidadores de la *Fundació Ave Maria* que estuvieran conectados en la misma red LAN en la que están los pacientes.

4. Fallos en la *High-level Event Analysis Layer*: este tipo de errores no deberían suceder, ya que al contratar el servicio *cloud* a un proveedor de servicios externo se tiene un gran nivel de QoS negociado.

6.6 PRIVACIDAD DEL PACIENTE

Un aspecto muy importante en las aplicaciones de AAL es el de la privacidad del paciente. En este caso, la arquitectura distribuida propuesta ha tenido en cuenta la privacidad ya que los *streams* de audio en *raw* nunca se mandan por la red. Se puede apreciar que en la primera capa (antes de mandar nada por WiFi) ya se calculan las características y sún únicamente los MFCC los que se mandan por la red. Además, estos y los eventos ya detectados pueden mandarse de forma encriptada por la red si se desea.

Estos hechos, junto a la naturaleza no-invasiva del sistema propuesto en el que no se graban vídeos del paciente, hace que los pacientes puedan estar seguros y tranquilos sin tener que llevar siempre encima dispositivos antiestéticos e incómodos [78, 81].

7 CONCLUSIONES

Resumen. En este capítulo se presentan las conclusiones obtenidas a lo largo del TFM. A continuación, se listan los problemas encontrados durante el desarrollo del mismo y las líneas de futuro que se quieren seguir. Finalmente, se hace una evaluación del coste económico y temporal del proyecto.

7.1 INTRODUCCIÓN

Después de haber diseñado y desarrollado una prueba de concepto para el sistema de AAL propuesto; y haber hecho una evaluación experimental del mismo (es decir, después de haber hecho la detección de eventos acústicos mediante técnicas de *machine learning*), en este capítulo se presentan los conceptos aprendidos y se discuten los retos que plantea éste problema y las líneas de futuro que se pretenden seguir de ahora en adelante.

7.2 CONCLUSIONES

Se puede concluir que los objetivos del proyecto se han cumplido satisfactoriamente. A continuación, se enumerarán los objetivos y se comentará en qué sentido se han cumplido.

1. **Hacer un estudio del estado del arte que existe actualmente en el ámbito del AAL**: Tal y como se puede ver reflejado en el Capítulo 2, se ha podido hacer un estudio exhaustivo de muchísimos proyectos de investigadores de alrededor de todo el mundo para poder tener en cuenta a escala global las tendencias en el ámbito del AAL.

2. **Analizar qué tipo de sonidos domésticos pueden dar información sobre el paciente para ver si todo va bien o si hay que generar una alarma y generar un *data set***: En este sentido, se han podido definir los sonidos de interés y se ha generado un *data set* completo. Esto puede verse reflejado en el Capítulo 4.

3. **Analizar qué parámetros acústicos (o *features*) modelan mejor los sonidos domésticos dentro de un conjunto finito de *features*, como primer estudio**: En el Capítulo 5 se ha comentado que para obtener las características de los ficheros de audio se han estudiado los parámetros MFCC y GTCC. Al final, tras hacer los experimentos se ha podido comprobar que siguiendo el *baseline* de MFCC ya se obtenían los resultados deseados.

4. **Crear un sistema de aprendizaje automático que permita clasificar los distintos tipos de sonidos**: También en el Capítulo 5 se ha explicado toda la parte de experimentación para la creación del sistema automático de clasificación. Al final, se ha hecho el sistema de dos capas (o niveles) en el que se ha diseñado una red neuronal (ANN) para la clasificación a tiempo real y, posteriormente, una capa de CBR que integra los datos recogidos por varios sensores que han sido pre-clasificados por la capa de tiempo real. Gracias a este segundo nivel se puede obtener más precisión de resultados y, por lo tanto, más fiabilidad en el sistema..

5. **Diseñar una arquitectura distribuida que permita que el proyecto de AAL pueda ser de gran escala para satisfacer las necesidades de la *Fundació Ave Maria***: En este sentido, en el Capítulo 3 se ha estudiado en profundidad qué necesidades tendría la fundación para poder diseñar el sistema y, posteriormente, en el Capítulo 6 se ha propuesto una arquitectura (concretamente, una arquitectura de *fog computing* que cumple con todos los requisitos de la fundación.

6. **Consolidar y ampliar los conocimientos actuales en el ámbito de procesado de audio, programación, minería de datos e inteligencia artificial; así como también se quieren ampliar los conocimientos de las arquitecturas distribuidas y de *fog computing***: Este objetivo se ha ido cumpliendo mientras se iba desarrollando el trabajo, pues se han tenido que leer muchos artículos científicos. Además, al haber de programar los algoritmos también se han entendido más en profundidad.

También es importante remarcar que, además de los objetivos iniciales, al final se ha terminado aprendiendo muchísimo más de lo esperado. Además, se han adquirido muchos conocimientos en el ámbito de la investigación según el método científico.

7.3 PUBLICACIÓN DE RESULTADOS

Tras llevar a cabo todo el trabajo, se ha procedido a escribir un artículo científico y se ha publicado junto a los Doctores Joan Navarro, Rosa Maria Alsina-Pages y Marcos Hervás en la revista *Sensors*, una revista de acceso abierto de *Multidisciplinary Digital Publishing Institute* (MDPI) (ISSN 1424-8220).

Esta revista está ubicada en el segundo cuartil (Q2) y tiene actualmente un factor de impacto de 2.745 (en 2017) y un factor de impacto en los últimos 5 años de 3.014 (2017).

El artículo completo puede leerse en http://www.mdpi.com/1424-8220/18/8/2492, donde tiene a fecha de 21 de agosto de 2018 260 lecturas completas y 137 descargas. Esto puede comprobarse en la Figura 22.

Por otra parte, en la Figura 23 también se puede ver que las lecturas que ha tenido el artículo han sido a nivel mundial (no únicamente a nivel global).

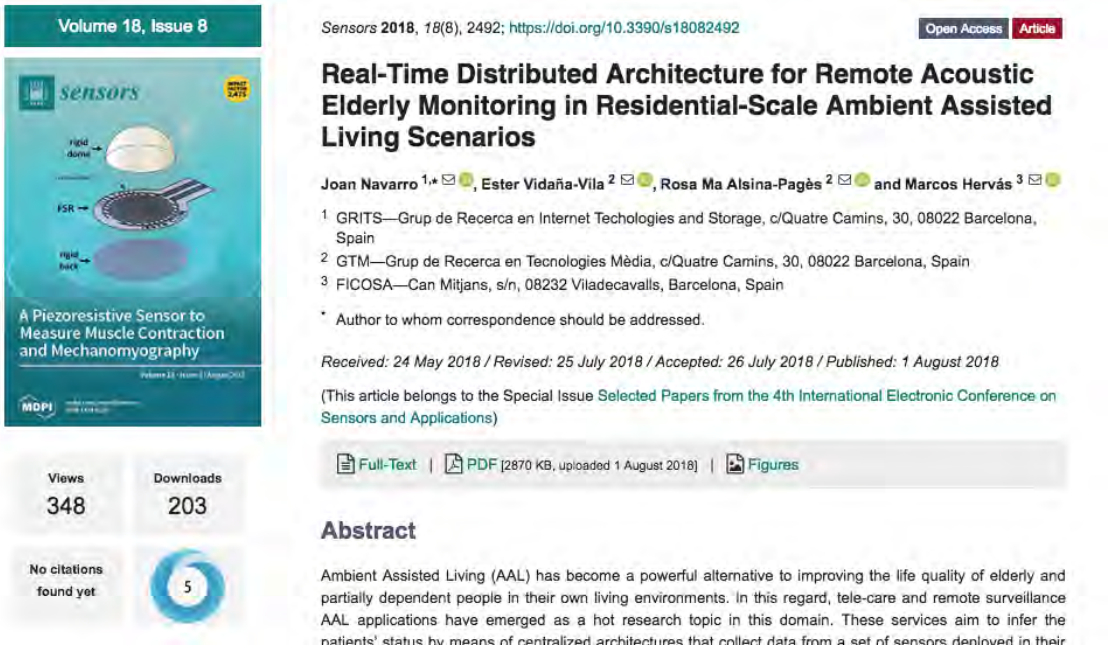

Figura 22: Página web de la revista *Sensors*.

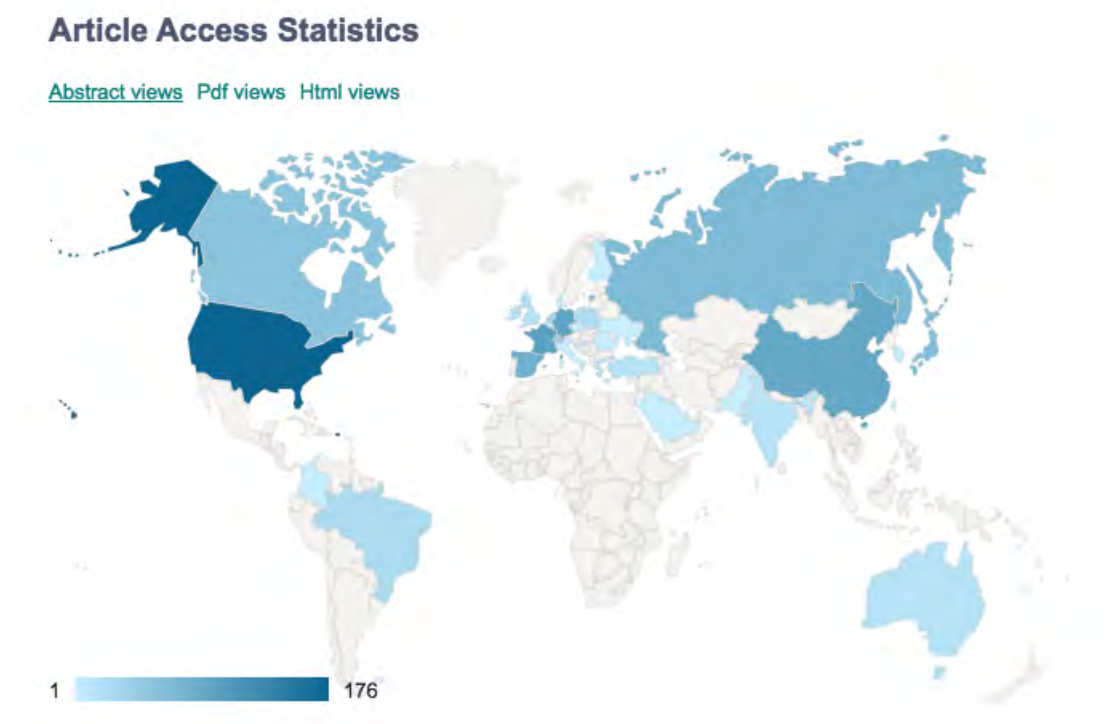

Figura 23: Orígenes demográficos de las lecturas del artículo publicado en la revista *Sensors*.

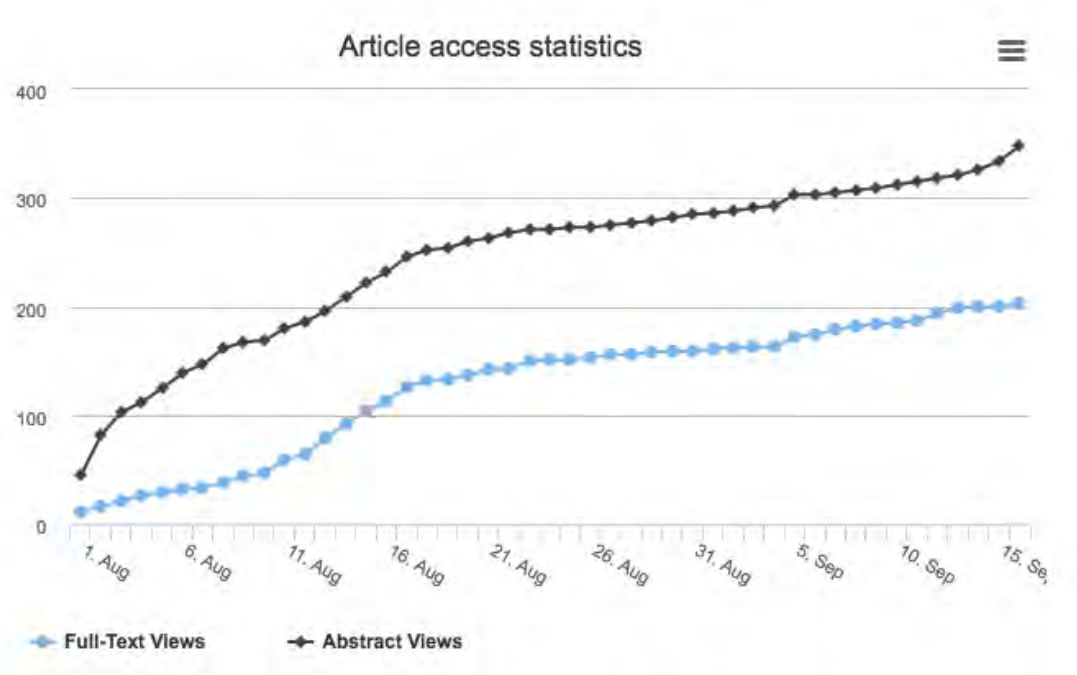

Figura 24: Evolución temporal de las lecturas del artículo en la revista *Sensors*.

Finalmente, en la Figura 24 se puede ver cómo ha sido la evolución de las lecturas del artículo (tanto del *abstract* como del artículo completo) desde que se publicó.

7.4 PROBLEMAS ENCONTRADOS

El principal problema que se ha encontrado en este trabajo ha sido el del desbalanceo del *data set*. Es decir, que hay clases del *data set* que contienen más muestras que otras. Es sabido que los clasificadores no suelen tener porcentajes de acierto perfectos cuando hay diferencias significativas en el número de muestras de cada clase. Sin embargo, ha sido imposible conseguir más datos de los eventos de interés de este trabajo (cabe recordar que en este trabajo se utiliza un *data set* de 7 116 segundos), pero para obtener este *data set* se han tenido que analizar más de 20 horas de audio de los eventos de interés. Así pues, para intentar minimizar este problema, se propone que cuando un prototipo ya esté en funcionamiento en la *Fundació Ave Maria* grabe las muestras de los eventos detectados para poder ir así ampliando el *data set* y haciéndolo más específico para los escenarios (por ejemplo, es evidente que el sonido de una puerta de madera no será la misma que el sonido de una puerta de cristal, por lo que se debe adaptar el *data set* al escenario).

Otro factor que se podría llevar a cabo sería hacer más *data augmentation*, pero como se ha comentado no se cree que esto proporcionara una mejora significativa ya que ya se ha aplicado *data augmentation* en el *data set* [82].

7.5 LÍNEAS DE FUTURO

Una línea de futuro a cubrir seria la de añadir muestras sintéticas en el *data set* para contemplar los casos en los que más de un evento pasa al mismo tiempo (por ejemplo, que llamen al teléfono y a la puerta en el mismo momento).

Por otro lado, también hay ciertos retos en el ámbito del procesado de señal que se pueden mejorar. En esta prueba de concepto para el diseño de un sistema AAL en un entorno residencial, únicamente se han tenido en cuenta los ficheros de audio provenientes de dos sensores WAS (que realmente han sido audios sintéticos) para poder medir y testear la *accuracy* del clasificador. Sin embargo, se sabe que en un futuro —cuando la prueba de concepto pase a implementarse en un caso real— se tendrá que mejorar y terminar de ajustar el algoritmo para obtener resultados más fiables.

En este sentido, hay varios grandes retos en la parte de procesado acústico del señal que se deben solventar.

1. El primer reto consiste en la selección del tipo y número de coeficientes cepstrales para hacer la extracción de características del algoritmo. En este trabajo, se han escogido los 13 MFCC partiendo de un *baseline* [83] y por coherencia teniendo en cuenta el trabajo que se ha hecho en esta línea dentro del grupo de investigación [21]. Además, a partir de esta base se hizo un *grid search* para ver si variando un poco el número de coeficientes o el tipo de los mismos incrementaba la *accuracy* del sistema. Sin embargo, se vio que los resultados no eran mucho mejores; pero tal vez una vez el sistema esté realmente implementado podría ser interesante volver a hacer un test con varios tipos de coeficientes proponiendo un sistema híbrido para el bloque de procesado del señal [84].

2. El segundo reto es el de procesar múltiples *streams* de datos provenientes de múltiples fuentes como si fueran una sola. En este sentido, es muy importante que la distribución de sensores en casa sea redundante (es decir, que todas las áreas de la casa o el área residencial estén cubiertos por un mínimo de dos sensores). En esta prueba de concepto se ha visto que la redundancia podía incrementar la *accuracy* del clasificador. Sin embargo, se cree que un sistema más inteligente —más inteligente que el sistema que se ha propuesto en este trabajo— también incrementaría el porcentaje de acierto del sistema. En este sentido, la capa de alto nivel (*High-level Event Analysis Layer*) debería refinarse para tener la capacidad de decidir si la alarma debe dispararse dependiendo de lo que ha captado el sensor en concreto y sus vecinos. Esto sería similar a la técnica de *Variable Neighbor Search* que se propone en [37]. El inconveniente es que para que esto fuera posible se necesitaría (1) saber en qué lugar de la casa se encuentra exactamente el sensor y (2) tener que reconstruir toda la memoria asignada hasta el momento.

Por último, los requerimientos de fiabilidad del sistema entero deberían tenerse en cuenta a la hora de generar las alarmas, ya que aún existen limitaciones en el marco de la detección de eventos que hacen que la *accuracy* no sea del 100 %. Para mejorar la fiabilidad del sistema, en ciertas áreas de la casa como la cocina o el comedor donde los pacientes no suelen pedir tanta intimidad, se podrían instalar otro tipo de sensores como se comenta en [85]. Por ejemplo, se podrían poner cámaras para procesar *streams* de vídeo únicamente cuando hubieran emergencias. Sin embargo, antes de proponer esto para el sistema final tendría que ponerse en una balanza la *accuracy* del sistema contra la privacidad y el bienestar del paciente, ya que es muy probable que muchos pacientes no aceptaran el sistema si se instalaran cámaras en casa.

7.6 COSTE ECONÓMICO Y TEMPORAL DEL PROYECTO

Para terminar el proyecto, se realiza un análisis de las horas que han sido necesarias para llevar a cabo este proyecto:

Investigación en el ámbito de la clasificación de audio: 45 horas.

Investigación en el ámbito de la arquitectura del sistema: 45 horas.

Reuniones de seguimiento: 30 horas.

Investigación y puesta en marcha de la GPGPU: 65 horas.

Generación del *data set*: 20 horas.

Programación de los algoritmos de clasificación: 130 horas.

Generación de resultados: 70 horas.

Redacción del artículo en la revista de *Sensors*: 80 horas.

Revisión del artículo antes de su publicación: 15 horas.

Redacción de la memoria del proyecto: 150 horas.

Por lo tanto, el total de horas invertidas en este proyecto es de 650 horas.

En la Figura 25 se puede ver un gráfico con la proporción de cada tarea respecto al total del tiempo invertido.

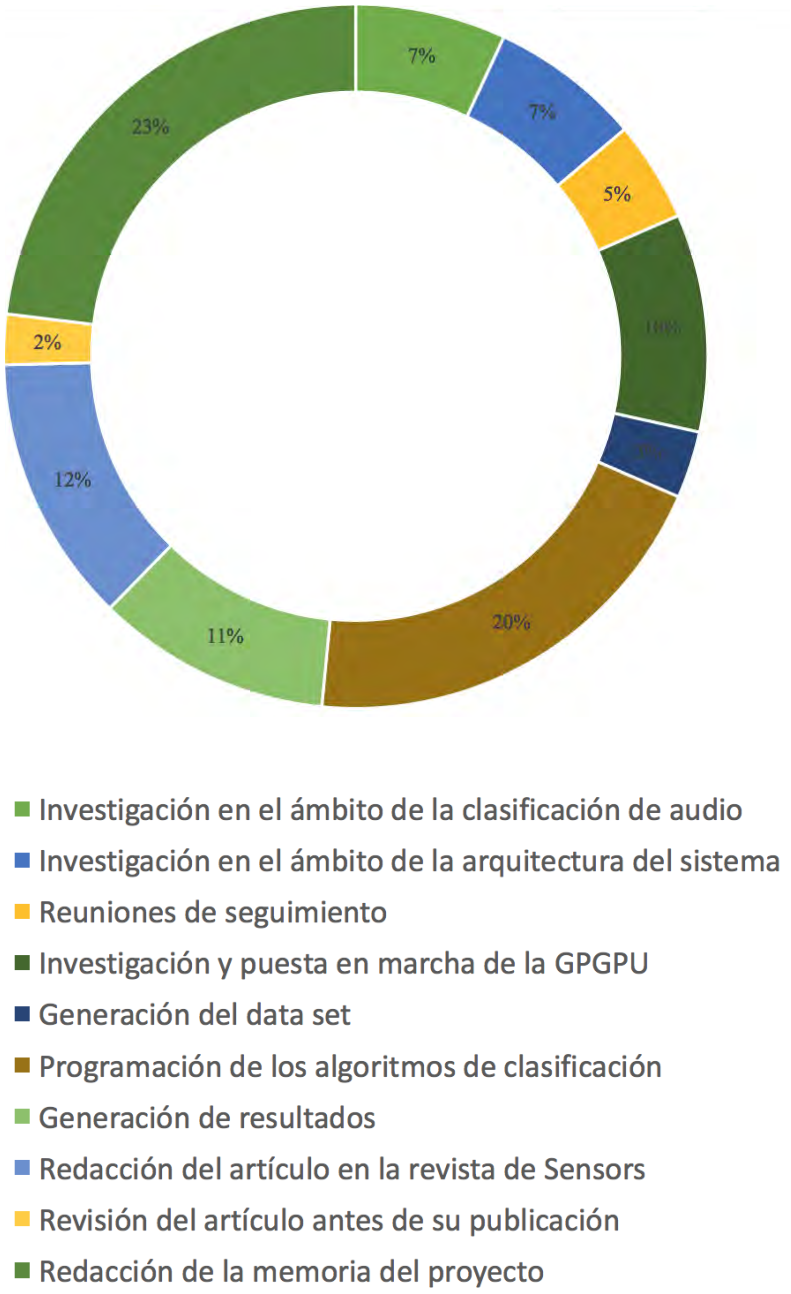

Figura 25: Porcentaje de dedicación a las diferentes tareas del proyecto.

BIBLIOGRAFÍA

[1] R Suzman y J Beard. «Global Health and Aging—Living Longer». En: *National Institute on Aging: Bethesda, MD, USA* (2015).

[2] Jaime Prats. *La esperanza de vida aumenta más de 40 años en un siglo*. https://elpais.com/politica/2015/02/26/actualidad/1424969363_446948.html (accessed on 3 Sep 2018).

[3] Michel Vacher, François Portet, Anthony Fleury y Norbert Noury. «Challenges in the processing of audio channels for ambient assisted living». En: *e-Health Networking Applications and Services (Healthcom), 2010 12th IEEE International Conference on*. IEEE. 2010, págs. 330-337.

[4] Parisa Rashidi y Alex Mihailidis. «A survey on ambient-assisted living tools for older adults». En: *IEEE journal of biomedical and health informatics* 17.3 (2013), págs. 579-590.

[5] Charalampos N Doukas e Ilias Maglogiannis. «Emergency fall incidents detection in assisted living environments utilizing motion, sound, and visual perceptual components». En: *IEEE Transactions on Information Technology in Biomedicine* 15.2 (2011), págs. 277-289.

[6] Andrey Temko. «Acoustic event detection and classification». En: *Ph.D Thesis, Universitat Politècnica de Catalunya, Barcelona, Spain* (2017).

[7] Lode Vuegen, Bert Van Den Broeck, Peter Karsmakers, Hugo Van Hamme y Bart Vanrumste. «Automatic monitoring of activities of daily living based on real-life acoustic sensor data: a˜ preliminary study». En: *Proceedings of the Fourth Workshop on Speech and Language Processing for Assistive Technologies*. 2013, págs. 113-118.

[8] Rosa Ma Alsina-Pagès, Francesc Alías, Joan Claudi Socoró y Ferran Orga. «Detection of Anomalous Noise Events on Low-Capacity Acoustic Nodes for Dynamic Road Traffic Noise Mapping within an Hybrid WASN». En: *Sensors* 18.4 (2018), pág. 1272.

[9] Chiara Bartalucci, Francesco Borchi, Monica Carfagni, Rocco Furferi, Lapo Governi, Alessandro Lapini, Raffaella Bellomini, Sergio Luzzi y Luca Nencini. «The smart noise monitoring system implemented in the frame of the Life MONZA project». En: *Euronoise 2018*. 2018, págs. 783-788.

[10] Alain Muzet. «Environmental noise, sleep and health». En: *Sleep medicine reviews* 11.2 (2007), págs. 135-142.

[11] Staffan Hygge, Gary W Evans y Monika Bullinger. «A prospective study of some effects of aircraft noise on cognitive performance in schoolchildren». En: *Psychological science* 13.5 (2002), págs. 469-474.

[12] Marco Chetoni, Elena Ascari, Francesco Bianco, Luca Fredianelli, Gaetano Licitra y Liliana Cori. «Global noise score indicator for classroom evaluation of acoustic performances in LIFE GIOCONDA project». En: *Noise Mapping* 3.1 (2016).

[13] Julia Dratva, Harish C Phuleria, Maria Foraster, Jean-Michel Gaspoz, Dirk Keidel, Nino Künzli, L-J Sally Liu, Marco Pons, Elisabeth Zemp, Margaret W Gerbase y col. «Transportation noise and blood pressure in a population-based sample of adults». En: *Environmental health perspectives* 120.1 (2012), pág. 50.

[14] HM Miedema y CG Oudshoorn. «Annoyance from transportation noise: relationships with exposure metrics DNL and DENL and their confidence intervals.» En: *Environmental health perspectives* 109.4 (2001), pág. 409.

[15] Fabrizio Minichilli, Francesca Gorini, Elena Ascari, Fabrizio Bianchi, Alessio Coi, Luca Fredianelli, Gaetano Licitra, Federica Manzoli, Lorena Mezzasalma y Liliana Cori. «Annoyance judgment and measurements of environmental noise: A focus on Italian secondary schools». En: *International journal of environmental research and public health* 15.2 (2018), pág. 208.

[16] Patrice Guyot, Julien Pinquier, Xavier Valero y Francesc Alias. «Two-step detection of water sound events for the diagnostic and monitoring of dementia». En: *Multimedia and Expo (ICME), 2013 IEEE International Conference on*. IEEE. 2013, págs. 1-6.

[17] Toshiyo Tamura, Atsushi Kawarada, Masayuki Nambu, Akira Tsukada, Kazuo Sasaki y Ken-Ichi Yamakoshi. «E-healthcare at an experimental welfare techno house in Japan». En: *The open medical informatics journal* 1 (2007), pág. 1.

[18] Tatsuya Yamazaki. «The ubiquitous home». En: *International Journal of Smart Home* 1.1 (2007), págs. 17-22.

[19] Marie Chan, Daniel Estève, Christophe Escriba y Eric Campo. «A review of smart homes—Present state and future challenges». En: *Computer methods and programs in biomedicine* 91.1 (2008), págs. 55-81.

[20] Adriana M Adami, Misha Pavel, Tamara L Hayes y Clifford M Singer. «Detection of movement in bed using unobtrusive load cell sensors». En: *IEEE Transactions on Information Technology in Biomedicine* 14.2 (2010), págs. 481-490.

[21] Rosa Ma Alsina-Pagès, Joan Navarro, Francesc Alías y Marcos Hervás. «homeSound: Real-Time Audio Event Detection Based on High Performance Computing for Behaviour and Surveillance Remote Monitoring». En: *Sensors* 17.4 (2017), pág. 854.

[22] Susanna Spinsante, Ennio Gambi, Laura Montanini y Laura Raffaeli. «Data management in ambient assisted living platforms approaching IoT: a case study». En: *Globecom Workshops (GC Wkshps), 2015 IEEE*. IEEE. 2015, págs. 1-7.

[23] Jaime Lloret, Alejandro Canovas, Sandra Sendra y Lorena Parra. «A smart communication architecture for ambient assisted living». En: *IEEE Commun. Mag* 53.1 (2015), págs. 26-33.

[24] Fatih Erden, Senem Velipasalar, Ali Ziya Alkar y A Enis Cetin. «Sensors in Assisted Living: A survey of signal and image processing methods». En: *IEEE Signal Processing Magazine* 33.2 (2016), págs. 36-44.

[25] EU Comission. *Active and Assisted Living Programme. ICT for Ageing Well*. http://www.aal-europe.eu/ (accessed on 21 Feb 2017).

[26] Gregory D Abowd y Elizabeth D Mynatt. «Designing for the human experience in smart environments». En: *Smart environments: technologies, protocols, and applications* (2005), págs. 151-174.

[27] Emmanuel Munguia Tapia, Stephen S Intille y Kent Larson. «Activity recognition in the home using simple and ubiquitous sensors». En: *Pervasive*. Vol. 4. Springer. 2004, págs. 158-175.

[28] NM Barnes, NH Edwards, DAD Rose y P Garner. «Lifestyle monitoring-technology for supported independence». En: *Computing & Control Engineering Journal* 9.4 (1998), págs. 169-174.

[29] M Cobos, JJ Perez-Solano y LT Berger. «Acoustic-based technologies for ambient assisted living». En: *Introduction to Smart eHealth and eCare Technologies; Taylor & Francis Group: Boca Raton, FL, USA* (2016), págs. 159-180.

[30] Miguel A Quintana-Suárez, David Sánchez-Rodríguez, Itziar Alonso-González y Jesús B Alonso-Hernández. «A Low Cost Wireless Acoustic Sensor for Ambient Assisted Living Systems». En: *Applied Sciences* 7.9 (2017), pág. 877.

[31] Daniel Ellis. «Detecting alarm sounds». En: *Proc. Workshop on Consistent and Reliable Acoustic Cues CRAC-2000*. 2001.

[32] Anastasios Vafeiadis, Konstantinos Votis, Dimitrios Giakoumis, Dimitrios Tzovaras, Liming Chen y Raouf Hamzaoui. «Audio-based Event Recognition System for Smart Homes». En: *IEEE Ubiquitous Intelligence Computing Conference*. IEEE. 2017.

[33] Qin Zhao, Feng Guo, Xingshui Zu, Yuchao Chang, Baoqing Li y Xiaobing Yuan. «An Acoustic Signal Enhancement Method Based on Independent Vector Analysis for Moving Target Classification in the Wild». En: *Sensors* 17.10 (2017), pág. 2224.

[34] Mihail Popescu y Abhishek Mahnot. «Acoustic fall detection using one-class classifiers». En: *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*. IEEE. 2009, págs. 3505-3508.

[35] Saıda Bouakaz, Michel Vacher, M-E Bobillier Chaumon, Frédéric Aman, Salima Bekkadja, François Portet, Erwan Guillou, Solange Rossato, Elodie Desserée, Pierre Traineau y col. «CIRDO: Smart companion for helping elderly to live at home for longer». En: *IRBM* 35.2 (2014), págs. 100-108.

[36] Florian Kraft, Robert Malkin, Thomas Schaaf y Alex Waibel. «Temporal ICA for Classification of Acoustic Events ia Kitchen Environment». En: *Ninth European Conference on Speech Communication and Technology*. 2005.

[37] Kun Wang, Yun Shao, Lei Shu, Guangjie Han y Chunsheng Zhu. «LDPA: A local data processing architecture in ambient assisted living communications». En: *IEEE Communications Magazine* 53.1 (2015), págs. 56-63.

[38] Amir Vahid Dastjerdi y Rajkumar Buyya. «Fog computing: Helping the Internet of Things realize its potential». En: *Computer* 49.8 (2016), págs. 112-116.

[39] Octavian Fratu, Catalina Pena, Razvan Craciunescu y Simona Halunga. «Fog computing system for monitoring Mild Dementia and COPD patients-Romanian case study». En: *Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS), 2015 12th International Conference on*. IEEE. 2015, págs. 123-128.

[40] Harishchandra Dubey, Jing Yang, Nick Constant, Amir Mohammad Amiri, Qing Yang y Kunal Makodiya. «Fog data: Enhancing telehealth big data through fog computing». En: *Proceedings of the ASE BigData & SocialInformatics 2015*. ACM. 2015, pág. 14.

[41] Flavio Bonomi, Rodolfo Milito, Jiang Zhu y Sateesh Addepalli. «Fog computing and its role in the internet of things». En: *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM. 2012, págs. 13-16.

[42] Dan Garlasu, Virginia Sandulescu, Ionela Halcu, Giorgian Neculoiu, Oana Grigoriu, Mariana Marinescu y Viorel Marinescu. «A big data implementation based on Grid computing». En: *Roedunet International Conference (RoEduNet), 2013 11th*. IEEE. 2013, págs. 1-4.

[43] Vladimir Stantchev, Ahmed Barnawi, Sarfaraz Ghulam, Johannes Schubert y Gerrit Tamm. «Smart items, fog and cloud computing as enablers of servitization in healthcare». En: *Sensors & Transducers* 185.2 (2015), pág. 121.

[44] Razvan Craciunescu, Albena Dimitrova Mihovska, Mihail Rumenov Mihaylov, Sofoklis Kyriazakos, Ramjee Prasad y Simona Halunga. «Implementation of Fog Computing for Reliable E-Health Applications». English. En: *2015 49th Asilomar Conference on Signals, Systems and Computers*. IEEE Press, nov. de 2015, págs. 459 -463. ISBN: 978-1-4673-8574-9. DOI: 10.1109/ACSSC.2015.7421170.

[45] Yu Cao, Songqing Chen, Peng Hou y Donald Brown. «FAST: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation». En: *Networking, Architecture and Storage (NAS), 2015 IEEE International Conference on*. IEEE. 2015, págs. 2-11.

[46] Yannis Nikoloudakis, Spyridon Panagiotakis, Evangelos Markakis, Evangelos Pallis, George Mastorakis, Constantinos X Mavromoustakis y Ciprian Dobre. «A fog-based emergency system for smart enhanced living environments». En: *IEEE Cloud Computing* 3.6 (2016), págs. 54-62.

[47] Rosa Ma Alsina-Pagès, Joan Navarro y Enric Casals. «Automated Audio Data Monitoring for a Social Robot in Ambient Assisted Living Environments». En: *New Friends. In Proceedings of the 2nd International Conference on Social Robots in Therapy and Education*. 2016.

[48] Emre Cakir, Toni Heittola, Heikki Huttunen y Tuomas Virtanen. «Polyphonic sound event detection using multi label deep neural networks». En: *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE. 2015, págs. 1-7.

[49] Michael Imhoff y Silvia Kuhls. «Alarm algorithms in critical care monitoring». En: *Anesthesia & Analgesia* 102.5 (2006), págs. 1525-1537.

[50] Code Project. *Spectrogram Generation in SampleTagger*. 2016. URL: https://www.codeproject.com/Articles/806042/Spectrogram-Generation-in-SampleTagger (visitado 02-08-2018).

[51] Pornntiwa Pawara, Emmanuel Okafor, Lambert Schomaker y Marco Wiering. «Data Augmentation for Plant Classification». En: *ACIVS*. 2017.

[52] Ester Vidaña-Vila, Joan Navarro y Rosa Ma Alsina-Pagès. «Towards Automatic Bird Detection: An Annotated and Segmented Acoustic Dataset of Seven Picidae Species». En: *Data* 2.2 (2017), pág. 18.

[53] Laurens van der Maaten y Geoffrey Hinton. «Visualizing data using t-SNE». En: *Journal of machine learning research* 9.Nov (2008), págs. 2579-2605.

[54] Matteo Alberti. *Incrustación estocástica de vecinos (SNE) y su corrección en t-SNE*. 2017. URL: http://www.deeplearningitalia.com/incrustacion-estocastica-de-vecinos-sne-y-su-correccion-en-t-sne/ (visitado 04-08-2018).

[55] L Van Der Maaten y G Hinton. «Visualizing high-dimensional data using t-sne. journal of machine learning research». En: *J Mach Learn Res* 9 (2008), pág. 26.

[56] Malcolm Slaney. *Auditory Toolbox*. [Online; visitado el 5-Agosto-2018]. 1998. URL: www.engineering.purdue.edu/~malcolm/interval/1998-010/AuditoryToolboxTechReport.pdf.

[57] Jinchang Ren. «ANN vs. SVM: Which one performs better in classification of MCCs in mammogram imaging». En: 26 (feb. de 2012).

[58] Saimadhu Polamuri. *Difference between Softmax function and Sigmoid function*. [Online; visitado el 7-Agosto-2018]. 2017. URL: http://dataaspirant.com/2017/03/07/difference-between-softmax-function-and-sigmoid-function/.

[59] James Leonard y MA Kramer. «Improvement of the backpropagation algorithm for training neural networks». En: *Computers & Chemical Engineering* 14.3 (1990), págs. 337-341.

[60] Diederik P Kingma y Jimmy Ba. «Adam: A method for stochastic optimization». En: *arXiv preprint arXiv:1412.6980* (2014).

[61] Scikit-learn developers. *sklearn.svm.SVC*. [Online; visitado el 2-Agosto-2018]. Ago. de 2017. URL: http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html.

[62] Jinchang Ren. «ANN vs. SVM: Which one performs better in classification of MCCs in mammogram imaging». En: *Knowledge-Based Systems* 26 (2012), págs. 144-153.

[63] Raul Parada, Joan Melia-Segui, Marc Morenza-Cinos, Anna Carreras y Rafael Pous. «Using RFID to detect interactions in ambient assisted living environments». En: *IEEE Intelligent Systems* 30.4 (2015), págs. 16-22.

[64] Juan Buhagiar. «Automatic Segmentation of Indoor and Outdoor scenes from Visual Lifelogging». Tesis doct. University of Malta, 2017.

[65] Huy Phan, Marco Maaß, Radoslaw Mazur y Alfred Mertins. «Random regression forests for acoustic event detection and classification». En: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.1 (2015), págs. 20-31.

[66] Axel Plinge, Rene Grzeszick y Gernot A Fink. «A bag-of-features approach to acoustic event detection». En: *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE. 2014, págs. 3704-3708.

[67] M. Stager, P. Lukowicz y G. Troster. «Implementation and evaluation of a low-power sound-based user activity recognition system». En: *Eighth International Symposium on Wearable Computers*. Vol. 1. Oct. de 2004, págs. 138-141. DOI: 10.1109/ISWC.2004.25.

[68] David Karpf-Cogan. «Distributed Wireless Sensor Networks (WSNs) Bottleneck Detection». Tesis doct. Hebrew University of Jerusalem, 2010.

[69] L Krishnamachari, Deborah Estrin y Stephen Wicker. «The impact of data aggregation in wireless sensor networks». En: *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*. IEEE. 2002, págs. 575-578.

[70] Mitilineos A Stelios, Argyreas D Nick, Makri T Effie, Kyriazanos M Dimitris y Stelios CA Thomopoulos. «An indoor localization platform for ambient assisted living using UWB». En: *Proceedings of the 6th international conference on advances in mobile computing and multimedia*. ACM. 2008, págs. 178-182.

[71] Celso P Figueiredo, Óscar S Gama, Carlos M Pereira, Paulo M Mendes, Sérgio Silva, Leonel Domingues y K-P Hoffmann. «Autonomy suitability of wireless modules for ambient assisted living applications: Wi-fi, zigbee, and proprietary devices». En: *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*. IEEE. 2010, págs. 169-172.

[72] Amir M Rahmani, Tuan Nguyen Gia, Behailu Negash, Arman Anzanpour, Iman Azimi, Mingzhe Jiang y Pasi Liljeberg. «Exploiting smart e-health gateways at the edge of healthcare internet-of-things: a fog computing approach». En: *Future Generation Computer Systems* 78 (2018), págs. 641-658.

[73] Bahar Farahani, Farshad Firouzi, Victor Chang, Mustafa Badaroglu, Nicholas Constant y Kunal Mankodiya. «Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare». En: *Future Generation Computer Systems* 78 (2018), págs. 659-676.

[74] Behailu Negash, Tuan Nguyen Gia, Arman Anzanpour, Iman Azimi, Mingzhe Jiang, Tomi Westerlund, Amir M Rahmani, Pasi Liljeberg y Hannu Tenhunen. «Leveraging fog computing for healthcare IoT». En: *Fog Computing in the Internet of Things*. Springer, 2018, págs. 145-169.

[75] Pawani Porambage, An Braeken, Andrei Gurtov, Mika Ylianttila y Susanna Spinsante. «Secure end-to-end communication for constrained devices in IoT-enabled Ambient Assisted Living systems». En: *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*. IEEE. 2015, págs. 711-714.

[76] NVIDIA. *JETSON TK1*. `http://www.nvidia.com/object/jetson-tk1-embedded-dev-kit.html` (accessed on 15 May 2016).

[77] David B Leake. «CBR in context: The present and future». En: *Case-Based Reasoning, Experiences, Lessons & Future Directions* (1996), págs. 1-30.

[78] Fabien Cardinaux, Deepayan Bhowmik, Charith Abhayaratne y Mark S Hawley. «Video based technology for ambient assisted living: A review of the literature». En: *Journal of Ambient Intelligence and Smart Environments* 3.3 (2011), págs. 253-269.

[79] Steven Weber, Jeffrey G Andrews y Nihar Jindal. «An overview of the transmission capacity of wireless networks». En: *IEEE Transactions on Communications* 58.12 (2010), págs. 3593-3604.

[80] Vsevolod P Nikolskiy, Vladimir V Stegailov y Vyacheslav S Vecher. «Efficiency of the Tegra K1 and X1 systems-on-chip for classical molecular dynamics». En: *High Performance Computing & Simulation (HPCS), 2016 International Conference on*. IEEE. 2016, págs. 682-689.

[81] Stefan Goetze, Niko Moritz, Jens-E Appell, Markus Meis, Christian Bartsch y Jörg Bitzer. «Acoustic user interfaces for ambient-assisted living technologies». En: *Informatics for Health and Social Care* 35.3-4 (2010), págs. 125-143.

[82] Naoya Takahashi, Michael Gygli, Beat Pfister y Luc Van Gool. «Deep convolutional neural networks and data augmentation for acoustic event detection». En: *arXiv preprint arXiv:1604.07160* (2016).

[83] Paul Mermelstein. «Distance measures for speech recognition, psychological and instrumental». En: *Pattern recognition and artificial intelligence* 116 (1976), págs. 374-388.

[84] Francesc Alías, Joan Claudi Socoró y Xavier Sevillano. «A review of physical and perceptual feature extraction techniques for speech, music and environmental sounds». En: *Applied Sciences* 6.5 (2016), pág. 143.

[85] Michele Amoretti, Folker Wientapper, Francesco Furfari, Stefano Lenzi y Stefano Chessa. «Sensor data fusion for activity monitoring in ambient assisted living environments». En: *International Conference on Sensor Systems and Software*. Springer. 2009, págs. 206-221.