



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DE TELECOMUNICACIÓN

Grado en Ingeniería de Sistemas de Telecomunicación

Trabajo Fin de Grado

**Posicionamiento en interiores a través de los sensores de
un terminal móvil utilizando algoritmos de clasificación**

Autor

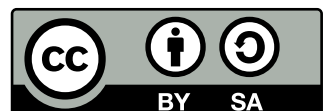
Jorge López-Cano Ávila

Tutor: Elena Aparicio Esteve

Co-tutor: David Casillas Pérez

Curso Académico 2023/2024

Esta obra está bajo una licencia Creative Commons
“Atribución-CompartirIgual 4.0 Internacional”.



Agradecimientos

Después del esfuerzo para terminar este trabajo quiero agradecer a mis tutores Elena Aparicio Esteve y David Casillas Pérez por su constante atención, dedicación y ayuda para la realización de este TFG.

También quiero agradecer a los compañeros que me apoyaron durante mi camino en el grado y a mis amigos más cercanos que me dieron la fuerza y apoyo necesario para que llegara hasta el final de este grado.

Finalmente, agradezco a mis padres Santiago López-Cano López y María Soledad Ávila Alias y a mi hermano Alejandro López-Cano Ávila por todo su apoyo y amor incondicional aportado desde el principio, para poder realizar el grado y darme las fuerzas para conseguir realizar este TFG.

Abstract

This work aims to study the use of visible light and *WiFi* waves through a mobile device to obtain the indoor position of a person. To achieve this, different data are collected and compared for each position along the trajectory that a person can follow inside a building. The data used include the coordinates of each position within the building, the light intensity measured at each point along the trajectories, and the precise moments when the measurements were taken for both light intensity and *WiFi* signal data.

Several trajectories with different parameters were used to study each position. This study was carried out indoors, specifically on the first floor of a building. Measurements were taken using a mobile terminal carried by a person along each trajectory, thus capturing the values mentioned above. This mobile terminal was responsible for collecting the data.

The main objective of this work is to develop an algorithm, using the data provided from the light and *WiFi* values database, that can accurately indicate the location along the trajectory at the time a measurement is taken. Subsequently, a Machine Learning SVM algorithm is developed using polynomial kernels and RBF to predict the position of the person based on the previously collected data. After analyzing the data, this algorithm predicts the most probable position based on the input measurements. Finally, it is observed in the results obtained that the most optimal way to predict the position of the path is using the algorithm with the WiFi data and an SVM with RBF kernel.

Resumen

Este trabajo tiene como objetivo estudiar el uso de la luz visible y las ondas *WiFi* a través de un dispositivo móvil para obtener la posición en interiores de una persona. Para lograr esto, se recopilan y comparan diferentes datos para cada posición a lo largo de la trayectoria que una persona puede seguir dentro de un edificio. Los datos utilizados incluyen las coordenadas de cada posición dentro del edificio, la intensidad de la luz medida en cada punto a lo largo de las trayectorias y los momentos precisos en los que se tomaron las medidas tanto para la intensidad de la luz como para los datos de la señal *WiFi*.

Se han empleado varias trayectorias con diferentes parámetros para estudiar cada posición. Este estudio se llevó a cabo en interiores, específicamente en la planta baja de un edificio. Se tomaron medidas utilizando un terminal móvil llevado por una persona a lo largo de cada trayectoria, capturando así los valores mencionados anteriormente. Este terminal móvil fue el responsable de recopilar los datos.

El objetivo principal de este trabajo es desarrollar un algoritmo, utilizando los datos proporcionados de la base de datos sobre los valores de la luz y del *WiFi*, que pueda indicar con precisión la ubicación a lo largo de la trayectoria en el momento en que se toma una medición. Posteriormente, se desarrolla un algoritmo de *Machine Learning SVM* utilizando *kernels polynomial* y RBF para predecir la posición de la persona en función de los datos previamente recopilados. Después de analizar los datos, este algoritmo predice la posición más probable basada en las mediciones de entrada. Finalmente se observa en los resultados obtenidos que la manera más óptima de predecir la posición del recorrido es utilizando el algoritmo con los datos del *WiFi* y un SVM con *kernel polynomial*.

Índice general

Agradecimientos	II
Abstract	III
Resumen	V
Lista de Figuras	X
Lista de Tablas	XI
Abreviaciones	XIII
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	3
1.3. Estructura de la memoria	4
2. Estado del arte	6
2.1. Introducción	6
2.2. Trabajos relacionados	7
3. Marco Teórico	16
3.1. ¿Qué es el Aprendizaje Máquina?	16
3.2. Tipos de Aprendizaje Máquina	17
3.3. Tipos de problemas de Aprendizaje Supervisado	20
3.4. Algoritmos de Aprendizaje Máquina Supervisados	20
3.4.1. <i>Logistic Regression</i> (Regresión Logística)	20

3.4.2.	<i>Decision Trees</i> (Árboles de Decisión)	21
3.4.3.	<i>Random Forest</i> (Árboles Aleatorios)	22
3.4.4.	<i>Support Vector Machine</i>	24
4.	Metodología y Desarrollo	28
4.1.	Introducción	28
4.2.	Base de datos	29
4.2.1.	Recogida de Datos	29
4.2.2.	Análisis de la Base de datos	30
4.2.3.	Arquitectura del Programa	34
4.2.4.	Métricas y Algoritmo de ML	37
5.	Resultados y Discusión	41
5.1.	Introducción	41
5.2.	Resultados Clasificadores	41
5.2.1.	Resultados RSS luminancias SVM <i>polynomial</i>	42
5.2.2.	Resultados RSS <i>WiFi</i> SVM <i>polynomial</i>	45
5.2.3.	Resultados RSS <i>WiFi</i> SVM RBF	47
5.3.	Comparativa	49
6.	Conclusión y Líneas Futuras	52
6.1.	Conclusiones	52
6.2.	Conocimientos adquiridos por el trabajo	53
6.3.	Impacto	54
6.3.1.	Impacto social	54
6.3.2.	Impacto medioambiental	54
6.3.3.	Impacto económico	54
6.4.	Líneas Futuras	56
A.	Figuras adicionales	58
A.1.	Capítulo 4	58
	Bibliography	67

Lista de Figuras

1.1.	Esquema de la incidencia de una señal de un satélite en un edificio.	2
1.2.	Figura estudio VLP de 4 LEDs.	3
1.3.	Esquema del estudio tratado en este TFG.	4
2.1.	Figura artículo tratado con VLP de 4 LEDs.	8
2.2.	Figura de mejora de precisión de un artículo.	9
2.3.	Figura estudio VLP de 3 WLEDs.	10
2.4.	Figura estudio VLP de 1 LEDs.	11
2.5.	Figura estudio posicionamiento por WiFi PSOSVR.	14
3.1.	Figura relación IA, ML y <i>Deep Learning</i>	17
3.2.	Figura de las aplicaciones del ML.	18
3.3.	Figura de los diferentes tipos del ML.	19
3.4.	Figura del modelo básico del Aprendizaje Reforzado.	19
3.5.	Figura de la función <i>logit</i>	21
3.6.	Figura <i>decision trees</i> como ejemplo del Titanic.	22
3.7.	Figura de <i>Random Forest</i>	23
3.8.	Figura de hiperplano de ejemplo SVM.	25
3.9.	Figura de las aplicaciones del ML.	26
4.1.	Esquema descriptivo pasos TFG.	28
4.2.	Figura descriptivo del entorno del TFG.	31
4.3.	Esquema de las localizaciones del TFG.	32
4.4.	Gráficas correspondientes a los valores del RSS de la luz.	33
4.5.	Gráfica de intensidad de luz por posiciones filtradas.	34

4.6.	Gráficas correspondientes a los valores del RSS del <i>WiFi</i> .	35
4.7.	Gráficas <i>box-plot</i> correspondientes a los valores del RSS de la luz.	36
4.8.	Gráficas <i>box-plot</i> correspondientes a los valores del RSS del <i>WiFi</i> .	37
4.9.	Gráficas correspondientes a los valores del RSS del <i>WiFi</i> .	38
4.10.	Gráficas correspondientes a los valores del RSS del <i>WiFi</i> .	39
5.1.	Matriz de confusión media de la luz SVM <i>polynomial</i> .	43
5.2.	Matriz de confusión media de la luz SVM RBF.	44
5.3.	Matriz de confusión media del <i>WiFi</i> SSID1 Freq 1 SVM <i>polynomial</i> .	45
5.4.	Matriz de confusión <i>WiFi</i> SSID1 frecuencia 2 SVM <i>polynomial</i> .	46
5.5.	Matriz de confusión <i>WiFi</i> SSID1 frecuencia 1 SVM RBF.	48
5.6.	Tabla de la media de métricas <i>WiFi</i> SVM RBF y frecuencia 2.	49
A.1.	Gráficas descriptivas de la trayectoria del recorrido.	58
A.2.	Intensidad de luz normalizada recibida por el receptor por cada recorrido.	59
A.3.	Gráfica intensidad de luz por posición sin media.	59
A.4.	Gráficas <i>box-plot</i> correspondientes a los valores del RSS del <i>WiFi</i> .	60
A.5.	Gráficas <i>box-plot</i> correspondientes a los valores del RSS del <i>WiFi</i> .	60
A.6.	Gráficas <i>box-plot</i> correspondientes a los valores del RSS del <i>WiFi</i> restando los valores de las medias.	61
A.7.	Gráficas <i>box-plot</i> correspondientes a los valores del RSS del <i>WiFi</i> restando los valores de las medias.	61
A.8.	Gráficas <i>box-plot</i> correspondientes a los valores del RSS del <i>WiFi</i> con frecuencia 2447 MHz y SSID2.	62
A.9.	Gráficas <i>box-plot</i> correspondientes a los valores del RSS del <i>WiFi</i> con frecuencia 2422 MHz y SSID3.	62
A.10.	Gráficas <i>box-plot</i> correspondientes a los valores del RSS del <i>WiFi</i> con frecuencia 2472 MHz y SSID4.	63
A.11.	Gráficas <i>box-plot</i> correspondientes a los valores del RSS del <i>WiFi</i> con frecuencia 2437 MHz y SSID5.	63

Lista de Tablas

2.1.	Tabla resumen de los artículos mencionados.	15
4.1.	Tabla de valores de la base de datos inicial.	30
5.1.	Tabla de la media de métricas luz SVM <i>polynomial</i>	43
5.2.	Tabla de la media de métricas luz SVM RBF.	44
5.3.	Tabla de la media de métricas <i>WiFi</i> SVM <i>polynomial</i> y frecuencia 1.	46
5.4.	Tabla de la media de métricas <i>WiFi</i> SVM <i>polynomial</i> y frecuencia 2.	47
5.5.	Tabla de la media de métricas <i>WiFi</i> SVM RBF y frecuencia 1.	47
5.6.	Tabla de la media de métricas <i>WiFi</i> SVM RBF y frecuencia 2.	48
5.7.	Tabla de la comparación de valores media de métricas.	50
5.8.	Tabla de la comparación de los mejores valores de las métricas.	50
6.1.	Tabla de las metas de los ODS de este TFG.	55

Abreviaciones

dBm decibelio milivatio. 29

DBSCAN *Density-Based Spatial Clustering of Applications with Noise*. 11, 12

ELM *Extreme Learning Machine*. 11, 12

FDM *Frequency Division Multiplexing*. 9

FFT *Fast Fourier Transform*. 9

GNSS *Global Navigation Satellite System*. 2, 7, 29, 30, 57

GPS *Global Positioning System*. 2, 6

IA Inteligencia Artificial. 16

IoT *Internet of Things*. 7

IPIN *International Conference on Indoor Positioning and Indoor Navigation*. 29

IPS *Indoor Positioning System*. 6

KNN *K-nearest neighbors*. 13, 17

LED *Light Emitting Diode*. 8–12

LEDs *Light Emitting Diodes*. 7, 8, 10

ML *Machine Learning*. 3, 5, 7, 10, 11, 16, 17, 27, 28, 33, 36, 41, 52

ODS *Objetivos de Desarrollo Sostenible*. 55

PD *PhotoDetector*. 7–11

PSO *Particle Swarm Optimization, Optimización de Enjambre de Partículas*. 13, 14

RBF *Radial Basis Function kernel.* 4, 5, 27, 37, 38, 41, 42, 44, 45, 47–50, 53

RGB *Red-Green-Blue.* 9, 10

RSS *Received Signal Strength.* 9, 10, 12, 14, 18, 29–34, 37, 41–44, 46, 50, 52, 58

SVM *Support Vector Machine.* 4, 5, 13, 14, 16, 17, 24–28, 35, 37, 41–50, 52, 56, 57

SVR *Support Vector Regression.* 13, 14

TFG *Trabajo de Fin de Grado.* 1, 3, 5, 28, 29, 34, 35, 42, 52–55

VLC *Visible Light Communication.* 9

VLP *Visible Light Positioning.* 1, 7, 9–11

WLED *White LED.* 9, 10

Capítulo 1

Introducción

En este capítulo se muestra una introducción al trabajo que se ha realizado y el motivo de ello. Comenzamos desarrollando la motivación que nos llevó a realizar este *Trabajo de Fin de Grado* (TFG), explicando algunas tecnologías y el motivo por el que se consideraron otras opciones, y se sigue por los objetivos que se han buscado cumplir en él. Finalmente se presenta una breve explicación de cómo se estructura este TFG.

1.1. Motivación

El posicionamiento en interiores es la tecnología y metodología utilizada para determinar la ubicación precisa de objetos o personas dentro de un edificio o espacio cerrado. Este proceso se basa en el uso de diversas tecnologías como señales de radiofrecuencia, *WiFi*, *Bluetooth*, *Visible Light Positioning* (VLP), sensores de movimiento, y tecnologías de mapeo y navegación, con el objetivo de proporcionar datos de localización en tiempo real, mejorar la gestión de recursos, y optimizar la experiencia del usuario en entornos interiores. En todo tipo de situaciones nos hemos encontrado con casos en los que no sabemos dónde se encuentra un lugar, persona u objeto en concreto, o bien por acumulación de personas, o bien porque se desconoce ese edificio o establecimiento en el que se encuentra [1]. Con el objetivo de poder ayudar a las personas, tanto como para encontrar lo que buscan, como evitar que se pierdan en aglomeraciones de personas, son algunos de los objetivos del posicionamiento en este caso.

El posicionamiento en interiores se ha convertido en uno de los objetivos para el estudio de la localización, debido a que no existían sistemas totalmente certeros para realizar un análisis y predicción correcta de una localización. Durante años se han realizado diversos estudios con respecto al posicionamiento, principalmente mediante la obtención de coordenadas en una posición. Estos estudios se realizaron con el objetivo de evitar el problema de no encontrar un lugar o persona en concreto mediante diferentes sistemas para

realizar esta búsqueda con certeza. Los sistemas más usados son el *Global Positioning System* (GPS)/*Global Navigation Satellite System* (GNSS), pero esos sistemas tienen sus limitaciones, las cuales perjudican su funcionamiento en entornos cerrados [2]. Esto es debido a que las señales de los satélites tienen dificultad de propagación, debido que a medida que van atravesando los medios y materiales la intensidad va disminuyendo y se va atenuando, de manera que puede llegar hasta a perderse la señal, como apreciamos en la Figura 1.1, donde se observa cómo la señal no llega hasta las personas/objetos que aparecen en el interior del edificio [3]. Además de esto, se encuentra que las ondas electromagnéticas pueden surtir efectos de reflexión y refracción que provoca la pérdida de la potencia de la señal, provocando que no sea una tecnología fiable en interiores. Esta es una limitación que provoca que sólo se pueda utilizar el sistema en puntos en los que se puedan recibir señales de los satélites.

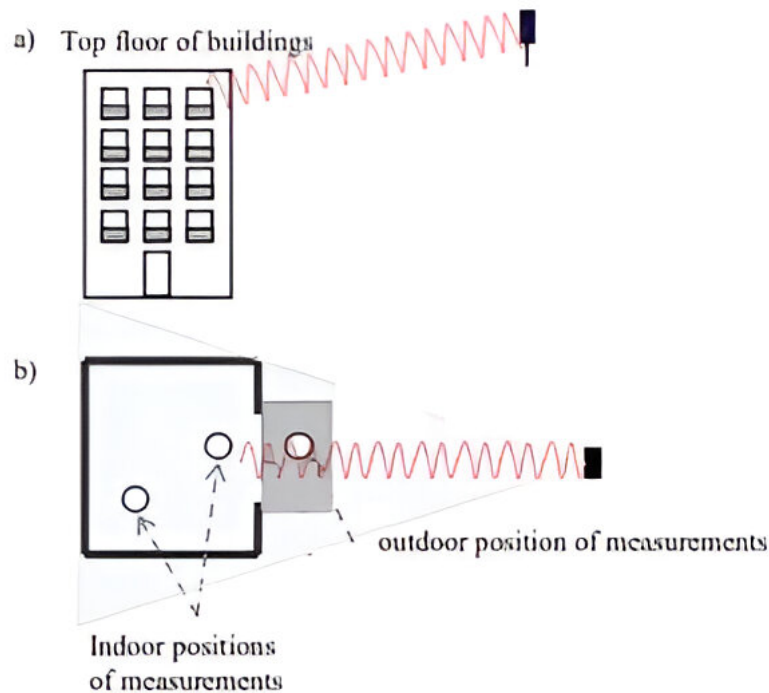


Figura 1.1: Esquema de la incidencia de una señal de un satélite en un edificio [3].

Debido a las características de los modelos existentes y los problemas causados en el resto de tecnologías como hemos mencionado anteriormente, se buscaron métodos de posicionamiento que utilizan otras tecnologías, dispositivos, etc. En este caso, hay estudios sobre posicionamiento en interiores a través de la luz emitida por una fuente exterior, y recibida por un receptor en el objeto a localizar, como se aprecia en la Figura 1.2, por lo que se empezaron a implementar estudios de sistemas que se basaban en luz visible [4]. También se han realizado estudios que intentaban predecir el posicionamiento en interiores en base al uso de la tecnología *WiFi* [5], puesto que es una tecnología que está muy

globalizada, y que su despliegue podría haber sido facilitado previamente por la instalación del edificio, si no sería necesaria la instalación *WiFi*.

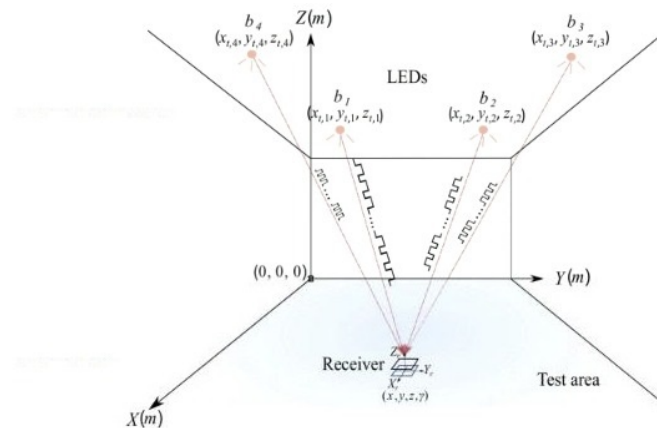


Figura 1.2: Esquema de un estudio de VLP con 4 LEDs en el techo cuya luz incide en un receptor en el suelo [4].

En este TFG se realiza un estudio de posicionamiento en interiores a través de un terminal móvil en un espacio, utilizando las diferentes señales de luz del entorno y el *WiFi* para obtener la localización de un usuario, y estima en qué posición se encuentra de las 7 seleccionadas. Mediante el uso de las señales de la luz y el *WiFi* y un clasificador, se busca estimar la posición de la persona en un entorno acotado, y evaluar su eficiencia en el caso de aplicarla. Este estudio se podría realizar en entornos de acumulación de personas en el interior de un edificio como un supermercado o un centro comercial, para poder ayudar a las personas que necesiten orientación o encontrar localizaciones dentro. Ésta tecnología se podría utilizar para conocer la posición de los distintos usuarios que se encuentren en el espacio y que tengan un receptor móvil. El problema que se trata en este TFG es el mostrado en la Figura 1.3. En esta figura aparece en color amarillo las posibles señales lumínicas que recibe el receptor móvil, y en rojo una de las posibles señales *WiFi* que recibe el receptor móvil. También recibe señales de otros *routers* y focos de luz.

Si se quisiera aplicar esta tecnología, podría hacerse en diferentes entornos, como el laboral, a nivel de empresas y para eventos en los hay aglomeraciones de personas, como puede ser un concierto, o un evento en centros cívicos, e incluso para ayudar a diferentes personas en hospitales.

1.2. Objetivos

El objetivo principal de este TFG es diseñar un algoritmo de *Machine Learning* (ML) que obtenga una estimación de un posicionamiento de un objetivo con un terminal móvil a partir de los valores de la luminancia y de la señal *WiFi*. Los objetivos secundarios de este TFG son:

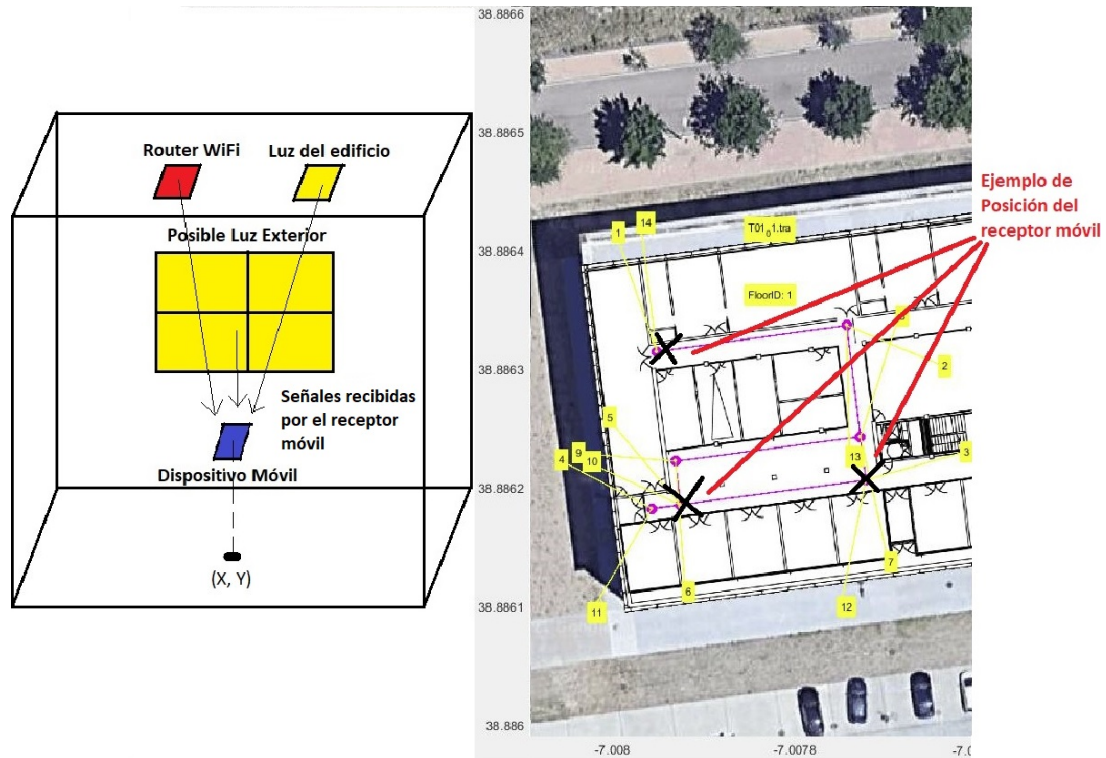


Figura 1.3: Esquema del estudio tratado en este TFG.

- Análisis y sustracción de los valores de la base de datos, para la realización del estudio y filtrado de los mismos.
- Desarrollo del algoritmo de clasificación *Support Vector Machine* (SVM), utilizando *kernels polynomial* y *Radial Basis Function kernel* (RBF) para estudiar cómo predice la posición de un usuario, tanto para los valores de la luminancia como para el *WiFi*, contemplando para este último dos frecuencias distintas y varios identificadores.
- Estudiar cómo influye la implementación de los dos tipos de *kernel* cuando se aplica cada tipo de datos.
- Comparativa de los resultados obtenidos, tanto utilizando la señal de luz como la señal *WiFi* y valoración del resultado entre ellos.

1.3. Estructura de la memoria

La memoria de este trabajo se ha dividido en 6 capítulos. En ellos, del Capítulo 1 al 3 se le dan contexto y se realiza una explicación de los puntos teóricos de la implementación y todo lo relacionado con el trabajo y el tipo de estudio. Los Capítulos del 4 al 6 se basan en la explicación del estudio realizado, la toma de datos de la base de datos, su

procesamiento y obtención de los diferentes resultados. Al final de la memoria se encuentra el Anexo con diferentes figuras explicativas para tener un contexto y una explicación un poco más detallada de los resultados y de los datos obtenidos y analizados durante el proceso. También se encuentra una Bibliografía donde se aprecian todas las citas utilizadas para analizar, encontrar y realizar la explicación de los diferentes puntos del estudio. Acto seguido, se explican brevemente los capítulos del TFG:

- Capítulo 1: Introducción. Se indica una descripción sobre qué trata este TFG y los objetivos y estructura del trabajo.
- Capítulo 2: Estado del arte. Se analiza una investigación sobre los diferentes estudios que se han realizado durante los últimos años del tema tratado en el trabajo, indicando y explicando diferentes artículos, experimentos y trabajos seleccionados para ello.
- Capítulo 3: Marco Teórico. Se exponen los diferentes conceptos teóricos de las tecnologías y métodos matemáticos realizados en el trabajo, incluyendo los métodos implementados en este TFG. En particular se describen los métodos de ML como el SVM con sus diferentes tipos de *kernel* que tiene, principalmente los utilizados en este trabajo, que son el *polynomial* y el RBF.
- Capítulo 4: Metodología y Desarrollo. Se presenta la base de datos utilizada y se analiza por medio de diferentes tipos de gráficas para observarlos detalladamente. Se presentan los métodos de filtrado y análisis de datos y la puesta a punto para obtener los distintos resultados.
- Capítulo 5: Resultados y Discusión. En este capítulo se presentan los resultados obtenidos tras la implementación del algoritmo para el análisis y comparativa. También se proporciona una discusión de los resultados obtenidos.
- Capítulo 6: Conclusión y Líneas Futuras. Por último, se desarrollan las conclusiones del estudio del trabajo y se presenta cómo se podrían mejorar los resultados o cómo se podría realizar un nuevo estudio partiendo de la base que proporciona este TFG para posibles investigaciones futuras.

Capítulo 2

Estado del arte

2.1. Introducción

Con el paso de los años, la mayoría de las actividades diarias se realizan en interiores, centros comerciales, museos, teatros, fábricas, hospitales, estaciones de tren, aeropuertos, etc. En estos lugares, en numerosas ocasiones ha habido problemas relacionados con encontrar un destino concreto en el interior debido al gran número de personas o por la complejidad de la estructura del lugar. El personal de estos sitios también encuentra, en multitud de ocasiones, problemas para dirigir a sus clientes a diferentes destinos, sus asientos, el baño o ascensores. Si bien en exteriores no hay graves dificultades a la hora de realizar una localización precisa de una persona, gracias a que existe el GPS, esto no ocurre en interiores, donde hay varios factores que no permiten que estas tecnologías sean eficaces, debido a las interferencias causadas por paredes u obstáculos [2]. Anteriormente se intentaba realizar este trabajo de localización *indoor* aplicando sistemas existentes como el GPS, pero se ha visto que tiene un rango de error demasiado grande, especialmente en pasillos, escaleras, o habitaciones sin ventanas [6]. Para intentar sobrepasar las limitaciones del GPS en interiores, se intentó implementar una versión mejorada del GPS, llamada *assisted* GPS [7]. Esta tiene una exactitud de 5 a 50 m en la mayoría de interiores. Adicionalmente, algunas compañías aportaron herramientas de detección de localización *indoor* aprovechando señales GPS débiles o transmitiendo señales tipo GPS aprovechando esta misma tecnología [8].

La baja precisión del GPS en interiores hace necesario el uso de sistemas de posicionamiento en interiores (IPS) (*Indoor Positioning System* (IPS)) específicos para poder determinar con exactitud el posicionamiento de personas en interiores. Las tecnologías más utilizadas se pueden clasificar en ópticas, mecánicas, magnéticas, acústicas y radiofrecuencia (UWB, *WiFi*, *Bluetooth*, entre otras) [9].

Los sistemas de posicionamiento basados en luz visible típicamente implementados

utilizan *Light Emitting Diodes* (LEDs) como transmisores y fotodiodos como receptores de la luz. Una de las características que tiene el uso de la luz visible es el bajo coste y efectividad. Además, gracias al extendido uso del teléfono móvil, su uso como receptor es un tema de investigación en los últimos años.

También existen sistemas de posicionamiento en interiores a través del *WiFi*, los cuales se realizan mediante el estudio de las ondas *WiFi* emitidas por un emisor, y recibidas por terminales. El posicionamiento por *WiFi* se ha estudiado por ejemplo mediante *deep learning* [10], o utilizando un modelo empírico [11].

Existen tecnologías actualmente para obtener el posicionamiento en interior como por ejemplo *Beacon Technology* [12], que utiliza pequeños transmisores que transmiten señales *Bluetooth*, o tecnologías GNSS (Galileo es la tecnología usada en la Union Europea), que son tecnologías de navegación por satélite, pero también encontramos sistemas que utilizan la VLP o el *WiFi*, e incluso utilizando *Internet of Things* (IoT) [12]. También se ha estudiado la posibilidad de usar tecnología como *Bluetooth* para poder realizar con exactitud un posicionamiento *indoor* [9].

2.2. Trabajos relacionados

Este apartado comprende diferentes artículos del estado del arte actual de los sistemas de posicionamiento en interiores del posicionamiento *indoor*, a través de *Google scholar*. Para garantizar que los resultados obtenidos en *Google scholar* son precisos, se ha realizado un filtrado temporal, que comprende artículos de los últimos 3 años. Las palabras clave utilizadas son “*indoor positioning*”, “*visible light*”, “*machine learning*”, y la sentencia booleana utilizada es:

$$\begin{aligned} & ('Indoor\ positioning')\ AND\ ('visible\ light'\ OR\ 'WiFi'\ OR\ 'sensor'\ OR\ 'high\ precision') \\ & AND\ ('machine\ learning'\ OR\ 'means\ as\ accurate'\ OR\ 'deep\ learning') \end{aligned}$$

A continuación se hablará sobre diferentes artículos que tratan el estudio del posicionamiento en interiores utilizando diferentes métodos, por ejemplo usando sistemas VLP y algoritmos ML [13]; otro artículo que trata sobre un sistema *indoor* VLP propuesto para una *multi-cell network* [14]; y entrando más para estudio utilizando *WiFi*, en un artículo se estudia cómo obtener la posición *indoor* utilizando un sistema de posicionamiento basado en *WiFi* y *Deep Learning* [10].

En este artículo se intentó demostrar que con el algoritmo utilizado se podía conocer la posición de un objeto determinado utilizando un *PhotoDetector* (PD) de ángulo receptor móvil. El algoritmo se basaba en modulación espacial. El caso que se le daba era una

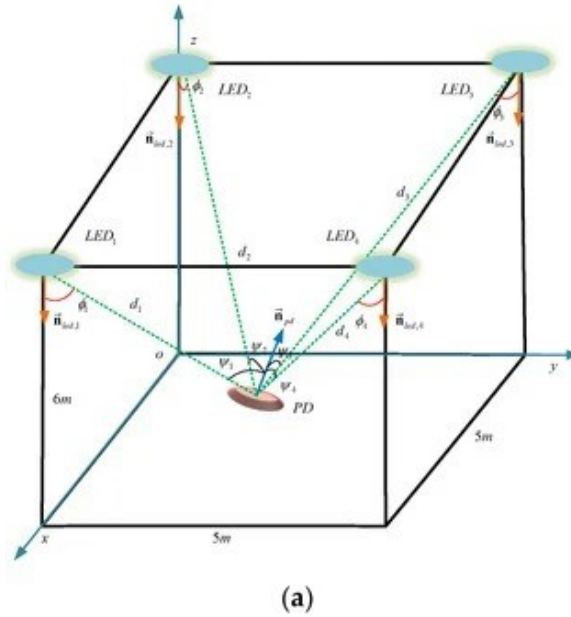


Figura 2.1: Plano de los 4 LEDs de la sala y el Fotodetector [15].

habitación que comprendía $3 \times 3 \times 4 \text{ m}^3$, como se aprecia en la Figura 2.1, colocando en el techo de ella 4 LEDs que eran los utilizados para el estudio del algoritmo [15].

En este estudio se utiliza un sensor de luz llamado Fotodetector 3D, con un algoritmo mejorado para optimizar los resultados y evitar los errores que aparecen debido a la poca movilidad en base al ángulo de rotación que tiene un PD. El algoritmo se llama *IWOA (Improved Whale Optimization Algorithm)* [15]. Uno de los problemas de la localización en interior 3D se puede solucionar con la implementación de este IWOA, que se caracteriza por una buena capacidad de búsqueda de optimización y pocos parámetros de control.

Se realizaron varias simulaciones para validar el rendimiento del método de posicionamiento implementado. El tamaño de los *Light Emitting Diode (LED)* colocados eran $5 \times 5 \times 6 \text{ cm}^3$. El experimento se realizó con y sin rotación del ángulo del PD colocado. Para comenzar las pruebas, se colocó el PD en el centro de la habitación, y luego se colocaron puntos de prueba. Después, se realizó un número de iteraciones para ver como se iban modificando las pruebas y se obtuvo una tasa de error de 2,34 cm.

Esta mejora de la tasa de error se debe a que se mejoró el algoritmo inicial, *WOA (Whale Optimization Algorithm)*, puesto que gracias al uso del IWOA sobre el WOA, los datos convergían entre si de manera más precisa para obtener la localización, como se observa en la Figura 2.2. Como se puede observar, a medida que incrementa el número de casos de prueba y test en el algoritmo, mayor es la precisión de éste, y menor tasa de error se encuentra.

Otro de los artículos que se pueden destacar [14], hablaba sobre un sistema de posicionamiento de luz visible que utilizaba redes multicelulares, en el cuál se usaban LEDs

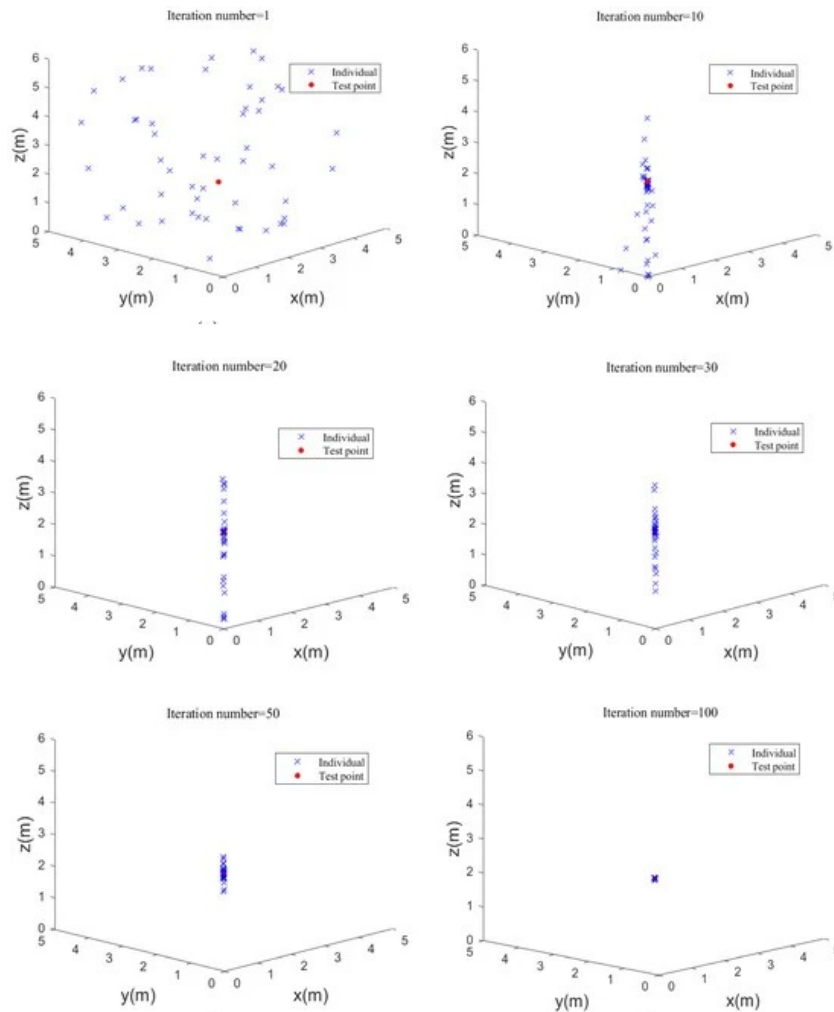


Figura 2.2: Seguimiento de gráficas que muestran el incremento de la precisión cuando aumenta el número de iteraciones con el algoritmo, a medida que aumentan las iteraciones, más preciso es [15].

Red-Green-Blue (RGB) y un esquema *Frequency Division Multiplexing* (FDM). Este sistema que se implementa utiliza funciones de un esquema FDM en los transmisores LED RGB y la *Received Signal Strength* (RSS) de los diodos para estimar la posición del receptor. Este artículo se implementó en dos pasos, lo primero fue desarrollar un nuevo sistema VLP con LED RGB para una red multicelular y transmitir la información VLP recibida por un único PD para estimar las señales FDM [14]. Segundo, se deriva una expresión para calcular la potencia de cada luz RGB incidente recibida por el PD. Para ello se utiliza la *Fast Fourier Transform* (FFT) para estimar las potencias y frecuencias recibidas. El uso de diferentes puntos de color en los LED mostraron una mejor precisión de posicionamiento.

En un principio, un sistema VLP basado en un PD de luz utiliza *Visible Light Communication* (VLC) para capturar la señal de luz. En este caso se utiliza *White LED* (WLED) y se usaba la potencia de la luz para obtener la información. Las ventajas de utilizar un

WLED es que tienen una mayor vida útil, menor tamaño y menor consumo. Este sistema utiliza 3 WLED en el techo en una red de múltiples celdas y un único PD móvil. La altura de cada WLED en el techo es la misma. El esquema del problema es el que representa la Figura 2.3.

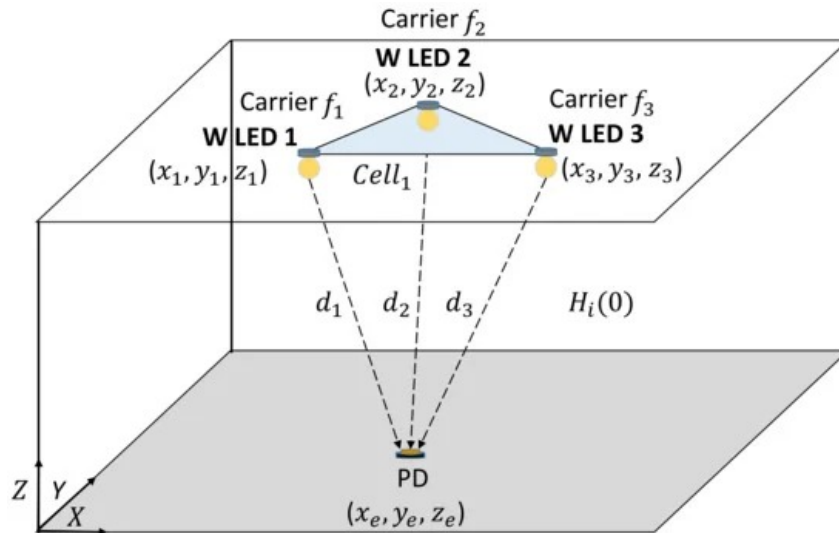


Figura 2.3: Esquema de la sala con 3 WLED y el receptor móvil [14].

Este sistema se estudió tanto para WLED como para LED RGB. Cada uno de los WLED transmiten una señal única para que los usuarios puedan calcular la posición del objeto a través del PD móvil. El uso del método de trilateración requiere la señal de al menos 3 WLED y así obtener la posición 2D del usuario en la habitación, gracias a la estimación de las distancias euclídeas entre el PD y los WLED utilizando su RSS. Se hace uso de un solo PD para convertir las señales ópticas recibidas en una corriente eléctrica, y ésta se utiliza para el problema de posicionamiento del PD gracias al RSS y la frecuencia. Para conocer la localización del objeto receptor se usa un algoritmo de identificación de frecuencia simple para conocer la posición del WLED que envió cada señal, y después se utiliza el algoritmo de trilateración con las distancias euclídeas a los LED y se estima la ubicación del receptor.

Este sistema VLP mostró un buen rendimiento de errores comparado con otras arquitecturas que utilizan WLED al igual. En los resultados se observó que eran más precisos utilizando una arquitectura LED RGB dado que se podía analizar también por la temperatura de color, lo cuál no es posible si se utiliza WLED, porque al ser siempre blanco es la misma temperatura para cada uno de los LEDs. El problema que se encuentra es que implementar RGB es más caro y requiere más componentes de hardware.

Este último artículo que trata sobre el VLP [16] habla sobre un caso de posicionamiento *indoor* que comprende de un solo emisor de luz LED y un solo PD rotatorio de luz en una habitación $4 \times 4 \times 3 \text{ m}^3$. Para la trata de datos implementa un algoritmo de ML teniendo en cuenta la reflexión de la luz en el sistema.

El proceso de posicionamiento que se propone se implementa en dos etapas: Clasificación de áreas y Posicionamiento. En la de Clasificación de áreas mediante la implementación del algoritmo de ML *Random Forest* la habitación en la que nos encontramos se divide entre un área interior y cuatro zonas de paredes o esquinas. El PD giratorio se utiliza para determinar la ubicación del objetivo. En la zona de las esquinas se implementa un algoritmo de posicionamiento híbrido comprendido entre *Extreme Learning Machine* (ELM) y *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) para mejorar la precisión de la localización cerca de las paredes y las esquinas interiores.

Consideramos el escenario como una habitación que contiene un solo LED instalado en el techo de esta, y un PD giratorio que detecta y demodula la señal óptica recibida como se aprecia en la Figura 2.4. La plataforma giratoria del PD incluye un cuerpo cilíndrico y un brazo giratorio para permitir que rote en sentido horario.

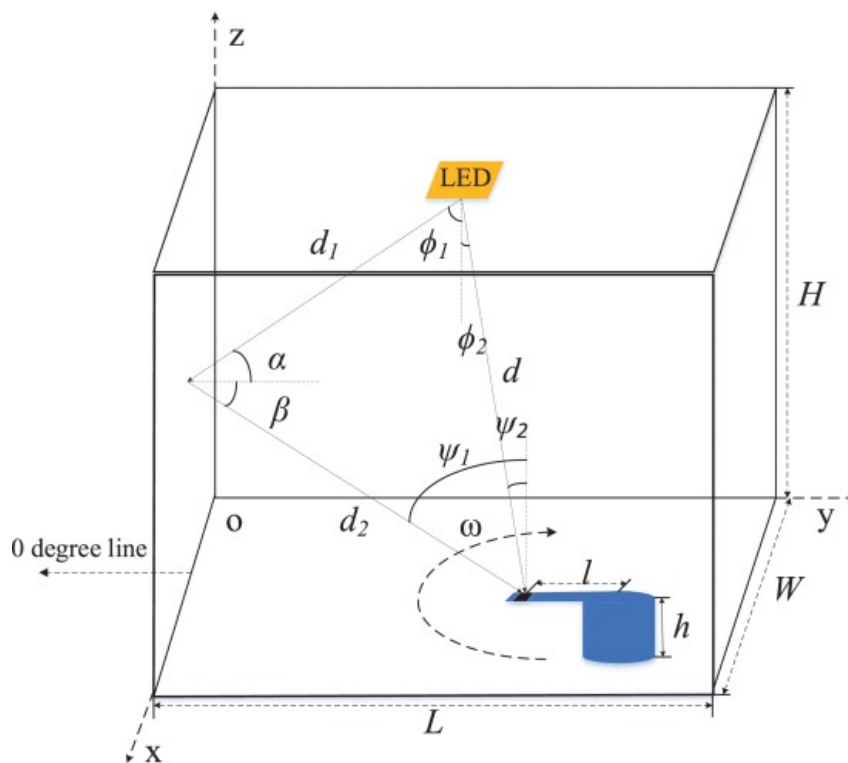


Figura 2.4: Esquema de este caso con un LED y un Fotodetector para localizar el objetivo [16].

La luz puede reflejarse en las paredes o instalaciones interiores, por lo que aparecen diferentes señales que recibe el PD a través de éstas, lo que provoca que disminuya la precisión de la posición. En los sistemas VLP el ruido normalmente se puede dividir en dos, térmico y de disparo. Con la luz incidente en el PD se obtiene el valor de la potencia eléctrica convertida a través de la potencia óptica y la potencia de ruido, provocada por el ruido mencionado, y con esta se estudia el posicionamiento.

Los algoritmos empleados en este artículo fueron:

- Algoritmo de clasificación de *Random Forest*, el cual al no requerir de demasiados parámetros se utiliza en diversos campos.
- Algoritmo ELM, que es un algoritmo de red neuronal basado en una única capa oculta que tiene la ventaja de tener una velocidad de aprendizaje rápida y buen rendimiento de generalización. Al configurarlo solo es necesario indicar el número de neuronas de la capa oculta y el tipo de función de activación para encontrar una solución óptima. Este algoritmo puede ajustar datos y clasificarlos.
- Algoritmo DBSCAN, que es un algoritmo de agrupamiento basado en densidad. Este algoritmo de agrupamiento suele suponer que la categoría puede determinarse por la rigidez de distribución de la muestra. Las muestras de la misma categoría se relacionan entre sí, es decir, todas las muestras de la misma categoría no se alejan mucho entre sí.

Los resultados de este sistema mostraron que funcionaba bien en habitaciones pequeñas, pero si se daba el caso de una habitación muy grande aparecen dos problemas: primero, que debido a la adquisición de datos de entrenamiento masivos necesarios para el aprendizaje automático, se encontraba una latencia muy grande, la cuál no favorece la implementación de un sistema a tiempo real, y la segunda que al ser una sala grande, un solo LED no daba una potencia suficiente para proporcionar una iluminación eficaz y posicionamiento preciso.

Los sistemas de posicionamiento basados en *WiFi* se intentaron aplicar desde el momento en el que la tecnología *WiFi* está tan globalizada. Se ha intentado aplicar el aprendizaje automático en el posicionamiento *indoor* basado en *fingerprinting* para mejorar la precisión de la localización *indoor*. En este artículo [17] se habla del uso del *fingerprinting* para proporcionar un método preciso para localizar en interiores utilizando el RSS.

Este estudio realizado presenta las técnicas utilizadas para la localización *indoor* con ventajas y desventajas y una revisión de las técnicas de aprendizaje automático y *WiFi fingerprinting* en interiores con sus ventajas y desventajas. Se han realizado diferentes estudios respecto a la localización *indoor* utilizando diferentes tecnologías ajenas al *WiFi*, entre las que se encuentran por ejemplo tecnologías *Bluetooth*, infrarrojos o ultrasonido.

En estas tecnologías implementadas es necesario la aplicación de *hardware* específico, pero para los sistemas basados solo en *WiFi* no es necesario recurrir a hardware adicional, lo que significa un coste más reducido y una ventaja sustancial frente al resto de tecnologías. Existen métodos de posicionamiento en interiores basados en *WiFi* utilizando el RSS, que son los basados en rango y los basados en *Fingerprinting*:

- Basados en rango, que se basan en la medición de distancias entre transceptores utilizando el RSS como método para convertir este en distancia y así calcular la

ubicación del cliente móvil. Estos métodos suelen ser algo más costosos debido a que requieren ciertas modificaciones de *hardware* o *software* en el punto de acceso y de la colocación de sensores de medición del ángulo de incidencia de la señal *WiFi*. También hay que tener en cuenta de que aparecen muchas interferencias debido a factores externos, como el efecto multitrayecto, la reflexión y refracción de la onda entre ellos.

- Basados en *Fingerprinting*, el cuál parte por recopilar señales y asociarlas con ubicaciones interiores, sin importar la distancia entre puntos del transmisor y receptor. En este método la característica de la ubicación se asocia con el modo de la señal detectada, por lo tanto, aunque se desconozca la ubicación exacta del punto de acceso *WiFi*, el reconocimiento de huellas dactilares puede proporcionar información del posicionamiento.

Los métodos de aprendizaje automático más utilizados son, el *K-nearest neighbors* (KNN), *random forest*, SVM, o algoritmos de agrupación, entre otros. La demanda de servicios de localización ha favorecido al desarrollo de la tecnología en interiores basadas en *Fingerprinting*. Este estudio se realizó para ayudar a proporcionar una referencia importante a los investigadores que desarrollaran métodos de localización *indoor* basados en *WiFi*. Una conclusión fue que se encontraron inconvenientes como los factores externos que provocan interferencias o el coste de aplicar el *hardware* necesario para la implementación de algunos sistemas. Entre estos problemas, el propio cuerpo de una persona humana es una interferencia de señales, y estos sistemas se quieren implementar principalmente en lugares donde frecuentan cantidades altas de personas.

En este otro artículo [5] se realizó el estudio de un sistema de localización *indoor* basado en *WiFi* mediante *Support Vector Regression* (SVR), optimizado por un algoritmo *Particle Swarm Optimization*, Optimización de Enjambre de Partículas (PSO) [5]. Este algoritmo se propuso para mejorar la precisión del sistema de localización. El SVR se dedica a resolver la localización como un problema de regresión, mediante un mapeo entre las características de la señal y las coordenadas espaciales en un espacio de altas dimensiones. El PSO se concentra en la estimación de parámetros óptimos del modelo SVR.

El sistema de posicionamiento PSOSVR *Positioning* implementado en este caso contiene dos etapas, etapa en línea y fuera de línea. En la Figura 2.5 vemos un esquema del funcionamiento del algoritmo mencionado. La dimensión de las coordenadas son las que definen el número de modelos que se implementará. Si tiene dos dimensiones, entonces se implementarán dos modelos PSOSVR.

Como podemos observar en la Figura 2.5, utiliza dos componentes clave en verde y en rosa, que son la construcción de modelos basados en SVR y de la estimación de parámetros

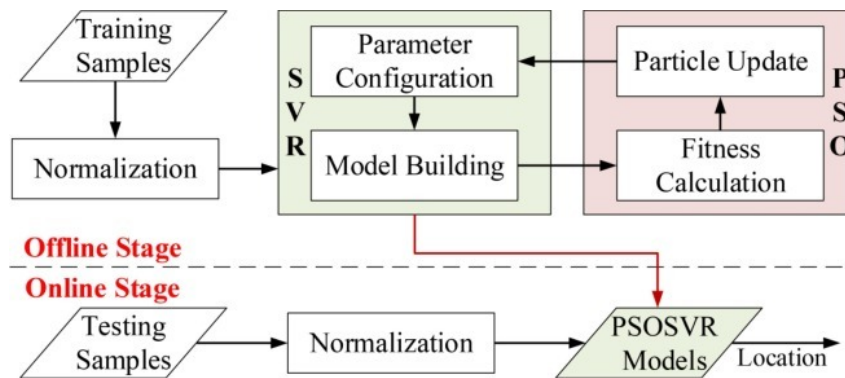


Figura 2.5: Esquema de un sistema de posicionamiento *indoor*. *WiFi* que utiliza el algoritmo PSOSVR *Positioning* y sus componentes [5].

mediante PSO. Este último entrega los parámetros óptimos globales, y el primero genera los modelos para su posicionamiento en la etapa en línea.

En el caso del posicionamiento interior *WiFi*, la regresión busca establecer una dependencia entre el vector RSS y la ubicación. Si la ubicación son dos dimensiones, se tienen que implementar dos modelos de regresión separados en función de las muestras de entrenamiento, y después se relaciona con el vector RSS para obtener la coordenada. El SVM es muy adecuado para conjuntos pequeños de muestras.

El algoritmo PSO es un método que optimiza un problema probando iterativamente un conjunto de soluciones llamadas partículas y desplazándolas hacia un espacio de soluciones en el espacio de búsqueda. En este caso, se coge el conjunto de muestras, se les optimizan los parámetros, como la posición, se le calcula el valor de aptitud de cada partícula en cada iteración para darle el mejor valor de aptitud posible, y se les actualiza el espacio de solución.

Este algoritmo de posicionamiento se aplicó para el caso de un entorno interior de longitud de 30 m y ancho de 4 m, separando en dos modelos de posicionamiento SVR basándose en dos parámetros esenciales estimados a partir del algoritmo PSO para evaluar la precisión del posicionamiento. Los resultados mostraron que el algoritmo podía lograr una precisión con tasa de fallo de 1 m, y que mejoraba los resultados de otros algoritmos de posicionamiento en el mismo caso hasta un 14 %, como era el caso de la implementación de un RMSE (*root-mean-square error*).

En la Tabla 2.1 se muestra un resumen de los artículos que se han mencionado y explicado, con sus tecnologías implementadas e información principal que aportan.

Tabla 2.1: Tabla resumen de los artículos mencionados sobre el VLP y la localización por *WiFi*, indicando el sistema de localización y el ML implementado.

	Sistema de localización	Sistemas utilizados	Información del Estudio	ML utilizado	Resultados
[15]	<i>Visible Light Positioning</i>	4 LEDs	Estudio realizado en un área de $3 \times 3 \times 4 \text{ m}^3$	IWOA	Errores de distancia de 2,34 cm
[14]	<i>Visible Light Positioning</i>	3 WLED y 3 RGB LED trilateración	Estudio realizado en un área de $1,8 \times 3,9 \times 1,47 \text{ m}^3$	Multiceldas y trilateración	Errores de distancia de 2,2 cm
[16]	<i>Visible Light Positioning</i>	LED y Fotodetector	Estudio realizado en un área de $4 \times 4 \times 3 \text{ m}^3$	Random Forest, ELM y DBSCAN	—
[17]	<i>WiFi Positioning</i>	<i>WiFi RSS Fingerprinting</i>	Estudio sobre el posicionamiento WiFi	—	—
[5]	<i>WiFi Positioning</i>	<i>WiFi RSS</i>	Estudio en un entorno variable	PSO y Regresión SVR	Errores de 1m en el 59,8% de las tomas

Capítulo 3

Marco Teórico

Este capítulo se estructura en cuatro secciones con respecto al ML. En la Sección 3.1 se proporciona una definición de la teoría del Aprendizaje Máquina. En la Sección 3.2 se explican los diferentes tipos de paradigmas de aprendizaje. En la Sección 3.3 se definen los tipos de problemas que existen de Aprendizaje Supervisado y su funcionamiento. Por último en la Sección 3.4 se explican algunos algoritmos de ML más utilizados, incluyendo el SVM que se implementa en el trabajo.

3.1. ¿Qué es el Aprendizaje Máquina?

La Inteligencia Artificial (IA) es el término que hace referencia a la creación de máquinas inteligentes intentando asemejarlas al cerebro humano, percibiendo su entorno y tomando acciones consecuentes a él para intentar garantizar sus objetivos. Como la mente humana, se busca que la máquina sea capaz de aprender [18]. El aprendizaje automático es el aprendizaje automático de las máquinas para clasificar e identificar datos, como imágenes o vídeos, entre otros. Utiliza algoritmos para analizar, aprender y tomar decisiones a partir de los datos sin procesar. Las técnicas convencionales de aprendizaje automático tienen una capacidad limitada para procesar los datos en su forma original. Los métodos que utilizan requieren una experiencia y comprensión en base a la selección de características que se aporten. El aprendizaje automático empezó en los años 50 a crear máquinas inteligentes con un programa de ajedrez a papel y lápiz, avanzando hasta desarrollar el inicio de las redes neuronales, y el desarrollo de algoritmos de ML [19]. A día de hoy, entre los algoritmos de aprendizaje automático más implementados destacan el SVM, el clasificador lineal o la regresión logística. El aprendizaje profundo es un enfoque avanzado de aprendizaje automático que se utiliza para hacer que las computadoras puedan analizar y comprender automáticamente la información de los datos sin procesar. El aprendizaje automático es un subcampo de la IA, y el *Deep Learning* es un subconjunto del aprendizaje

automático, como podemos apreciar en la Figura 3.1.

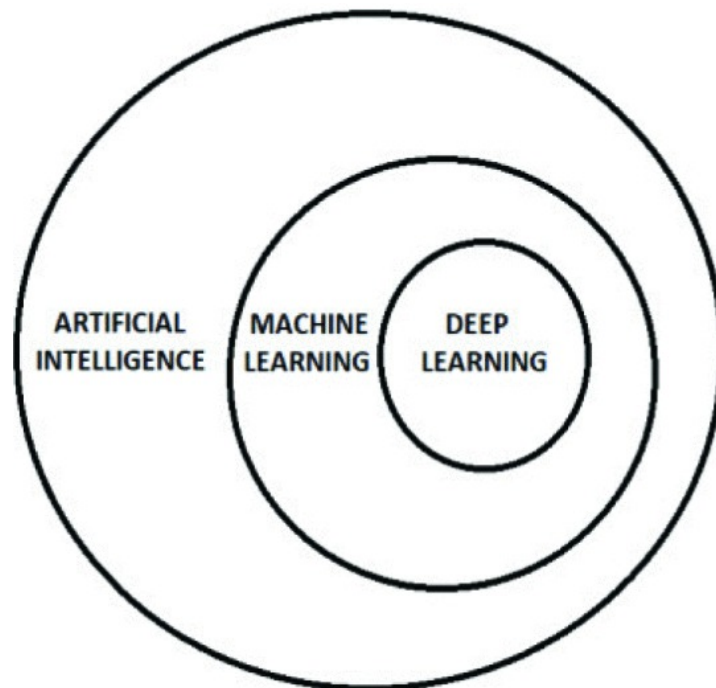


Figura 3.1: Esquema de la relación existente entre IA, ML y *Deep Learning* [19].

3.2. Tipos de Aprendizaje Máquina

Existen varios dominios y subdominios de aplicaciones de aprendizaje automático, como la visión por computadora, que trata de reconocimiento y detección de objetos; predicción, que se divide en clasificación, análisis y recomendación; y análisis semántico y procesamiento del lenguaje natural, que el primero es un proceso de relación de estructuras sintácticas de párrafos, y el segundo es cómo programar un equipo para que procese correctamente datos del lenguaje natural. En la Figura 3.2 podemos observar las aplicaciones del ML que han sido implementadas por investigadores, como reconocimiento de imágenes o diagnósticos médicos.

El aprendizaje automático se implementa dependiendo del requerimiento que se le de al sistema, por ejemplo, si busco reconocimiento de gestos manuales, se implementaría un algoritmo como el de vecinos más próximos KNN o el SVM. En esta Figura 3.3 podemos observar una clasificación de los tipos de Aprendizaje Máquina y qué algoritmos se utilizan en cada uno de ellos. Existen tres tipos de paradigmas de aprendizaje automático:

- Aprendizaje Supervisado: en este tipo se le enseña al algoritmo a inferir la relación entrada-salida con un conjunto de datos etiquetados bajo una característica para que todos sigan algo en común y sea capaz de realizar un análisis. Al conjunto de

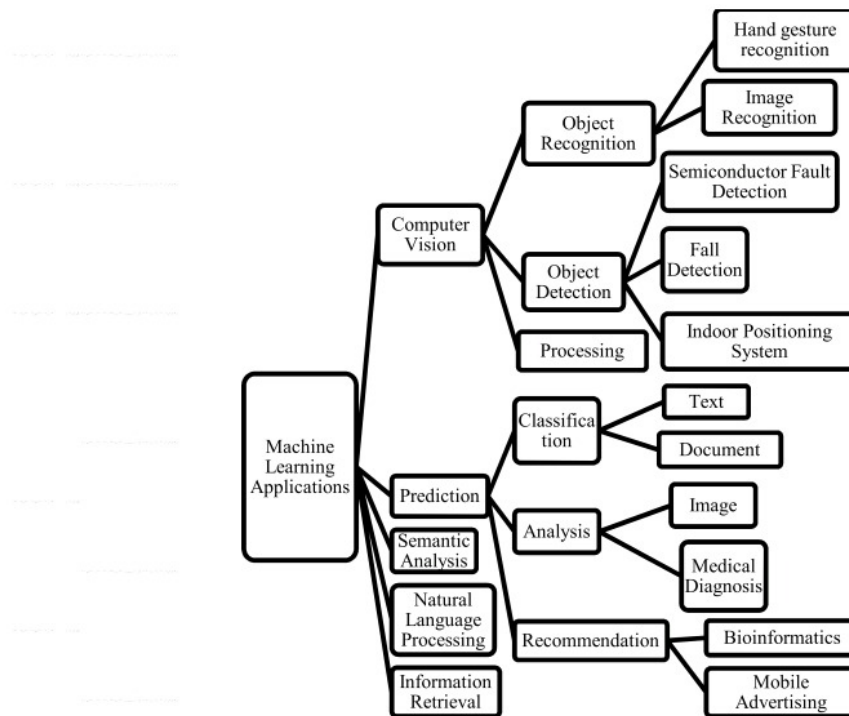


Figura 3.2: Esquema de las aplicaciones del ML [18].

datos etiquetados se le suele llamar base de datos. El método que se utiliza requiere de un conjunto de ejemplares que se han clasificado y almacenado en la base de datos, y este conjunto se llama conjunto de entrenamiento [21]. Con esta base de datos y los patrones que se encuentran, se determinan las categorías a las cuales pertenecen los nuevos ejemplares que se introducen en el modelo. En nuestro caso particular, como veremos en Sección 4.2, la base de datos estará formada por un conjunto de entradas (RSS de luminancias y RSS del *WiFi*) y un conjunto de salidas (posiciones o localizaciones de un lugar).

- **Aprendizaje No Supervisado:** este se define como un modelo predictivo que se entrena similar al supervisado, pero no se dispone de una base de datos etiquetada con las entradas y las salidas. Se tienen que inferir solas las relaciones a partir de los datos que se disponen y descubre él mismo patrones entre los grupos de datos. El principal problema surge cuando el sistema toma la decisión de elegir un determinado patrón entre todos los proporcionados, el propio sistema es el que debe elegir la clase de la cual se obtiene ese patrón. Otro problema es la toma de decisiones, sean correctas o no, para las cuales se utilizan técnicas como el *clustering* o agrupamiento. El *clustering* es el proceso de dividir un conjunto de datos en un conjunto de subclases llamadas *clusters*. Estos *clusters* son una colección de objetos de datos similares a otros para tratarlos como grupo. El agrupamiento es una forma de clasificación No Supervisada en la que no se conocen las etiquetas de las *class* y puede que tampoco se conozca su número de grupos [22].

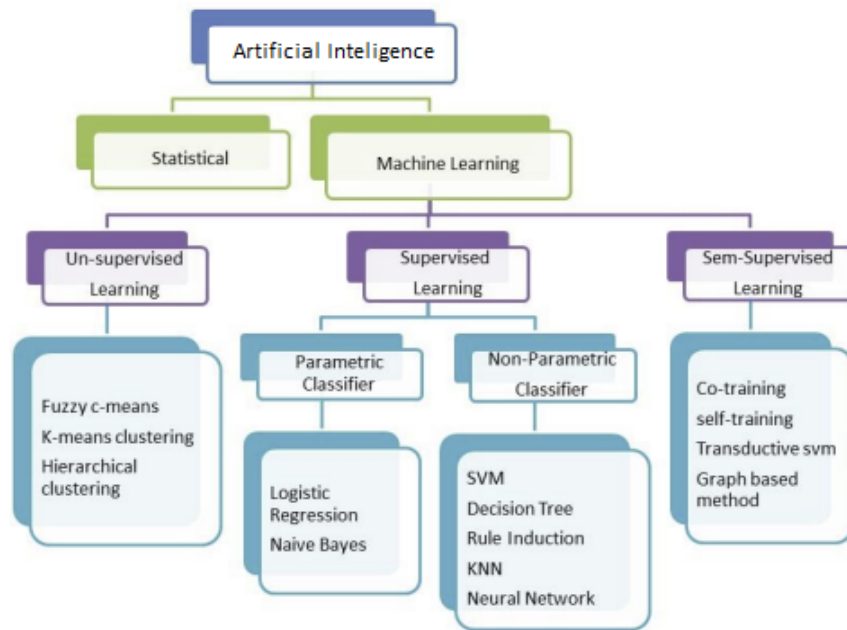


Figura 3.3: Esquema de los diferentes tipos de Aprendizaje Máquina y ejemplos de algoritmos [20].

- Aprendizaje Reforzado: en este tipo de aprendizaje automático el sistema aprende en un entorno que no tiene información sobre la posible salida de los datos, si no que obtiene resultados a través de acciones y busca resolver el problema de la mejor manera posible [23]. En el paradigma del Aprendizaje Reforzado, un agente se conecta a su ambiente, y percibe su estado actual, y selecciona una acción para cambiar este estado, como se aprecia en la Figura 3.4.



Figura 3.4: Modelo básico del Aprendizaje Reforzado [23].

3.3. Tipos de problemas de Aprendizaje Supervisado

El aprendizaje supervisado se utiliza en los problemas en los que nos encontramos con un conjunto de grupos de datos que hemos caracterizado y dividido, de manera que ellos se pueden identificar por una característica o un rasgo común entre ellos. Por ejemplo, si dividiésemos un conjunto de valores de potencia medidos por un fotodetector, lo podríamos agrupar en base a la potencia que nos está dando, escogiendo rangos de 10 W por ejemplo. Existen dos tipos de problemas de aprendizaje supervisado:

- **Clasificación:** Los problemas de tipo clasificación generalmente son aquellos que intentamos determinar los valores de una variable dependiente categórica, que pertenezca a un grupo, como que sea doctor, que una persona haya estudiado X carrera, etc., a partir de otras variables continuas predictoras. Por ejemplo, sería predecir quién va a aprobar un examen y quién no. Esto son problemas simples de clasificación binaria, donde la variable dependiente categórica solo puede asumir dos valores distintos y excluyentes entre sí. El objetivo trata de predecir las etiquetas de clase categóricas de nuevos registros con base en observaciones pasadas, es decir, predecir de qué clase serían los nuevos datos aportados. Si solo tiene dos clases se llama clasificación binaria, si tiene más se llama clasificación multiclase. En el caso que se propone en el Capítulo 4, es un problema de Clasificación en el cuál se busca determinar en cuál de las posiciones se encuentra un usuario, por lo que sería multiclase.
- **Regresión:** Los problemas de regresión son aquellos donde intentamos predecir los valores de una variable continua a partir de variables predictoras categóricas. Si queremos obtener, por ejemplo, la probabilidad de precios de una consola (esto sería el valor de la variable dependiente continua), a partir de otros predictores continuos como son, el tiempo que lleva en el mercado, o los juegos que estén sacando últimamente para esa consola (que serían variables de naturaleza categórica).

3.4. Algoritmos de Aprendizaje Máquina Supervisados

3.4.1. *Logistic Regression* (Regresión Logística)

En estadística, el modelo logístico es un modelo utilizado para predecir el resultado de una variable categórica (puede obtener un número limitado de categorías) en función de las variables independientes o predictoras. Es importante considerar la probabilidad de que un evento pueda ocurrir por diferentes factores [24]. El análisis de regresión logística

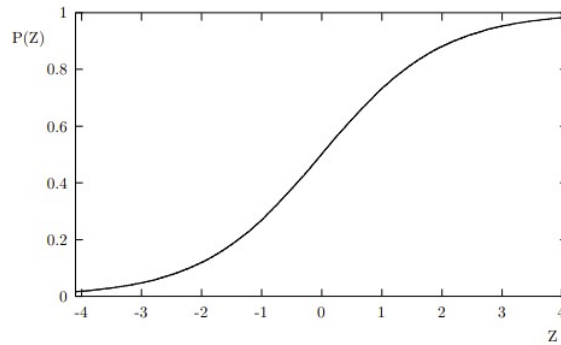


Figura 3.5: Esquema de la función *logit* de 0 a 1 usando base e [25].

se enmarca en el conjunto de modelos lineales generalizados en base a la función de enlace “*logit*” que se ve en la Figura 3.5.

Las probabilidades que describen el posible resultado de un único ensayo se modelan como una función de variables usando esta función *logit*. La regresión logística analiza los datos binomialmente, según la Ecuación (3.1):

$$Y_i = B(p_i, n_i), \forall i = 1, \dots, m, \quad (3.1)$$

en la cuál, n_i son los datos conocidos y p_i son las probabilidades de éxito, que son desconocidas. La i es el índice del número de datos que tenemos.

Después, los *logit* de las probabilidades binomiales se forman por la Ecuación (3.2):

$$\text{logit}(p_i) = \frac{\text{Ln}(p_i)}{(1 - p_i)} = B_0 + B_1 X_{1,i} + \dots + B_k X_{k,i} \quad (3.2)$$

En este caso la variable X puede variar para todo i , obteniéndose una variable independiente del modelo. En este caso, el valor que corresponda a la B será usualmente hallado por el método de máxima verosimilitud. Este último es un método para estimar los valores y aproximarlos entre si a los más repetidos.

3.4.2. *Decision Trees* (Árboles de Decisión)

Los árboles de decisión es un algoritmo de *Machine Learning* basado en la estructura de un árbol. Es uno de los métodos de aprendizaje supervisado más utilizados. Es un método eficaz para el mapeo de las relaciones no lineales. Los métodos en árbol potencian los modelos predictivos con alta precisión, estabilidad y facilidad de interpretación [26]. Este modelo sirve para poder resolver los problemas planteados de Regresión y Clasificación.

En *Machine Learning* la estructura de árbol es similar a un diagrama de flujo donde existe un nodo interno que representa un atributo, cada rama representa una regla de

decisión y cada hoja es el resultado. La raíz es el nodo superior del árbol. Se empieza a dividir en función del atributo principal que se estudia de manera recursiva. Esta estructura ayuda a tomar decisiones tomando forma como de comportamiento humano a la hora de elegir.

Los árboles de decisión clasifican por el árbol desde la raíz hasta algún nodo hoja, enfoque *top down*. Cada nodo del árbol es un caso de prueba para algún atributo y cada borde que desciende hasta ese nodo corresponde a una de las respuestas al caso de prueba. Este proceso se repite para cada subárbol volviendo a los nuevos nodos de manera recursiva. Este sería el esquema de *decision trees* en la Figura 3.6, tomando un ejemplo de datos de pasajeros del Titanic [27].

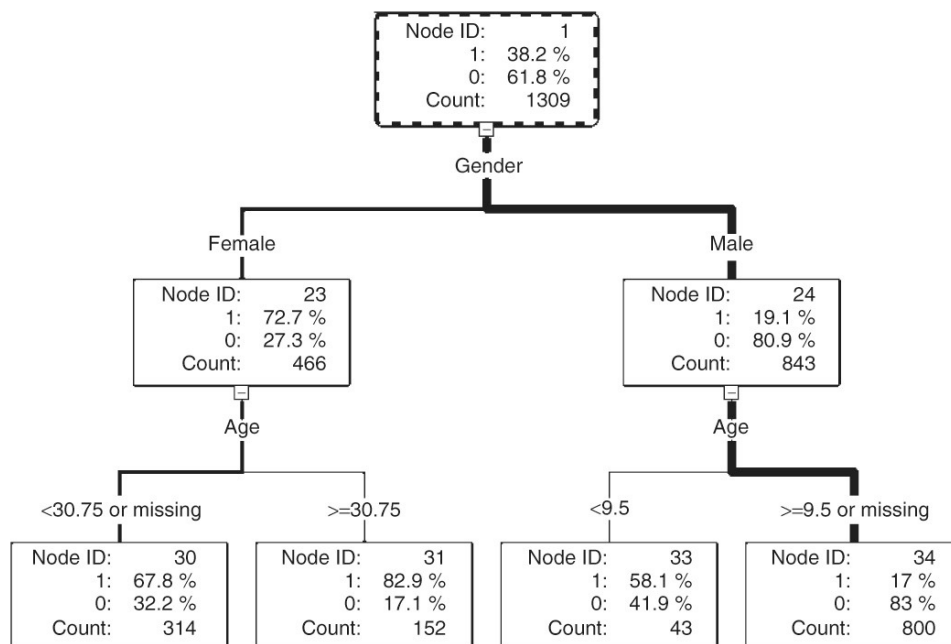


Figura 3.6: Esquema *decision trees* en base a datos de pasajeros del Titanic [27].

3.4.3. *Random Forest* (Árboles Aleatorios)

El algoritmo del bosque aleatorio es uno de los más comúnmente usados para tratar datos en problemas de Clasificación y Regresión. Básicamente construye diferentes árboles de decisión con distintos ejemplos y los valores mayoritarios para la clasificación y los valores aproximados para el caso de regresión. Este algoritmo es muy versátil, dado que aprovecha una cantidad múltiple de árboles de decisión para generar predicciones o clasificadores, combinando las salidas de estos árboles y con ellas consolidando un resultado más acertado. Una de las características más importantes de este algoritmo es que puede manejar la lista de datos de clasificación en variables continuas para los problemas de regresión o como variables categóricas en los problemas de clasificación [26].

Un ejemplo de bosque aleatorio es, por ejemplo, una persona que acaba de terminar selectividad, y no sabe qué carrera hacer, y le pregunta a varias personas de su familia, primos, hermanos, padres, etc. Entonces las preguntas que este les hace son diversas y variadas, como qué trabajo podría conseguir, qué becas tendría para cada uno, notas de corte, etc. Al final, tras consultar a estas personas, acabará por seleccionar aquella respuesta que más personas le hayan dado. En la Figura 3.7 se ve un ejemplo general de un esquema de funcionamiento del algoritmo *Random Forest*.

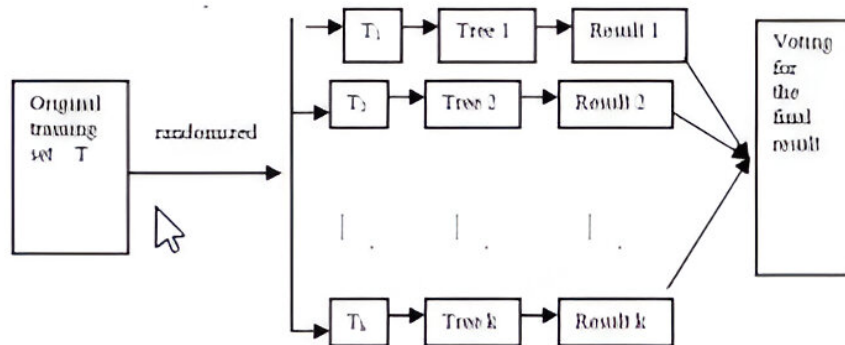


Figura 3.7: Esquema *Random Forest* [28].

Para trabajar con este algoritmo primero hay que entender el agrupamiento de datos. Simplemente esto consta de combinar diferentes modelos de datos, una colección que se usará para hacer las predicciones del modelo individual. Hay dos métodos de agrupamiento de datos [29]:

- *Bagging*: crear diferentes conjuntos de datos de entrenamiento desde los datos de entrenamiento para reemplazarlos, y el resultado final esta basado en el voto de la mayoría.
- *Boosting*: combina ejemplos simples de los datos recogidos con ejemplos más complejos o completos creando modelos secuenciales para que el modelo final sea el más preciso posible.

En este caso, *Random Forest* utiliza *Bagging*. Este método de agrupamiento de datos básicamente observa primero un conjunto aleatorio de nuestro almacén de datos. Entonces cada modelo es generado desde estos *sets* de datos (*Bootstrap Samples*), provistos desde el conjunto de datos original, y los reordena como filas de datos (*row sampling*). Este paso de alinear los datos y reemplazar se llama *Bootstrap*. Ahora cada modelo realiza el entrenamiento de manera independiente, lo que genera unos resultados, y la salida final se basará en la mayoría de los votos combinando los resultados de cada uno de los modelos. La combinación de los resultados generados por cada uno de estos modelos se denomina *aggregation*.

Los pasos del algoritmo son:

1. En el modelo de arboles aleatorios se coge un conjunto de datos y de características seleccionadas para construir cada árbol de decisión. Básicamente, un numero n *random* de ejemplos y numero m de características cogidas desde el set de datos, teniendo un numero total de ejemplos k .
2. Cada árbol de decisión se construye para cada muestra.
3. Cada árbol de decisión obtiene un resultado.
4. El resultado final se considera a través del promedio de la mayoría de resultados obtenidos para Clasificación y Regresión.

3.4.4. *Support Vector Machine*

SVM es una familia de métodos de aprendizaje automático diseñados para resolver el problema de clasificación y generalizarlo a otras situaciones [30]. Está basado en principios de la teoría del aprendizaje estadístico y optimización convexa. Este algoritmo utiliza funciones discriminantes de dos clases a partir de conjuntos de entrenamiento. Este tipo de algoritmos se han utilizado en un amplio área de análisis de datos de predicción en diferentes campos de estudio. Existen dos tipos distintos de SVM: Lineales y No lineales.

En el SVM lineal, el objetivo es buscar eficientemente un hiperplano de separación bueno en un espacio de características de alta dimensión. El rendimiento de un SVM se define en base a su margen, que es la distancia mínima entre el hiperplano que separa las dos clases y los puntos de datos más cercanos al hiperplano. El más óptimo será el que tiene el máximo margen de separación entre las dos clases. Este algoritmo se puede separar entre dos casos: el caso separable, con el clasificador de margen máximo y el caso no separable, con el clasificador de margen blando. El clasificador de margen máximo, o SVM de margen duro, no se utiliza con frecuencia en situaciones reales, puesto que solo funciona con datos separables linealmente. Este clasificador funciona de la siguiente manera, tengo una muestra de entrenamiento linealmente separable definida en la Ecuación (3.3), y un espacio de entrada. Las variables x e y representan los valores de los puntos de entrenamiento de la base de datos.

$$S = (x_1, y_1), \dots, (x_j, y_j) \quad (3.3)$$

Los SVM se basan en la clase de hiperplanos correspondientes a funciones de decisión definidas en la Ecuación(3.4). Los valores de w y x dan forma y dirección al plano, y b su desplazamiento. Los puntos del hiperplano siempre están en dirección perpendicular

al hiperplano, y si se varían los valores de los puntos, siempre se moverá el hiperplano de manera paralela a él, en la Figura 3.8 se observa un ejemplo de hiperplano de datos, donde γ es el valor de la distancia mínima entre los puntos de las dos clases de datos.

$$f(x) = \text{sign}((w, x) + b) \quad (3.4)$$

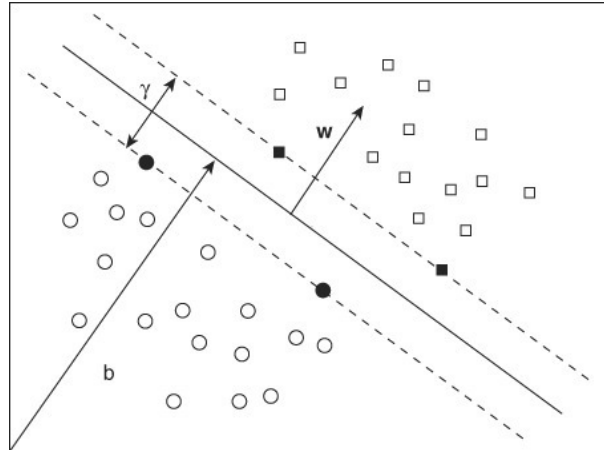


Figura 3.8: Ejemplo de un hiperplano de separación para datos bidimensionales en el caso lineal, donde las líneas discontinuas son los vectores de soporte [30].

Por lo tanto, el caso separable, siempre busca el hiperplano de separación con el margen mas grande, utilizando la Ecuación (3.5) como función de optimización. Las variables son las mismas que en la Ecuación (3.4):

$$y_i((x_i, w) + b) - 1 \geq 0 \quad (3.5)$$

Los puntos de datos más cercanos al hiperplano se les llama vectores de soporte, los cuales son los elementos críticos de la muestra de entrenamiento, porque contienen toda la información necesaria para reconstruir el hiperplano. Si los otros puntos de entrenamiento se eliminaran de la muestra de entrenamiento, o se movieran para no acercarse al hiperplano de separación, y se repitiera el entrenamiento, se encontraría el mismo hiperplano de separación máximo, lo que implica que los puntos que no son vectores de soporte se pueden eliminar sin pérdidas de información sobre la función de clasificación.

En el caso no separable se busca una versión más robusta del algoritmo SVM que puede tolerar ruido y valores atípicos en el modelo sin alterar la solución de manera drástica. Para conseguirlo se permite introducir clasificaciones erróneas en los datos de entrenamiento, reduciendo las restricciones. Esto se consigue introduciendo un valor con una variable positiva mayor o igual a 0 (la llamaremos E), obteniendo la Ecuación (3.6), cuyas variables son las mismas que en las Ecuaciones (3.4) y (3.5).

$$y_i((x_i, w) + b) - 1 \geq 1 - E \quad (3.6)$$

Por lo tanto, para que ocurra un error, la E debe de exceder a 1. Una manera de asignar un costo adicional por errores en el algoritmo sería que el propio usuario le diera valor manualmente, dependiendo de cuánto quieran penalizar los errores de clasificación al algoritmo. A mayor valor le den a la E , mayor será la penalización.

Los SVM no lineales son los más utilizados en el mundo real, más expresivo que las funciones lineales simples que hemos visto. Si tenemos en cuenta el caso anterior de datos bidimensionales, pero en este caso no se pueden separar linealmente, podríamos encontrar un caso como el que se describe en la Figura 3.9.

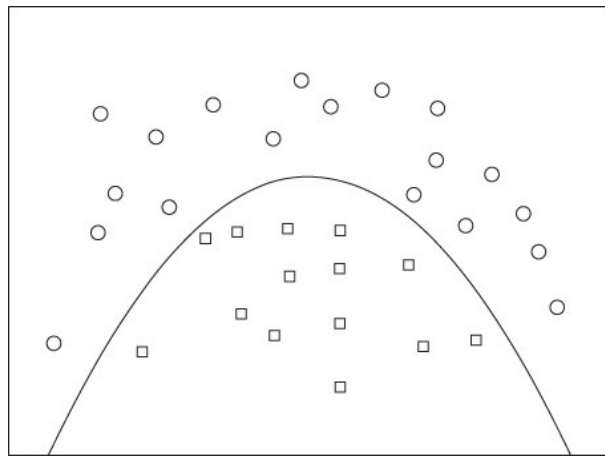


Figura 3.9: Ejemplo de un hiperplano de separación para datos bidimensionales en el caso no lineal [30].

Para obtener una mejor representación de los datos en este caso, se le asignan los puntos de datos a un espacio alternativo de dimensiones superiores llamado espacio de características. Al reemplazar el producto interno con una función central escogida de manera apropiada, los datos se pueden volver linealmente separables en el espacio de características, a pesar de no serlo en el espacio de entrada. Esto depende de la función objetivo que se va a aprender. Las relaciones lineales de los datos se pueden representar como productos internos entre todos los pares de puntos observados en los datos de entrenamiento sin usar sus coordenadas. La función que devuelve el producto interno entre dos puntos de datos cualesquiera se llama *kernel* (núcleo). La dimensión del espacio característico no se representa explícitamente, por lo que la cantidad de operaciones necesarias para calcular el producto interno en el *kernel* no es proporcional a la cantidad de características, si no a la de puntos de datos.

La única información utilizada en los ejemplos de entrenamiento es la matriz del *kernel*, definida como una matriz cuadrada compuesta por los conjuntos de vectores. Para crear las funciones del *kernel*, se realizan diferentes operaciones, como determinar las pro-

propiedades que son necesarias para garantizar que la función es un núcleo de algún espacio de características, verificar que existe un mapa de espacio de características para el que la función calcula el producto interno correspondiente, o operar con datos no vectoriales.

Para garantizar que el SVM conduzca a buenos resultados, se analiza su rendimiento de generalización. El objetivo de este es describir qué factores deben controlarse en el ML para garantizar un buen rendimiento, encontrando un límite en la probabilidad de error que se mantenga fiable. Se puede encontrar otro límite de error de generalización esperado utilizando una estimación de exclusión, que trata de eliminar un punto de entrenamiento, volver a entrenar, y después utilizar el punto eliminado anteriormente. A partir del resultado del vector de soporte sabemos que eliminar cualquier punto de entrenamiento que no sea vector de soporte no tiene efecto en el hiperplano encontrado. La elección del *kernel* es importante para el rendimiento de generalización de un SVM. Existen 4 tipos de *kernel*: RBF Ecuación (3.7), Lineal Ecuación (3.8), *polynomial* Ecuación (3.9) y Sigmoide Ecuación (3.10).

$$K(X_1, X_2) = \exp \frac{-\left(\|X_1 - X_2\|^2\right)}{(2\sigma^2)} \quad (3.7)$$

$$y_i = f(X, B) = B_0 + B_1X_1 + B_2X_2 + \dots + B_pX_p = 0 \quad (3.8)$$

$$K(x, y) = (x \cdot y + c) \cdot d \quad (3.9)$$

$$K(x, y) = \tanh(\alpha \cdot x \cdot y + B) \quad (3.10)$$

Capítulo 4

Metodología y Desarrollo

4.1. Introducción

En este capítulo se explica la metodología seguida para el desarrollo de este TFG. También se describe la información de la procedencia de los datos utilizados en los experimentos, cómo se han extraído y cómo se ha llevado a cabo el tratamiento de los datos. En la Figura 4.1 se puede ver detallado el proceso realizado.

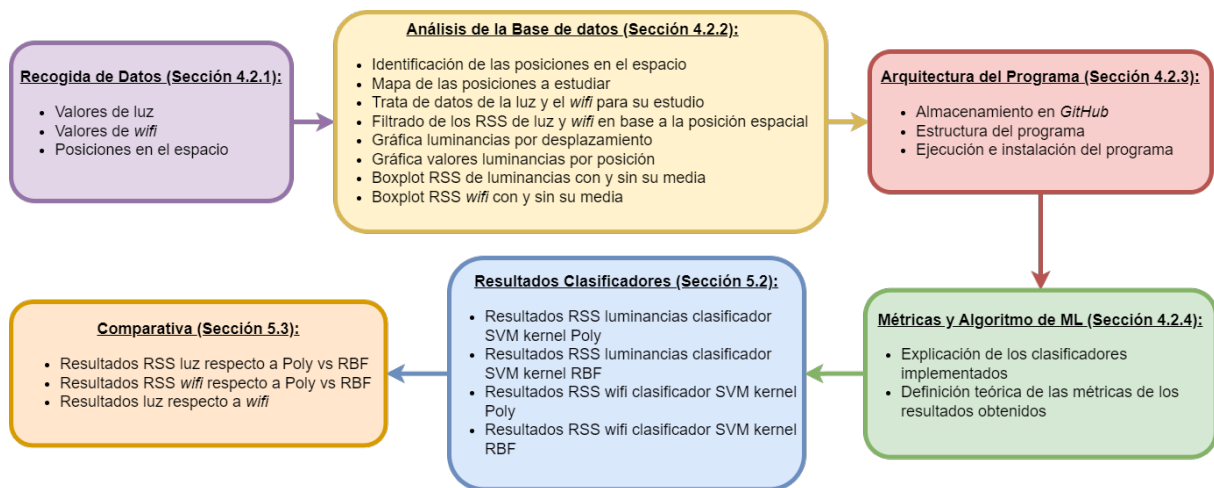


Figura 4.1: Esquema descriptivo de los pasos del TFG.

En el esquema se aprecian los pasos llevados a cabo para el análisis de los datos. Se comienza con la recogida de ellos desde la base de datos existente en los *bloc* de notas para introducirlos en la base de datos del proyecto. Después, se comienza con el análisis de los datos realizando operaciones con los valores obtenidos para poder realizar diferentes gráficas, como *box-plot* para interpretarlos. Por último, se realiza el estudio de estos datos utilizando como algoritmo de ML el SVM y observando la precisión del algoritmo implementado. En la Sección 4.2.2 de Análisis de la Base de Datos, veremos las

diferentes gráficas obtenidas a lo largo del proceso y cómo se reflejan estos datos en base al RSS de las luminancias por posiciones y del RSS del *WiFi*.

4.2. Base de datos

En esta sección, vamos a analizar los datos que tenemos de muestra, cómo hemos realizado el proceso de depuración de los mismos y por qué esta formada. Esta toma de datos se ha tomado a través de los recogidos en *International Conference on Indoor Positioning and Indoor Navigation* (IPIN), por el grupo de investigación *LOPSI research group* en España [31].

4.2.1. Recogida de Datos

Los datos fueron capturados a través de un móvil *Samsung* modelo *SM – N960F* que llevaba una persona a través del recorrido realizado en el interior del edificio, como se aprecia en la Figura 4.2. Se ha realizado el mismo recorrido de captura de datos diez veces, recogiendo los datos en un fichero distinto en cada ocasión. Los valores recogidos por el receptor móvil son valores correspondientes al acelerómetro, datos del giróscopo, magnetómetro, datos de la presión, luminancia, humedad, temperatura ambiente, orientación, GNSS, *WiFi*, *bluetooth*, datos del sonido y de la posición, como vemos en la Tabla 4.1. En este TFG nos centramos en los valores de posición y de los RSS del *WiFi* y las luminancias.

El recorrido se encuentra en un entorno iluminado, el pasillo de una universidad que tiene ventanas e iluminación por los focos del edificio. También se encuentran diferentes redes *WiFi* en el recorrido, de las que se reciben sus diferentes señales con varias frecuencias e identificadores para distinguir cada una de ellas.

Cada uno de los valores de la luz que se recogieron contiene la medida en segundos desde que empieza el recorrido y el valor de la intensidad de la luz (lux) recibida por el receptor lumínico del móvil en cada momento. Para la parte del *WiFi*, los datos obtenidos contenían valores correspondientes al tiempo en el que se tomó la medida desde que comenzó el recorrido, la intensidad del RSS medido en decibelio milivatio (dBm), el valor de la frecuencia del *WiFi* en MHz, y el SSID, que es el identificador de cada *WiFi* que se encuentra en la medida. El recorrido se mide en coordenadas cartesianas como se ve en la Figura 4.3, donde se muestran las posiciones clave destacadas en el trayecto realizado en la toma de datos.

Tabla 4.1: Tabla de los valores que se encuentran en los ficheros txt de la base de datos inicial, mostrando en negrita los utilizados en este TFG.

Acelerómetro	Tiempo (s)	Tiempo del sensor	Accel X	Accel Y	Accel Z	Precisión
Giroscopio	Tiempo (s)	Tiempo del sensor (s)	Gyr X	Gyr Y	Gyr Z	Precisión
Magnetómetro	Tiempo (s)	Tiempo del sensor (s)	Mag X	Mag Y	Mag Z	Precisión
Presión	Tiempo (s)	Tiempo del sensor (s)	Presión (Milibares	Precisión		
Luminancia	Tiempo (s)	Tiempo del sensor (s)	RSS Luminancia (lux)	Precisión		
Humedad	Tiempo (s)	Tiempo del sensor (s)	proximidad	Precisión		
Temperatura ambiente	Tiempo (s)	Tiempo del sensor (s)	Temperatura (Celsius)	Precisión		
Orientación	Tiempo (s)	Tiempo del sensor (s)	X	Y	Z	Precisión
GNSS	Tiempo (s)	Tiempo del sensor (s)	Latitud	Longitud	Altitud	Precisión
WiFi	Tiempo (s)	Tiempo del sensor (s)	Identificador SSID	MAC SSID	Frecuencia	RSS
Bluetooth	Tiempo (s)	Nombre	Dirección MAC	RSS		
Sonido	Tiempo (s)	RMS	Presión (Pa)	SPL(dB)		
Posición	Número POSI	Tiempo	Latitud	Longitud	Planta	Identificador

4.2.2. Análisis de la Base de datos

En esta sección se expone el análisis y tratamiento llevado a cabo en la base de datos. Tras haber realizado la recogida de los datos, se empezó el tratamiento de ellos mediante la creación de gráficas para observar cómo estaban compuestos estos datos. Como la información viene en los ficheros, se tuvo primero que realizar un filtrado para almacenar los que son relevantes. Para obtener los datos precisos de las 7 localizaciones específicas en este caso, se obtienen observando sus valores de tiempo, latitud y longitud de las líneas que contienen la palabra clave “POSI”, obteniendo la Figura 4.3 mostrada en la Sección 4.2.1. Analizando los valores y tratándolos con varias funciones del programa, se observa que a la hora de realizar este recorrido se hacía dos veces, lo que significa que la mitad de los valores de la base de datos correspondían a las posiciones por las que ya se había pasado anteriormente. Los recorridos de ida y vuelta los podemos observar en las Figuras A.1a y A.1b, que se encuentran en el Apéndice A.

En el caso de almacenar datos correspondientes a la luminancia recibida por el receptor móvil, se almacena ese valor del RSS de la luz, y el tiempo. Estos datos los obtenemos por las líneas del fichero que contienen la palabra clave “LIGH”. Los datos de luminancias son muy numerosos, se obtienen datos tanto de los que se recibe en las posiciones destacadas en la Figura 4.3, como de todo el trayecto. En la Figura 4.4a observamos la variación de la señal recibida lumínica a lo largo del recorrido. El principal problema de los datos, es que en algún punto el RSS de la luminancia obtenía un pico de valor del RSS muy alto, por lo que no se podía realizar una buena comparativa de valores por este motivo.

A continuación se detalla el procesado de los datos realizado sobre la intensidad de luz

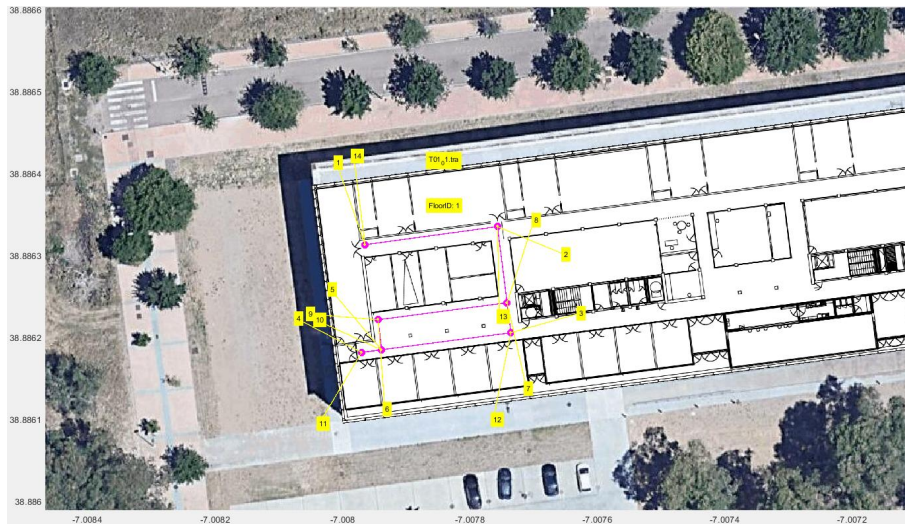


Figura 4.2: Esquema descriptivo del recorrido y entorno para la toma de la base de datos.

adquiridas de la base de datos. En primer lugar, tan solo se seleccionarán las intensidades de luz capturadas en las siete posiciones de estudio del trayecto. Para ello, como en la base de datos se encuentra la marca temporal indicando cuándo el usuario se encuentra en cada una de las siete posiciones, se almacenan los valores de la intensidad de la luz correspondientes a la misma marca temporal en el que el usuario se encontraba en un rango de tiempo de 0,5 segundos de rango temporal con respecto a la marca de tiempo almacenada cuando se encontraba en la posición específica. Este proceso se repitió para las diez iteraciones realizadas de este trayecto.

Se hizo una primera prueba con los datos de la luz que consistía en realizar una normalización de los datos sin filtrarlos por cada una de las siete posiciones, como podemos apreciar en la Figura A.2. El proceso de normalización consiste en pasar cada uno de los valores que tenemos por la Ecuación (4.1), donde el valor de X es cada uno de los RSS de luminancias, y los valores máximos y mínimos de entre todas las RSS que existen en la base de datos. La Figura A.2 muestra el resultado de haber realizado esta transformación de los datos:

$$X' = \frac{(X - X_{min})}{(X_{max} - X_{min})} \quad (4.1)$$

Como el resultado de los valores no mostraba una diferenciación, se pasó a realizar el filtrado de los datos por cada posición, obteniendo la Figura 4.4b, donde los datos mostraban una mejor diferenciación sin apreciarse picos de RSS de luminancias que no permitían observar mejor los resultados.

A partir de estos resultados obtenidos, se realiza el cálculo de la media aritmética del RSS de luminancia en todos los ficheros, obtenida por la Ecuación (4.2), donde cada X_i es cada uno de los datos y n es el número de datos que tenemos para realizar esta

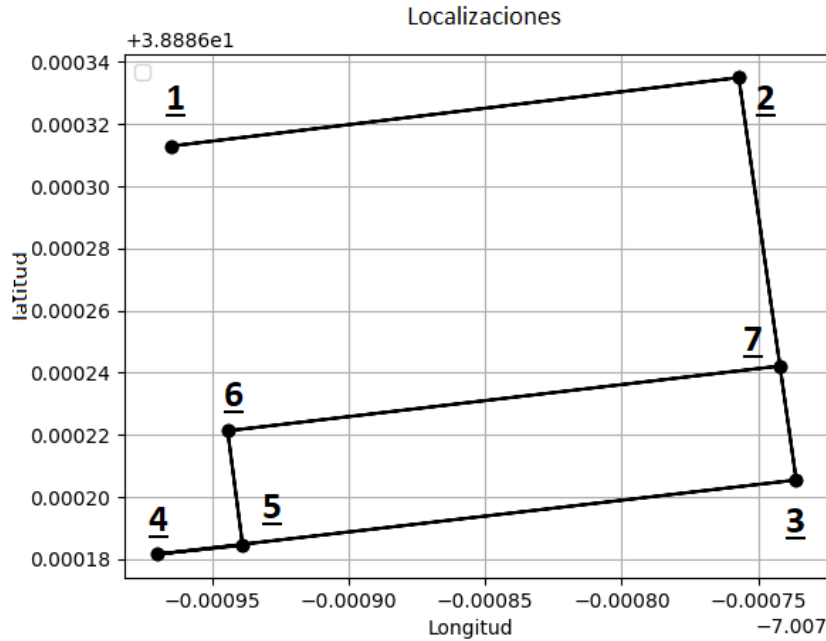
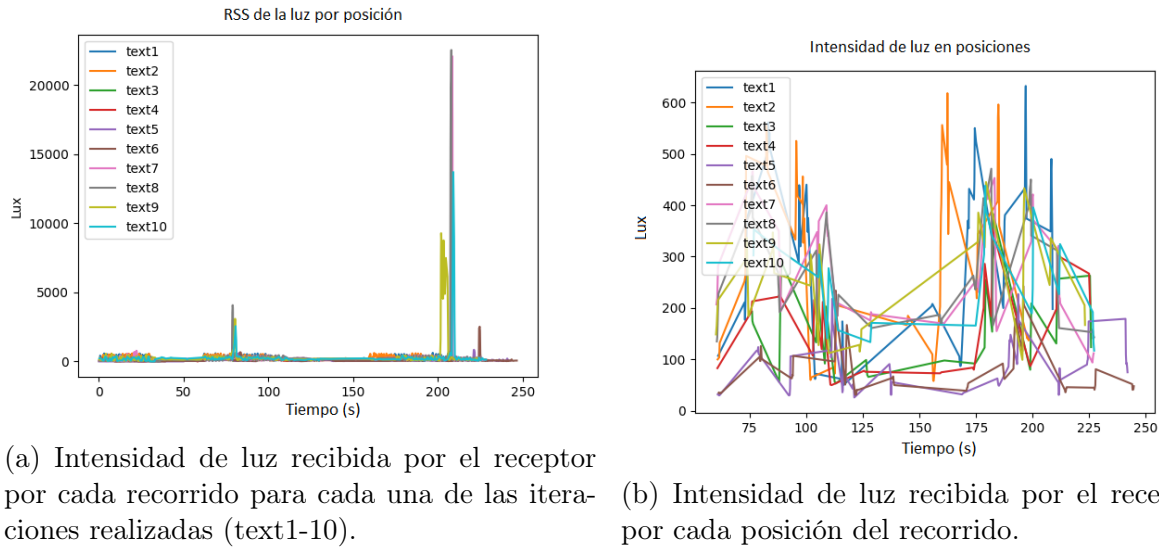


Figura 4.3: Esquema descriptivo de las localizaciones del recorrido, numeradas indicando cada localización.

media. Con esta media realizada por posición, se realizan gráficas para mostrar los valores en cada posición, viendo su parentesco con la media aritmética de los datos y quedando como resultado la Figura 4.5. También se estudia el valor de cada RSS restando el valor de la media aritmética de los valores en cada posición, para centrar los resultados en el 0. Esto se observa en la Figura A.3, que se encuentra en el Apéndice A.

$$X' = \frac{(X_1 + X_2 + X_3 + \dots + X_n)}{n} \quad (4.2)$$

Para el estudio de los valores del *WiFi*, se realiza primero una distinción de los valores del RSS del *WiFi* por cada una de las posiciones que tenemos. Tras ello, se divide el proceso en 5, por identificador SSID que tenemos del *WiFi* y por una de las frecuencias del sistema y poder apreciar sus valores. Tenemos en total 5 identificadores fundamentalmente del *WiFi*, que son los SSID del 1 al 5, y observamos los valores del RSS de ellos dando una frecuencia para cada identificador. Las frecuencias utilizadas son de 2437, 2447, 2472 y 5560 MHz. Para estos valores se obtienen gráficas mostrando sus valores del RSS por cada SSID y frecuencia utilizada en cada caso, dando como resultado las siguientes Figuras: *SSID1*–2437 MHz en la Figura 4.6a, y las Figuras en el Apéndice A, *SSID2*–2447 MHz MHz en la Figura A.4a, *SSID3*–2422 MHz MHz en la Figura A.4b, *SSID4*–2472 MHz MHz en la Figura A.5a y *SSID5*–2437 MHz MHz en la Figura A.5b. Al igual que para el RSS de las luminancias, también se realiza el cálculo de la media aritmética de los valores del RSS del *WiFi* en cada caso mostrado antes, es decir, la media para los valores con



(a) Intensidad de luz recibida por el receptor por cada recorrido para cada una de las iteraciones realizadas (text1-10).

(b) Intensidad de luz recibida por el receptor por cada posición del recorrido.

Figura 4.4: Gráficas correspondientes a los valores del RSS de la luz.

SSID1 y frecuencia 2437 MHz, SSID2 y frecuencia 2447 MHz, etc, y realizamos la resta de los valores menos el valor de la media calculada por cada posición en cada caso, dando como resultado la siguiente Figura 4.6b mostrada, y el resto de Figuras en el Apéndice A A.6a - A.7b.

También se realizan gráficas del estilo *box-plot*, que sirven para ver la aproximación de los valores y ver qué diferenciación con el RSS se encuentra. Cuanto más distinto sea entre cada posición, mejor será el clasificador a la hora de predecir en cuál de las siete posiciones se encuentra. Para el caso del RSS de las luminancias se obtiene la Figura 4.7a, donde se ven los valores del RSS de las luminancias y sobre qué valores rondan. También se realiza la gráfica del *box-plot* del RSS de las luminancias restando el valor de la media por posición, obteniendo la Figura 4.7b.

Para los valores del *WiFi* también se realizan gráficas *box-plot*, donde se ve la tendencia del RSS del *WiFi* por cada posición del trayecto. Se obtiene una gráfica por cada caso estudiado anteriormente, obteniendo las Figuras correspondientes a cada caso 4.8a mostrada, y el resto de Figuras que se encuentran en el Apéndice A para el resto de casos A.8a - A.11a. También se realiza la gráfica *box-plot* correspondiente a cada caso restando el valor de la media del RSS del *WiFi* por cada posición, obteniendo la Figura 4.8b mostrada, y el resto de Figuras en el Apéndice A A.8b - A.11b.

Para el análisis del *WiFi* a la hora de aplicar el algoritmo de ML, se aplicará una frecuencia de 2412 MHz en uno de los experimentos de la toma de datos, y otra frecuencia de 2462 MHz y se aplicarán los valores correspondientes al SSID1. Esto es debido que a la hora de hacer pruebas es necesario que exista una cantidad de valores del RSS suficiente para que, a la hora de entrenar al modelo introduciendo los datos, tenga suficientes para

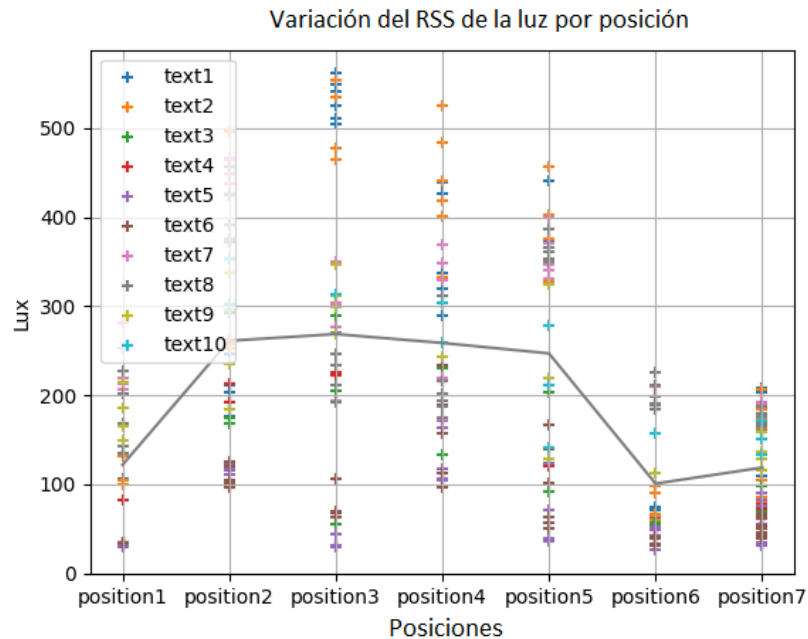


Figura 4.5: Intensidad de luz recibida por el receptor por cada recorrido en sus posiciones.

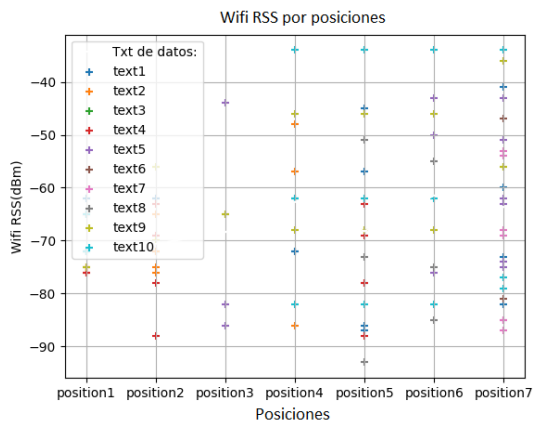
introducir a cada iteración del experimento un mínimo de un valor por clase, y se ha realizado un estudio de las diferentes combinaciones existentes variando el SSID y la frecuencia, y se encontró que la mayor cantidad de valores y diferenciación entre posiciones se encontraba en estas combinaciones indicadas. La mayoría de combinaciones probadas no se podían implementar en el algoritmo debido a la falta de datos suficientes para realizar experimentos.

En esta Figura 4.9a vemos los valores correspondientes al RSS del *WiFi* por posición en el caso de la primera frecuencia, y su Figura *box-plot* es 4.9b. En la Figura 4.10a corresponde al uso de la segunda frecuencia, y su Figura *box-plot* es 4.10b.

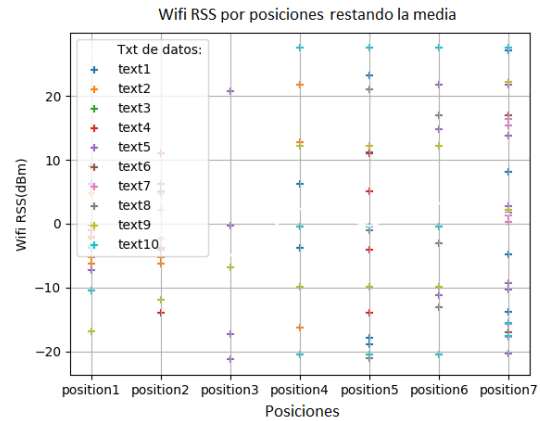
4.2.3. Arquitectura del Programa

Este TFG se ha realizado mediante programación de *scripts* en el lenguaje de programación *Python*, utilizando el editor de texto *Visual Studio Code*. Este programa ha ido creciendo y almacenándose poco a poco en *GitHub*. *GitHub* es un almacén de repositorios que está especializado en proyectos de cualquier índole. Para cada actualización del programa se ha creado una nueva rama, mediante el comando en el terminal “git checkout -b”. En este enlace¹ se puede observar el seguimiento realizado para la creación del proyecto. Las librerías externas requeridas para realizar las funciones del programa son *basemap*, *matplotlib* y *scikit-learn*. *Basemap* es un *toolkit* de *matplotlib* que nos facilita la tarea de representar información 2D sobre mapas. *Matplotlib* es una librería *Python*

¹<https://github.com/JLopezcano/TFG-2023>



(a) *Received Signal Strength* del *WiFi* por posición y *SSID1* con frecuencia 2437 MHz.



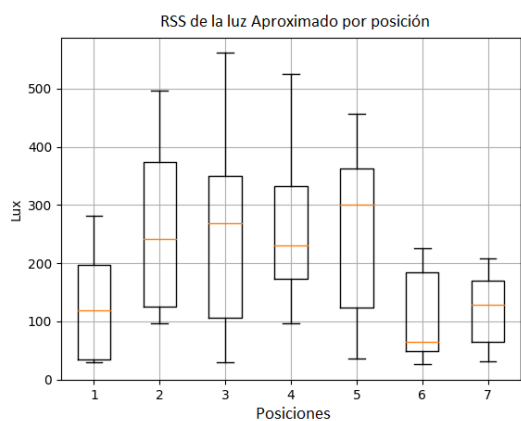
(b) *Received Signal Strength* del *WiFi* por posición del *WiFi* y *SSID1* con frecuencia 2437 MHz, restando el valor de su media por posición.

Figura 4.6: Gráficas correspondientes a los valores del RSS del *WiFi*.

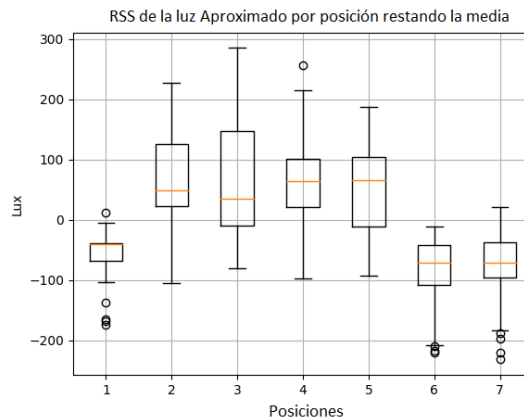
open source que permite crear visualizaciones de datos. *Scikit-learn* es una biblioteca de *Python* que proporciona acceso a versiones eficaces de muchos algoritmos comunes, como el SVM implementado en este TFG.

Los *scripts* se han dividido en carpetas, separado de manera que en estas carpetas están organizados los *scripts* por tipo de tratamiento de datos aplicado en ellos. Las carpetas en las que se divide el programa son las siguientes:

- Carpeta principal, en la que se encuentran todas las subcarpetas y el *script* “*main.py*”, en el que se realiza la ejecución del programa, es decir, el programa principal. Este *script* contiene la llamada a las distintas funciones que aportan los valores y gráficas. En esta carpeta se encuentra el resto de las carpetas del programa, que son “clases” y “servicios”. También hay un archivo de texto llamado *README.md*, dedicado a dar las indicaciones mencionadas al final de la Sección 4.2.3 para la ejecución.
- Clases, en la que se encuentran las clases utilizadas en el programa. Las clases del programa se encuentran en el *script* “*classes.py*”. En el *script* se encuentra la clase llamada “*data*”, en la que se almacenan los valores recogidos en los ficheros de texto, y que se utilizan para realizar las funciones del programa.
- Servicios, en esta carpeta se encuentran 3 carpetas:
 1. gráficas: aquí encontramos dos *scripts*, que son “*graficas.py*” y “*graficasML.py*”, los cuales contienen las gráficas mostradas en las Figuras, que muestran los



(a) *Box-plot* del *Received Signal Strength* de la luz por cada posición del recorrido.



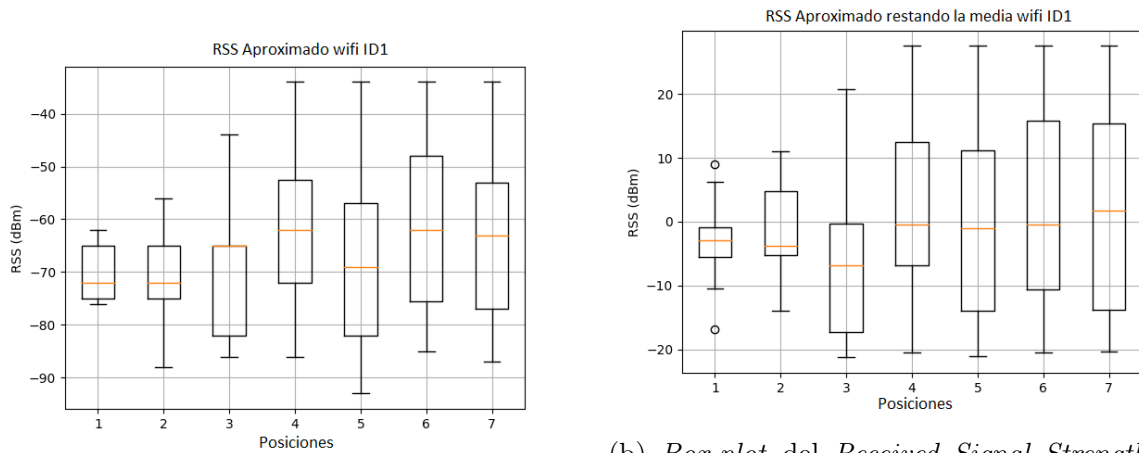
(b) *Box-plot* del *Received Signal Strength* de la luz por cada posición del recorrido, restando el valor de la media del RSS de cada posición.

Figura 4.7: Gráficas *box-plot* correspondientes a los valores del RSS de la luz.

valores y datos representados después de su tratamiento en la base de datos. En el segundo tenemos el estudio del algoritmo de ML realizado. También se encuentra, después de haber ejecutado una vez el programa, que en esta carpeta se guarda caché en una carpeta llamada “`__pycache__`”.

2. mapeo: En esta carpeta se encuentran 4 ficheros, que son “*mapeos.py*”, “*mapeosML.py*”, “*mapeosWifi.py*”, “*mapeosWifiFilters.py*” y “*mapeosWifiSingle.py*”. Estos ficheros son los que contienen las funciones del programa que realizan las operaciones matemáticas o introducen los datos en los *arrays* requeridos. Básicamente están dedicados a realizar las operaciones interiores del programa para mostrar los datos tras su tratamiento. También se encuentra, después de haber ejecutado una vez el programa, que en esta carpeta se guarda caché en una carpeta llamada “`__pycache__`”.
3. rutas: En esta carpeta se encuentra el *script* “*rutas.py*”. En este fichero se encuentran las funciones dedicadas a la obtención de los datos desde los bloc de notas mencionados en las anteriores secciones. Estos valores se almacenan en la base de datos del programa. También se encuentra, después de haber ejecutado una vez el programa, que en esta carpeta se guarda caché en una carpeta llamada “`__pycache__`”.

Aquí se expone la forma de ejecutar el código para observar los datos y resultados. El archivo ejecutable es el “*main.py*”, al ejecutarlo aparecerán los datos y gráficas. Para ello, se ha dispuesto de un *input* en la ejecución del programa, el cuál pedirá que se inserte un valor numérico del 0 al 9. Al insertar este valor, podrá observar las diferentes gráficas explicadas por pantalla. Para poder realizar la ejecución, es necesario tener instaladas las librerías



(a) *Box-plot* del *Received Signal Strength* del *WiFi* con SSID1 y con frecuencia 2437 MHz por cada posición del recorrido.

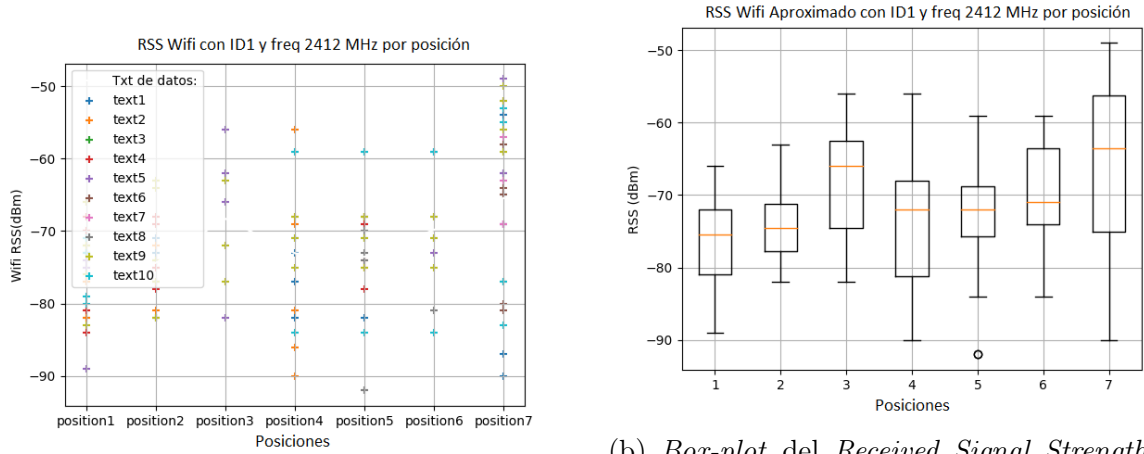
(b) *Box-plot* del *Received Signal Strength* del *WiFi* con SSID1 y con frecuencia 2437 MHz por cada posición del recorrido, restando el valor de la media del RSS de cada posición.

Figura 4.8: Gráficas *box-plot* correspondientes a los valores del RSS del *WiFi*.

externas y *toolkits* explicados en el inicio esta sección. A medida que van apareciendo las gráficas, si se cierra la ventana de ella, se verá la siguiente, y así continuamente hasta que se termina la ejecución. En el caso de los resultados del clasificador, aparecerán los valores ordenados en el terminal de la pantalla, apareciendo primero cada uno de los resultados de las 10 veces que se realiza el proceso de clasificación de la luz, y 2 en caso del *WiFi*, y al final aparecen las métricas y matriz de confusión, primero correspondiente al clasificador con *kernel polynomial* y después a los resultados del clasificador con *kernel RBF*.

4.2.4. Métricas y Algoritmo de ML

Cuando ya se han obtenido los datos filtrados y tratados se implementa un clasificador SVM por el que se pasarán los datos para predecir la posición del usuario en el recorrido. El clasificador que se utiliza es el SVM, y se introducen los datos tal y como los hemos filtrado anteriormente, aplicando nuestros valores del RSS de las luminancias y del *WiFi*. A la hora de aplicarlo en el clasificador hacemos una división de los datos en datos de entrenamiento y datos de test. Esta división se realiza de manera que tendremos un 70 % de datos de entrenamiento y un 30 % de datos de test, tanto por los valores de la luz como del *WiFi*. Esta división y cálculo de datos se realiza de manera aleatoria, cogiendo aleatoriamente los valores que tenemos almacenados para esa división 70/30 %, por lo que cada vez que se pasa la base de datos por el clasificador da resultados distintos. Este experimento se realiza 10 veces para cada vez que se realiza la predicción en el caso de la luz, y 2 en el *WiFi*, y posteriormente se realiza la media de los resultados obtenidos, tanto para la luz como el *WiFi*. En este experimento se tienen en cuenta 2 tipos de *kernel*, el



(a) *Received Signal Strength* del *WiFi* por posición y *SSID1* con frecuencia 2412 MHz.

(b) *Box-plot* del *Received Signal Strength* del *WiFi* por posición y *SSID1* con frecuencia 2412 MHz.

Figura 4.9: Gráficas correspondientes a los valores del RSS del *WiFi*.

polynomial y el RBF, explicados en la Sección 3.4.4.

A la hora de aplicar el clasificador vamos a obtener diferentes valores, que son la matriz de confusión, y las métricas. Las métricas son de dos tipos, la binarias, que corresponden a las métricas *accuracy*, precisión, *recall* y *f1-score*; y las métricas multiclase que son las mismas que en la binaria, pero se incluyen también los valores de *Macro Average*, que son el cálculo de la métrica de interés para cada clase y luego promedia estos valores, y *Micro Average*, que es el cálculo de las métricas sumando los verdaderos positivos, falsos positivos, falsos negativos y verdaderos negativos.

- La matriz de confusión es una herramienta que sirve para visualizar el desempeño de un algoritmo de aprendizaje supervisado, es decir, nos indica como de efectivo es el algoritmo implementado. Los puntos localizados en la diagonal principal de la matriz indican los resultados acertados en el modelo para un valor real positivo, es decir, indica que es un positivo y la prueba lo predijo correctamente.
- La *accuracy* es el ratio entre las predicciones correctas y las precisiones totales, es decir, que valor (cuyo máximo posible es 1) tiene un clasificador de precisión sobre todos los valores de la base de datos y cuántos acierta a qué clase corresponde. La fórmula para calcular la *accuracy* de una clase es la que se ve en la Ecuación (4.3), siendo N_{PC} el número de predicciones correctas y N_{PT} el número total de predicciones.

$$Accuracy = \frac{(N_{PC})}{(N_{PT})} \quad (4.3)$$

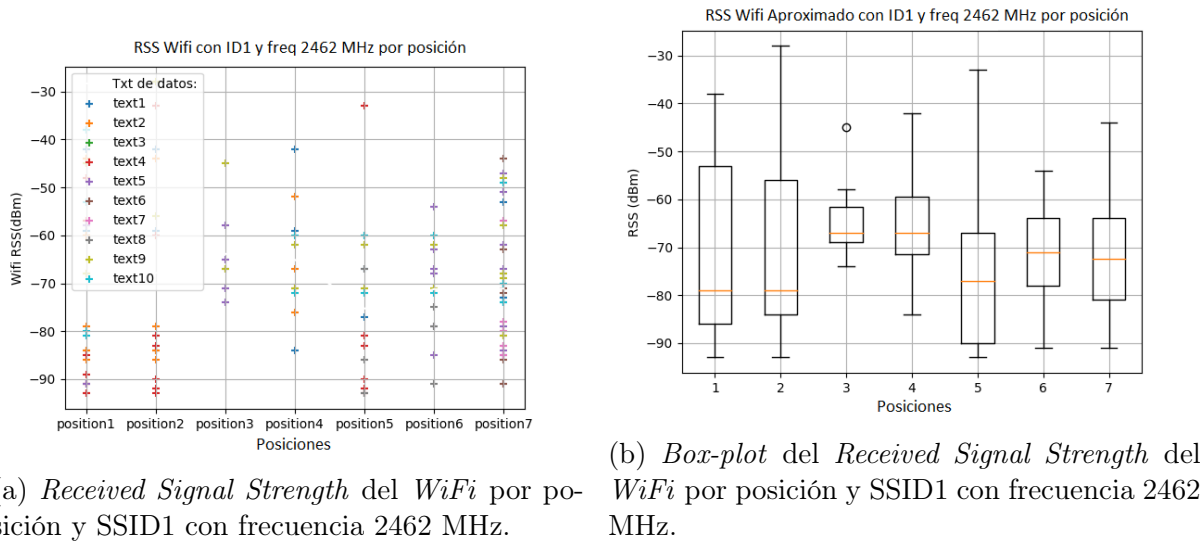


Figura 4.10: Gráficas correspondientes a los valores del RSS del *WiFi*.

- La precisión es la tasa de valores positivos verdaderos que tiene un algoritmo, es decir, la tasa que define cuántos de los valores que ha considerado como pertenecientes a una clase en efecto pertenecen a esa clase. La fórmula para calcular la precisión de una clase es la que se ve en la Ecuación (4.4), siendo VP el número de predicciones correctas y FN el número de falsos positivos.

$$Precision = \frac{(VP)}{(VP + FP)} \quad (4.4)$$

- El *recall* es la tasa que define el valor de la proporción de verdaderos positivos frente a falsos negativos, cómo de bien diferencia entre un valor positivo verdadero a un valor negativo que realmente no lo era. La fórmula para calcular el *recall* de una clase es la que se ve en la Ecuación (4.5), siendo VP el número de predicciones correctas y FN el número de falsos negativos.

$$recall = \frac{(VP)}{(VP + FN)} \quad (4.5)$$

- El *f1-score* es el alta precisión y sensibilidad. Es la tasa que define si un modelo ha escogido de manera perfecta esa clase y acertando. La fórmula para calcular el *f1-score* de una clase es la que se ve en la Ecuación (4.6), siendo VP el número de predicciones correctas y FN el número de falsos negativos.

$$F1 = 2 \cdot \frac{(Precision \cdot Recall)}{(Precision + recall)} \quad (4.6)$$

- El valor de la desviación típica nos proporcionará cuánto varía el valor de una

métrica con respecto a la media aritmética de esas métricas, es decir, cuánto puede variar el valor de una métrica con respecto al valor de la media de esa métrica. La desviación típica se calcula como se ve en la Ecuación (4.7). En ésta fórmula, σ representa el valor de la desviación típica, x_i representa cada valor individual en el conjunto de datos, μ es la media aritmética de los valores y N es el número de observaciones en la muestra.

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad (4.7)$$

- El *support* es el número de apariciones reales de la clase en el conjunto de datos especificado.
- *Macro average* es el valor que permite evaluar mejor cómo de equilibrado es el comportamiento del clasificador.
- *Weighted average* es el valor de la media aritmética de las métricas medidas por cada clase, es decir, indica el valor de la media aritmética de Precisión, *Recall* y *F1-Score*.

Capítulo 5

Resultados y Discusión

5.1. Introducción

Tras haber realizado el filtrado y tratamiento de los datos, se introducen en el algoritmo de ML. En este capítulo se presentarán los resultados obtenidos mediante la implementación de un clasificador SVM de *kernel polynomial* y de *kernel RBF*, aplicados tanto para los datos obtenidos del sensor de la luz como para los datos adquiridos relativos a la señal de *WiFi*. En la Sección 5.2.1 se verán los resultados para los valores del RSS de la luz, y en la Sección 5.2.3 y Sección 5.2.2 para los valores del *WiFi*. Posteriormente en la Sección 5.3 se realiza una comparativa de los resultados obtenidos.

5.2. Resultados Clasificadores

Para la implementación de los datos de la luz partimos como base de datos que implementaremos en el clasificador los RSS de luz y las posiciones a los que corresponden esos valores como clase. En total tenemos 7 posiciones en la trayectoria. La división de los datos de entrenamiento y datos de test se divide en un 70% de los datos en *training* y un 30% de los datos en *test*. Con los datos que hemos dividido previamente para *training* se rellena el modelo de entrenamiento para el clasificador, y los datos de *test* se utilizan para realizar la predicción. Cuando ésta se ha realizado se calcula la matriz de confusión y las métricas. Este proceso se realiza 10 veces para validar la propuesta, dando como resultado 10 valores de métrica diferentes y 10 matrices de confusión, porque la división de datos de entrenamiento y test se realiza de forma aleatoria.

Para la implementación de los datos del *WiFi* partimos con los valores del RSS y sus posiciones como clase. En este caso, se han seleccionado los valores que correspondían al SSID1 y que tuvieran una frecuencia de 2412 y 2462 MHz. La división de los datos de

training y de *test* es la misma que en el caso de la luz, 70/30%, y la aplicación en el clasificador de los datos se realiza de la misma forma, primero se entrena el modelo de entrenamiento del clasificador con los datos de *training* y después se realiza la predicción con los datos que se han dividido para *test*. Después se realiza el cálculo de la matriz de confusión y de las métricas, y se repite el proceso 2 veces obteniendo de la base de datos los valores de forma aleatoria para crear los datos de entrenamiento y de test, para que se hagan varias predicciones diferentes.

Al final de este proceso, tanto para la luz como para el *WiFi*, se calcula la media de los valores obtenidos y se muestran, tanto para las matrices de confusión como para los valores de precisión, *recall*, *f1-score*, etc. Los clasificadores implementados fueron SVM con dos tipos de *kernel* distintos: *polynomial* y RBF.

5.2.1. Resultados RSS luminancias SVM *polynomial*

Como se ha mencionado en este TFG, para el clasificador SVM *polynomial* se han llevado a cabo 10 iteraciones, y se ha calculado la media de los valores tanto para las métricas como para la matriz de confusión. Los valores que introducimos en el clasificador son los que se aprecian en la Figura 4.7a, y como se observa tenemos en total las 7 localizaciones y el valor del RSS para cada localización. Para conocer el desempeño de un algoritmo se utiliza la matriz de confusión, pues esta representa las predicciones realizadas por él.

En las matrices de confusión los valores correspondientes a las columnas representan el número de predicciones de cada clase y las filas muestran qué tipos de aciertos y errores está teniendo nuestro modelo al pasar el proceso de aprendizaje con los datos. Los valores que se encuentran en la parte superior derecha de la diagonal muestran los falsos positivos del modelo, valores que ha indicado el algoritmo implementado como predicciones, pero que no lo predijo correctamente, puesto que en nuestro caso se encontraba en otra posición. Esto sería si el usuario se encuentra en la posición 4, pero el algoritmo ha predicho que se encuentra en la posición 5. Los valores que se encuentran en la parte inferior izquierda de la diagonal principal son los falsos negativos, es decir, que el algoritmo ha predicho que no se encuentra en una posición y realmente si se encontraba en ella.

El proceso de coger los valores de la base de datos proporcionada al algoritmo y dividirlo en un porcentaje de 70/30% se ha realizado un total de 10 veces, y para cada una de ellas se ha calculado el valor de las métricas y de la matriz de confusión. La matriz de confusión media en este caso SVM *polynomial* con los RSS de las luminancias es el que se ve en la Figura 5.1.

Ésta matriz tiene como diagonal principal [0; 2,1; 3,3; 1,4; 0,3; 0,0; 24], cada posición es indicada por cada índice de la diagonal, es decir, el primer valor de la diagonal representa

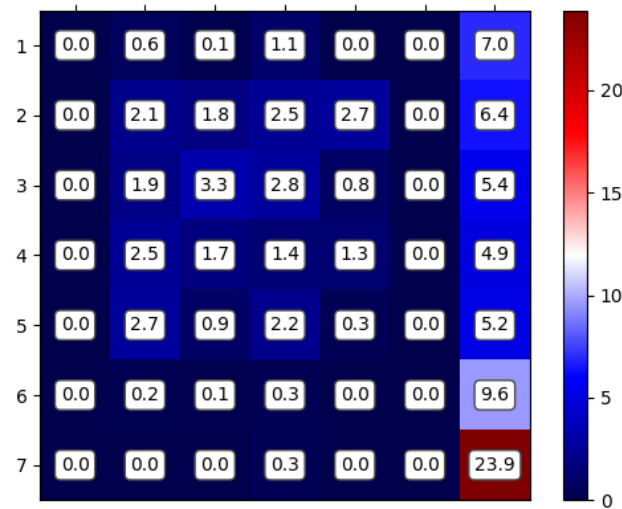


Figura 5.1: Matriz de confusión media del RSS de la luz para el SVM *polynomial*.

Tabla 5.1: Datos de la media aritmética de las métricas del experimento con el RSS de la luz en las posiciones para el SVM *polynomial*.

<i>Accuracy</i>	Precisión	<i>Recall</i>	<i>F1-Score</i>
$0,330 \pm 0,034$	$0,157 \pm 0,040$	$0,229 \pm 0,018$	$0,156 \pm 0,026$

el acierto en la primera posición, el segundo índice representa la segunda posición, y así sucesivamente, por lo que nos indica que tiene un valor alto de predicción para la posición 7, y un valor decente para la posición 3. Esto se debe a que corresponden a un valor alto en comparación al resto de la fila, pero al no ser el valor más alto, no es una predicción buena, salvo en la posición 7. Al haber realizado el experimento con 10 iteraciones, también tenemos las métricas de cada iteración realizada. El valor de la media de las métricas de todas las iteraciones es el que vemos en la Tabla 5.1.

Cada clase del experimento serían las posiciones de la trayectoria, y los valores de las métricas que aparecen aquí corresponden a las métricas explicadas en la Sección 4.2.4. El valor del *Accuracy* en este caso indica que de todos los valores del RSS introducidos en el algoritmo, acierta a qué clase pertenece el 33% de las veces, es decir, 1/3 de los resultados los clasifica correctamente. El valor de la Precisión indica que el 16% de las veces en las que indica que un valor pertenece a una clase sí pertenece a esa clase, siendo un valor bajo. El valor del *recall* indica que el algoritmo tiene un 23% de tasa de valores verdaderos positivos frente a falsos negativos, y el *F1-Score* indica que el 16% elige a la perfección que un valor es de la clase que a la que pertenece.

Para cada experimento, se ha realizado el cálculo de la desviación típica. Los valores de la desviación típica en el caso del SVM *polynomial* se calculan para cada una de las

Tabla 5.2: Datos de la media aritmética de las métricas del experimento con el RSS de la luz en las posiciones para el SVM *RBF*.

<i>Accuracy</i>	Precisión	<i>Recall</i>	<i>F1-Score</i>
$0,332 \pm 0,048$	$0,181 \pm 0,060$	$0,251 \pm 0,019$	$0,176 \pm 0,030$

métricas, dando como resultados los que se representan en la Tabla 5.1. Como podemos observar, se obtienen unos valores de desviación típica de 0,03 aproximadamente, por lo que podemos encontrar que sus métricas varían ese valor con respecto a la media obtenida.

Para el experimento del clasificador SVM RBF utilizado se han llevado a cabo 10 iteraciones, y se ha calculado la media de los valores tanto para las métricas como para la matriz de confusión. Como este experimento se ha realizado con 10 iteraciones, hemos obtenido 10 matrices de confusión distintas, y se ha calculado la media aritmética de todas ellas, mostrada en la Figura 5.2.

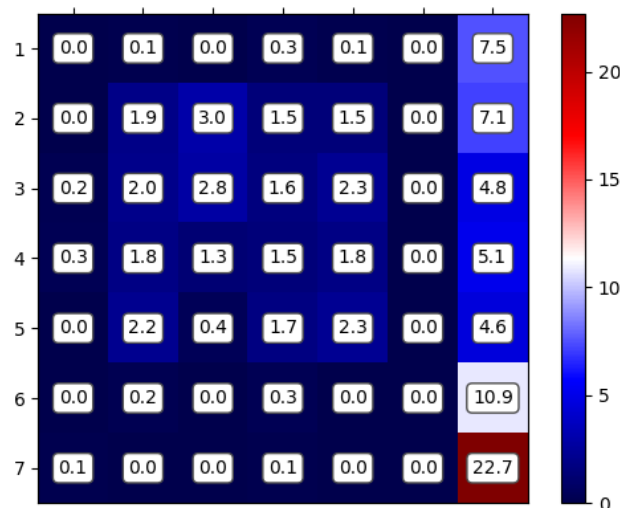


Figura 5.2: Matriz de confusión media del RSS de la luz para el SVM RBF.

La matriz media tiene como diagonal principal $[0; 1,9; 2,8; 1,5; 2,3; 0,0; 22,7]$, lo que nos indica que tiene alto valor de predicción para las posición 7. Esta clase sería la única con una buena tasa de predicción. Al haber realizado el experimento con 10 iteraciones, también tenemos las métricas de cada iteración realizada, y su media de métricas de todas las iteraciones es el que vemos en la Tabla 5.2.

El valor del *Accuracy* en este caso indica que de todos los valores del RSS introducidos en el algoritmo, acierta a qué clase pertenece el 33,2% de las veces los clasifica correctamente. El valor de la *Precisión* indica que el 18% de las veces en las que indica que un valor pertenece a una clase sí pertenece a esa clase, siendo un valor mejor que en el caso del *polynomial*. El valor del *recall* indica que el algoritmo tiene un 25% de tasa de

valores verdaderos positivos frente a falsos negativos, y el *F1-Score* indica que el 17,5% elige a la perfección que un valor es de la clase que a la que pertenece. Los valores de la desviación típica en el caso del SVM RBF se calculan para cada una de las métricas, dando como resultados los que se representan en la Tabla 5.2. Como podemos observar, se obtienen unos valores de desviación típica de 0,045 aproximadamente, por lo que podemos encontrar que sus métricas varían ese valor con respecto a la media obtenida.

5.2.2. Resultados RSS *WiFi* SVM *polynomial*

Para el clasificador SVM *polynomial* considerando los datos del *WiFi* se ha llevado a cabo el experimento con 2 iteraciones, y se ha calculado la media de los valores tanto para las métricas como para la matriz de confusión. En la base de datos del *WiFi* se ha hecho el experimento contemplando siempre el SSID1, pero modificando su frecuencia en dos posibles valores, primero con una frecuencia de 2412 MHz (frecuencia 1) y después con una frecuencia de 2462 MHz (frecuencia 2). Al realizar el primer caso de las 2 iteraciones realizados para el SVM *polynomial* en el caso del *WiFi* utilizando la frecuencia 1 obtenemos un valor medio de la matriz de confusión, que se observa en la Figura 5.3.

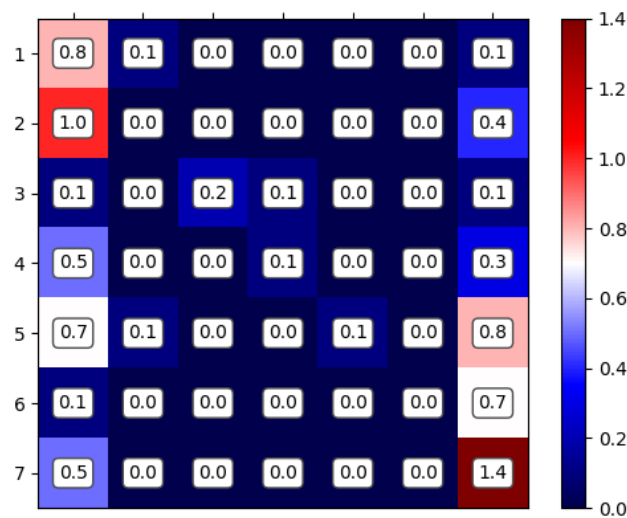


Figura 5.3: Matriz de confusión media del RSS del *WiFi* con SSID1 y frecuencia 2412 para el SVM *polynomial*.

La matriz media tiene como diagonal principal $[0,8; 0,0; 0,2; 0,1; 0,1; 0,0; 1,4]$, lo que nos indica que tiene alto valor de predicción para las posición 1 y 7. Estas dos clases tendrían una buena tasa de predicción. También encontramos que predice sin demasiado error algunos datos en la posición 3. El valor de la media de las métricas en este caso corresponde a lo que se observa en la Tabla 5.3.

Tabla 5.3: Datos de la media aritmética de las métricas del experimento con el RSS del *WiFi* con SSID1 y frecuencia 2412 MHz en las posiciones para el SVM *polynomial*.

<i>Accuracy</i>	Precisión	<i>Recall</i>	<i>F1-Score</i>
$0,389 \pm 0,039$	$0,184 \pm 0,038$	$0,224 \pm 0,005$	$0,218 \pm 0,012$

El valor del *Accuracy* indica una tasa del 39 %, por lo que clasifica correctamente los datos en su posición ese porcentaje de los valores considerando toda la masa de RSS. El valor de la Precisión indica que el 18 % de las veces en las que indica que un valor pertenece a una clase sí pertenece a esa clase. El valor del *recall* indica que el algoritmo tiene un 23 % de tasa de valores verdaderos positivos frente a falsos negativos, y el *F1-Score* indica que el 22 % elige a la perfección que un valor es de la clase que a la que pertenece.

Los valores de la desviación típica en el caso del SVM *polynomial* se calculan para cada una de las métricas, y en el caso del *WiFi* con el *kernel polynomial* y frecuencia 2412 MHz da como resultados los que se representan en la Tabla 5.3. Como podemos observar, se obtienen unos valores de desviación típica de 0,03 aproximadamente, por lo que podemos encontrar que sus métricas varían ese valor con respecto a la media obtenida.

Para el clasificador *polynomial* también se realiza el estudio para una frecuencia de 2462 MHz. Con esta frecuencia también se repite el experimento con 2 iteraciones, obteniendo de esta manera la matriz de confusión de la Figura 5.4.

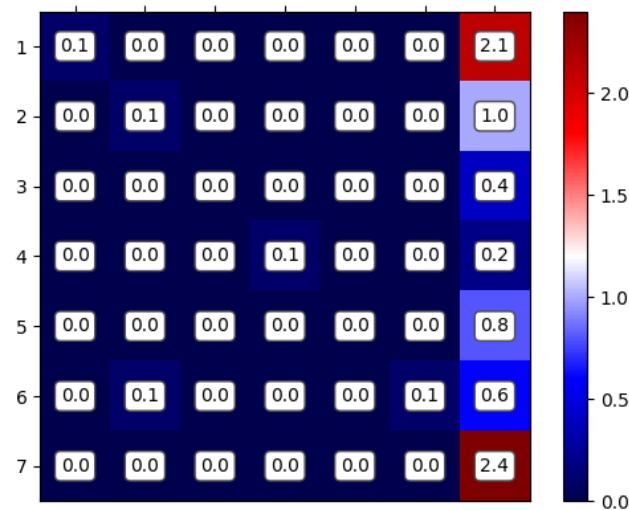


Figura 5.4: Matriz de confusión media del RSS del *WiFi* con SSID1 y frecuencia 2462 MHz para el SVM RBF.

La matriz media tiene como diagonal principal $[0,1; 0,1; 0,0; 0,1; 0,0; 0,1; 2,4]$, lo que nos indica que tiene alto valor de predicción para las posición 7. Esta clase tendrá una buena tasa de predicción. También encontramos que predice sin demasiado error algunos datos

Tabla 5.4: Datos de la media aritmética de las métricas del experimento con el RSS del *WiFi* con SSID1 y frecuencia 2462 MHz en las posiciones para el SVM *polynomial*.

<i>Accuracy</i>	Precisión	<i>Recall</i>	<i>F1-Score</i>
$0,408 \pm 0,040$	$0,178 \pm 0,014$	$0,243 \pm 0,008$	$0,251 \pm 0,016$

Tabla 5.5: Datos de la media aritmética de las métricas del experimento con el RSS del *WiFi* con SSID1 y frecuencia 2412 MHz en las posiciones para el SVM *RBF*.

<i>Accuracy</i>	Precisión	<i>Recall</i>	<i>F1-Score</i>
$0,352 \pm 0,013$	$0,201 \pm 0,022$	$0,230 \pm 0,018$	$0,234 \pm 0,011$

en la posición 4. El valor de la media de las métricas en este caso corresponde a lo que se observa en la Tabla 5.4.

Como se puede observar, tenemos un valor del *Accuracy* del 41 %, por lo que clasifica ese porcentaje de las veces sobre todos los datos en la clase correcta. la Precisión indica que el 18 % de las veces elige el valor de la clase a la que pertenece. El *Recall* y *F1-Score* indican tasas del 25 %. Los valores de la desviación típica en el caso del SVM *polynomial* se calculan para cada una de las métricas, y en el caso del *WiFi* con el *kernel polynomial* y frecuencia 2412 MHz da como resultados los que se representan en la Tabla 5.4. Como podemos observar, se obtienen unos valores de desviación típica de 0,02 aproximadamente, por lo que podemos encontrar que sus métricas varían ese valor con respecto a la media obtenida.

5.2.3. Resultados RSS *WiFi* SVM RBF

Para el clasificador SVM RBF considerando los datos del *WiFi* se han llevado a cabo el experimento con 2 iteraciones, y se ha calculado la media de los valores tanto para las métricas como para la matriz de confusión. También se ha implementado siempre el SSID1 modificando su frecuencia en dos posibles valores, con una frecuencia de 2412 MHz y con una frecuencia de 2462 MHz. Al realizar el primer caso de las 2 iteraciones realizadas para el SVM RBF en el caso del *WiFi* utilizando la frecuencia 1 obtenemos un valor medio de la matriz de confusión, que se observa en la Figura 5.5.

La matriz media tiene como diagonal principal $[0,9; 0,1; 0,0; 0,0; 0,2; 0,0; 1,0]$, que nos indica un alto valor de predicción para las posición 1 y 7. Estas dos clases tienen buena tasa de predicción con respecto al resto. El valor de la media de las métricas en este caso corresponde a lo que se observa en la Tabla 5.5.

El valor del *Accuracy* indica una tasa que indica que el 35 % de las veces clasifica correctamente los datos en su posición considerando toda la masa de valores. El valor de la Precisión indica que el 20 % de las veces en las que afirma y acierta si un valor pertenece a una clase. El valor del *recall* indica que el algoritmo tiene un 23 % de tasa de valores

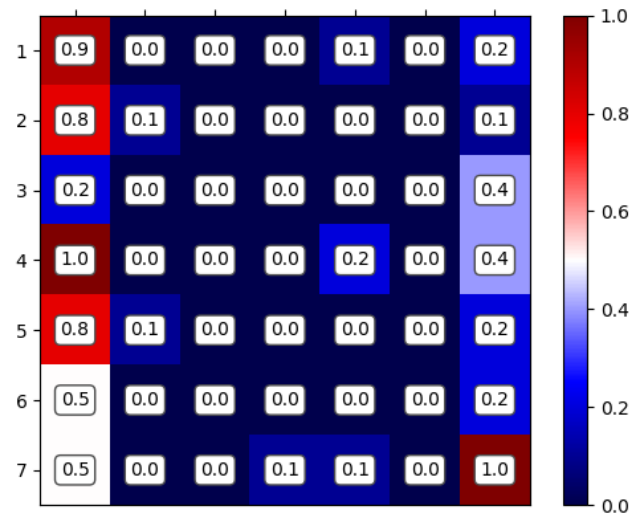


Figura 5.5: Matriz de confusión media del RSS del *WiFi* con SSID1 y frecuencia 2412 para el SVM RBF.

Tabla 5.6: Datos de la media aritmética de las métricas del experimento con el RSS del *WiFi* con SSID1 y frecuencia 2462 MHz en las posiciones para el SVM *RBF*.

<i>Accuracy</i>	Precisión	<i>Recall</i>	<i>F1-Score</i>
$0,369 \pm 0,040$	$0,180 \pm 0,026$	$0,251 \pm 0,018$	$0,274 \pm 0,031$

verdaderos positivos frente a falsos negativos, y el *F1-Score* indica que en el 23% de los datos elige a la perfección que un valor es de la clase que a la que pertenece.

Los valores de la desviación típica en el caso del SVM RBF se calculan para cada una de las métricas, y en el caso del *WiFi* con el *kernel* RBF y frecuencia 2412 MHz da como resultados los que se representan en la Tabla 5.5. Como podemos observar, se obtienen unos valores de desviación típica de 0,025 aproximadamente, por lo que podemos encontrar que sus métricas varían ese valor con respecto a la media obtenida.

Para el clasificador RBF se realiza otro estudio, ahora para una frecuencia de 2462 MHz. Con esta frecuencia también se realiza el experimento con 2 iteraciones, obteniendo la matriz de confusión de la Figura 5.6.

La matriz media tiene como diagonal principal $[0,2; 0,1; 0,0; 0,1; 0,1; 0,0; 1,8]$, y esto indica un alto valor de predicción mejor que para el resto de posiciones para la clase 7. El valor de la media de las métricas de este caso RBF corresponde a lo que se observa en la Tabla 5.6.

El valor del *Accuracy* indica una tasa del 37%, que es el porcentaje de las veces que clasifica correctamente los datos en su posición considerando todos los valores. La Precisión indica que el 18% de las veces afirma y acierta si un valor pertenece a una clase.

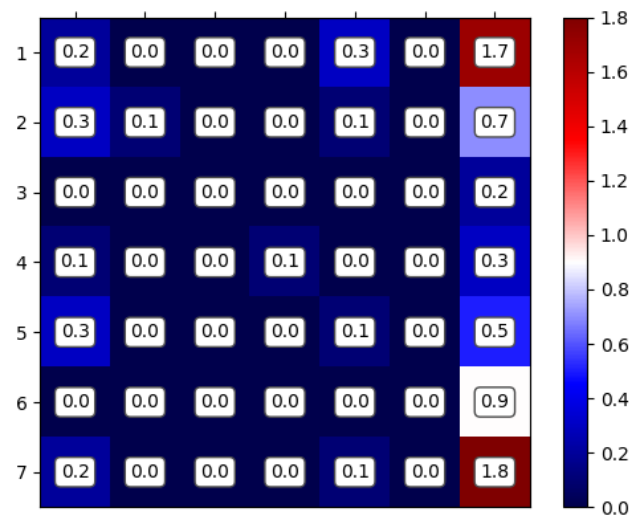


Figura 5.6: Matriz de confusión media del RSS del *WiFi* con SSID1 y frecuencia 2462 para el SVM RBF.

El valor del *recall* muestra que predice un 25% de valores verdaderos positivos frente a falsos negativos, y el *F1-Score* indica que en el 27% de los datos se eligen correctamente si un valor es de la clase que a la que pertenece.

Los valores de la desviación típica en el caso del SVM RBF se calculan para cada una de las métricas, y en el caso del *WiFi* con el *kernel* RBF y frecuencia 2462 MHz da como resultados los que se representan en la Tabla 5.6. Como podemos observar, se obtienen unos valores de desviación típica de 0,03 aproximadamente, por lo que podemos encontrar que sus métricas varían ese valor con respecto a la media obtenida.

5.3. Comparativa

Tras haber calculado los resultados de las métricas de cada uno de los experimentos de la luz, el valor de la media de ellos indica que en base al valor del *Accuracy*, tenemos una mejor tasa, aunque no por mucha diferencia, a la hora de implementar el SVM RBF. Esto quiere decir que tiene una pequeña mayor probabilidad de acierto a la hora de decir que un valor pertenece a una clase con respecto al SVM *polynomial*. En base a los valores de la Precisión también tiene una mejor tasa el RBF, por lo que predice de mejor forma que un valor pertenece a la clase que se ha seleccionado a priori. Tanto en el caso del valor del *Recall* como del *F1-Score* también hay una mejoría en el RBF con respecto al SVM *polynomial*, por lo que si se implementase un algoritmo de predicción en base a la luz por posición, sería una mejor opción implementar el RBF frente al *polynomial*. La

Tabla 5.7: Comparación de los valores de las métricas en el caso del RSS de la luz y el *WiFi* para tanto el *polynomial* como el RBF, contemplando ambas frecuencias.

	<i>Accuracy</i>	Precisión	<i>Recall</i>	<i>F1-Score</i>
RSS Luz <i>polynomial</i>	0,330 ± 0,034	0,157 ± 0,040	0,229 ± 0,018	0,156 ± 0,026
RSS Luz RBF	0,332 ± 0,048	0,181 ± 0,060	0,251 ± 0,019	0,176 ± 0,030
RSS <i>WiFi</i> <i>polynomial</i> Freq 1	0,389 ± 0,039	0,184 ± 0,038	0,224 ± 0,005	0,218 ± 0,012
RSS <i>WiFi</i> RBF Freq 1	0,352 ± 0,040	0,201 ± 0,014	0,230 ± 0,008	0,234 ± 0,016
RSS <i>WiFi</i> <i>polynomial</i> Freq 2	0,408 ± 0,013	0,178 ± 0,022	0,243 ± 0,018	0,251 ± 0,011
RSS <i>WiFi</i> RBF Freq 2	0,369 ± 0,040	0,180 ± 0,026	0,251 ± 0,018	0,274 ± 0,031

Tabla 5.8: Comparación de los valores de las métricas del RSS de la luz y el *WiFi*, considerando los más altos.

	<i>Accuracy</i>	Precisión	<i>Recall</i>	<i>F1-Score</i>
RSS Luz RBF	0,332 ± 0,048	0,181 ± 0,060	0,251 ± 0,019	0,176 ± 0,030
<i>WiFi</i> <i>polynomial</i> Freq 2	0,408 ± 0,013	0,178 ± 0,022	0,243 ± 0,018	0,251 ± 0,011

comparativa de los valores la podemos apreciar en la Tabla 5.7. Esto es debido a que tenemos una cantidad de datos alta, y el experimento se realiza con múltiples iteraciones.

Con respecto a los valores del *WiFi*, como podemos observar tenemos un valor medio de *Accuracy* de 39% en el caso de la implementación del SVM *polynomial* aplicando una frecuencia de 2412 MHz, y un valor de 41% al aplicar una frecuencia de 2462 MHz, frente al valor medio de 35% del SVM RBF utilizando la primera frecuencia y de 37% utilizando la segunda frecuencia. En el caso del *WiFi* tenemos mejores valores a la hora de aplicar el *polynomial* frente al RBF. Esto es influido porque tenemos una cantidad de datos en la base de datos y un número de iteraciones en el experimento menor que en el caso de la luz. La comparativa de los valores la podemos apreciar en la Tabla 5.7. Para la implementación de un algoritmo teniendo en cuenta estos valores se elegiría el *polynomial* por delante del RBF.

Si comparamos los valores obtenidos en las métricas de la clasificación, podemos observar que tenemos como valor máximo contemplando tanto el *polynomial* y el RBF de la luz, y como valor máximo contemplando ambas frecuencias y tanto el caso de *polynomial* y el RBF del *WiFi*, la comparación observada en la Tabla 5.8. Como podemos observar, los valores más altos se encuentran en la clasificación del *WiFi*, por lo que la predicción será mas exacta utilizando la base de datos filtrada para la clasificación del *WiFi* ante la de la luz. Debido a que los datos que se encontraban en la base de datos para los valores del RSS del *WiFi* en las 7 posiciones no era muy grande, no se pueden obtener mejores medidas y resultados para el caso del estudio, por lo que como observamos en la Tabla 5.7 no son cercanos al 1, lo que sería el mejor caso para una predicción. En el caso que encontramos no llegan al 0,5, por lo que los valores de las predicciones calculadas no son muy eficaces. Los valores de las métricas del *WiFi* pueden variar debido a la cantidad de datos que existe por posición.

En base a los valores de la desviación típica encontrada en cada uno de los experimentos, podemos observar sus valores en la Tabla 5.8. Cuanto menor sea el valor de la desviación típica de una métrica, significa que su valor varía poco en comparación al valor de la media de esa métrica. En este caso, lo más interesante sería que el valor no sea muy alto, dado que de esa manera puedes asegurar que el modelo sea más estable y indicaría que la variabilidad de los resultados del modelo cada vez que se ejecuta no es alta, por lo que un modelo con una desviación típica baja es generalmente más confiable. Esto significa que siempre se preferirá un modelo que tenga menor desviación típica que otro.

Observando los resultados se puede observar que no son muy buenos, debido a que ni para los valores de la luz ni para los valores del *WiFi* se han conseguido *accuracies* cercanas al 1, ni ninguna otra métrica. Esto es debido a que los valores observados en las figuras del Capítulo 4 son muy parecidos entre sí por cada posición, es decir, no existe una distinción entre los valores de las posiciones para poder determinar de manera precisa a qué posición corresponde cada valor.

Capítulo 6

Conclusión y Líneas Futuras

En esta sección se van a exponer las conclusiones que se han observado en este TFG. También se va a explicar qué limitaciones se han encontrado en el trabajo y qué se podría implementar en el futuro para poder mejorar los resultados y continuar con el desarrollo.

6.1. Conclusiones

En este TFG se ha estudiado cómo obtener el posicionamiento en interiores a través de un terminal móvil, utilizando los valores del RSS de la luz y el *WiFi*, para conocer en qué posición de las 7 posibles se encuentra una persona. El estudio de este trabajo y la forma de predecirlo, se ha hecho mediante la implementación de un algoritmo ML, implementando un clasificador SVM de dos *kernel* distintos tanto para los valores de la luz como los del *WiFi*.

Cuando se han realizado los diversos experimentos se ha observado que para el estudio de la luz se tiene una gran cantidad de datos, por lo que se han podido realizar una cantidad de iteraciones por experimento grande, y hallar la media de los resultados para poder ver el valor de las métricas. Para el estudio del *WiFi* se han utilizado dos frecuencias diferentes y el número de iteraciones por experimento ha sido menor, debido a que la cantidad de datos y valores del RSS por posición era muy inferior en comparación a la luz. Los datos seleccionados tanto para entrenamiento y *test* en el algoritmo se han escogido de manera aleatoria para cada experimento a la hora de obtener los resultados. Como se ha podido observar, en el caso del *WiFi* se obtuvieron mejores resultados que cuando se implementaban los del RSS de la luz, por una diferencia no demasiado grande. Como hemos podido observar en el Capítulo 5, en la Tabla 5.8, se obtienen mejores resultados cuando se implementa en el algoritmo los datos del *WiFi* con la frecuencia de 2462 MHz, y un *kernel polynomial*. También hemos podido observar que cuando tenemos una mayor cantidad de datos y de valores, es una mejor opción para obtener mejores

resultados implementar un *kernel* RBF, como apreciamos en la Tabla 5.7 de los valores de la luz, mientras que si tenemos una menor cantidad de datos y experimentos, es mejor implementar el *kernel polynomial*, como se aprecia también en la Tabla 5.7.

Se obtiene una mejor precisión aplicando el algoritmo para los valores del *WiFi*, pero la diferencia no es muy grande con respecto a los valores de las métricas que se obtienen en el estudio con los datos de la luz. A la hora de implementar el algoritmo para los valores de la luz se obtienen unas métricas que no distan mucho de si entre mejora o no, sin embargo para los valores del *WiFi* pueden variar por cada ejecución, debido a que no hay demasiados valores para realizar un gran número de iteraciones por cada experimento. Tras el análisis realizado en este TFG, se puede afirmar que se pueden mejorar los valores si se implementa el algoritmo utilizando el *WiFi* antes que los de la luz.

6.2. Conocimientos adquiridos por el trabajo

Para la realización de este TFG se han aprendido diversos conocimientos con respecto a temas como el posicionamiento en interiores y la implementación de diversas herramientas, que son:

- Conocimientos sobre la aplicación de algoritmos de clasificación sobre una base de datos para realizar predicciones.
- Expansión de conocimientos sobre el lenguaje de programación *Python*, sobre las clases e realización de gráficas de diversos tipos.
- Aprendizaje sobre el uso de repositorios en *GitHub* y almacenamiento en la nube mediante un control de versiones de actualización del proyecto.
- Para la redacción de la memoria de este TFG se adquirieron conocimientos sobre el lenguaje de *LaTeX*.
- Cómo realizar búsquedas de datos y diversos artículos con palabras clave en *Google Scholar*.
- Técnicas de posicionamiento en interiores utilizando las señales *WiFi* y lumínicas.
- Aprendizaje sobre el procesado de bases de datos y filtrado para su posterior uso.
- Algoritmos de clasificación, utilizando diversos tipos de *kernels* y sus métricas, tanto binarias como multiclase.

6.3. Impacto

En esta sección se explica el impacto que tiene este TFG a nivel social, medioambiental y económico.

6.3.1. Impacto social

Con el desarrollo de este TFG se ha realizado un estudio para predecir la posición de un objeto receptor de señales de luz y *WiFi*, por lo tanto esto se puede aplicar en multitud de aplicaciones para obtener la localización de una persona u objeto en interiores. Esto tiene un gran impacto social, debido a que puede ayudar de diferentes maneras a las personas para hacerles la vida más cómoda y eficiente.

La aplicación de este sistema creado podrían ser por ejemplo ayudar a personas con algún problema de visión reducida o con dificultades de movilidad para encontrar lo que buscan con menos esfuerzo dentro de un edificio y ofrecer una ruta para llegar al destino deseado. Este tipo de tecnología es atractiva también a nivel laboral, dado que se podría llegar a implementar para máquinas automatizadas para que puedan realizar trabajos de traslado utilizando el sistema. También puede servir como sistema de localización para diferentes lugares concretos a los que se necesite guiar a la gente, como en un centro comercial. Otro caso podría ser utilizar el sistema en un *parking* para encontrar diversos vehículos que se necesiten encontrar para el cliente y que no recuerde donde se aparcó, pero que se tenga un receptor que pueda medir las señales.

6.3.2. Impacto medioambiental

Uno de los beneficios de este sistema es que el receptor utilizado es un móvil, por lo tanto no necesitaría de la implementación de ningún sistema externo, debido a que también utiliza la luz existente y el *WiFi* que haya en la zona. Las luces que son necesarias para el algoritmo y el sistema de *WiFi* que haya implementado es el que haya existente en el edificio donde se realice la estimación, por lo que no necesita de ninguna implementación adicional, y el consumo sería el que tuviera habitualmente.

El único consumo podría ser el que se necesitara para realizar la estimación, por lo que sería un consumo de recursos informáticos, que se podría tener en el medioambiente.

6.3.3. Impacto económico

El impacto económico de este trabajo sería muy bajo, debido a que no se requiere una implantación de ningún sistema externo, si no que se utilizaría lo que tuviera el propio

edificio para realizar las tomas de señales requeridas. El consumo de la luz o del *WiFi* sería el que tenga el propio lugar, por lo que no requeriría de ningún coste adicional.

Este TFG se enmarca dentro de los *Objetivos de Desarrollo Sostenible* (ODS), establecidos por la ONU. Estos son 17 objetivos globales adoptados como parte de la Agenda 2030 para el Desarrollo Sostenible. La implantación de ellos fue con el fin de abordar los desafíos más urgentes para promover la igualdad y el cuidado del medioambiente. En la Tabla 6.1 se encuentran los objetivos beneficiados por el estudio del TFG.

Tabla 6.1: Metas específicas de los ODS correspondientes a este TFG.

Objetivo	Meta	Descripción
ODS 7: Garantizar el acceso a una energía asequible, segura, sostenible y moderna	7.1	De aquí a 2030, garantizar el acceso universal a servicios energéticos asequibles, fiables y modernos
	7.3	De aquí a 2030, duplicar la tasa mundial de mejora de la eficiencia energética
	7.b	De aquí a 2030, ampliar la infraestructura y mejorar la tecnología para prestar servicios energéticos modernos y sostenibles para todos en los países en desarrollo, en particular los países menos adelantados, los pequeños Estados insulares en desarrollo y los países en desarrollo sin litoral, en consonancia con sus respectivos programas de apoyo
ODS 8: Promover el crecimiento económico inclusivo y sostenible, el empleo y el trabajo decente para todos	8.2	Lograr niveles más elevados de productividad económica mediante la diversificación, la modernización tecnológica y la innovación, entre otras cosas centrándose en los sectores con gran valor añadido y un uso intensivo de la mano de obra
	8.4	Mejorar progresivamente, de aquí a 2030, la producción y el consumo eficientes de los recursos mundiales y procurar desvincular el crecimiento económico de la degradación del medio ambiente, conforme al Marco Decenal de Programas sobre modalidades de Consumo y Producción Sostenibles, empezando por los países desarrollados

Tabla 6.1: (continuación)

Objetivo	Meta	Descripción
ODS 9: Construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación	9.1	Desarrollar infraestructuras fiables, sostenibles, resilientes y de calidad, incluidas infraestructuras regionales y transfronterizas, para apoyar el desarrollo económico y el bienestar humano, haciendo especial hincapié en el acceso asequible y equitativo para todos
	9.2	Promover una industrialización inclusiva y sostenible y, de aquí a 2030, aumentar significativamente la contribución de la industria al empleo y al producto interno bruto, de acuerdo con las circunstancias nacionales, y duplicar esa contribución en los países menos adelantados
	9.4	De aquí a 2030, modernizar la infraestructura y reconvertir las industrias para que sean sostenibles, utilizando los recursos con mayor eficacia y promoviendo la adopción de tecnologías y procesos industriales limpios y ambientalmente racionales, y logrando que todos los países tomen medidas de acuerdo con sus capacidades respectivas
ODS 11: Lograr que las ciudades sean más inclusivas, seguras, resilientes y sostenibles	11.3	De aquí a 2030, aumentar la urbanización inclusiva y sostenible y la capacidad para la planificación y la gestión participativas, integradas y sostenibles de los asentamientos humanos en todos los países

6.4. Líneas Futuras

En el estudio del posicionamiento en interiores se encuentran muchas posibilidades con diferentes campos y líneas de estudio de investigación y desarrollo. Para la continuación de este estudio realizado, se podría realizar una clasificación con los demás tipos de *kernel* que tiene el SVM, para cotejar resultados y obtener valores distintos o si se podrían mejorar los resultados aplicándolos.

Este estudio se ha realizado teniendo en cuenta 7 posiciones principales del recorrido,

sin tener en cuenta el tipo de iluminación encontrada, por lo que se podría intentar analizar desde donde incide la luz, conocer la localización de los focos y tener en cuenta otros factores externos para analizar cómo varían los resultados, como lo son la luz solar, reflexiones de luz, posibles fuentes externas de luz o puertos *WiFi*, en el caso de la señal *WiFi* incidente.

Otra de las posibilidades utilizando la base de datos existente sería implementar una clasificación multiclase implementando también el SVM, pero teniendo en cuenta tanto los valores del *WiFi* como de la luz al mismo tiempo para predecir la posición del usuario con el terminal móvil.

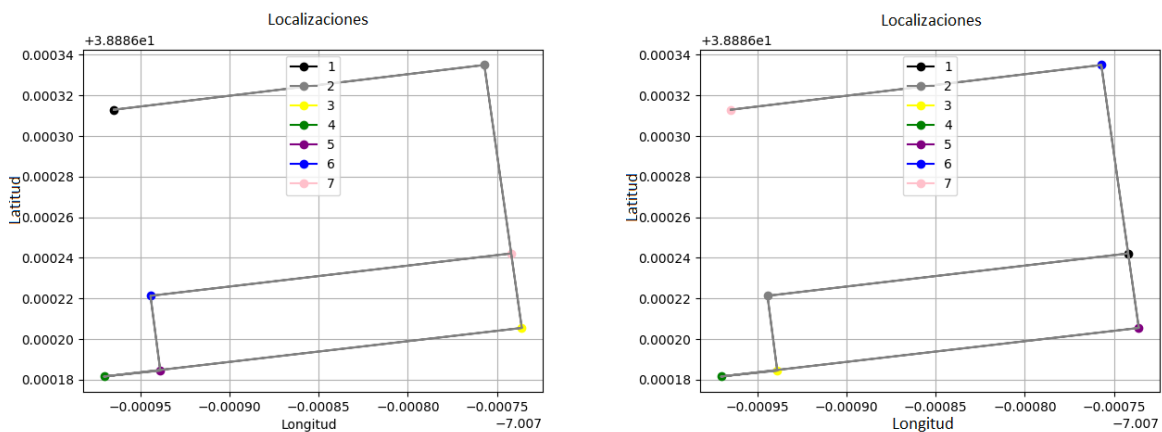
Por último, en este trabajo se han tenido en cuenta únicamente los valores de la luz y del *WiFi*, pero en la base de datos de los ficheros originales, se encuentran medidas sobre otras características de las señales recibidas, como son señales del sonido, la temperatura ambiente o el *Bluetooth*, acelerómetro, el GNSS, magnetómetro, giróscopo, presión, etc, como se observa en la Tabla 4.1, por lo que se podría realizar una clasificación incluyendo estos valores también.

Apéndice A

Figuras adicionales

A.1. Capítulo 4

Las Figuras A.1a A.1b representan las 7 posiciones del recorrido, siendo numeradas e indicando el camino que se realizó en la trayectoria de ida y de vuelta.



(a) Esquema descriptivo de las localizaciones del recorrido en el primer trayecto.

(b) Esquema descriptivo de las localizaciones del recorrido en el segundo trayecto.

Figura A.1: Gráficas descriptivas de la trayectoria del recorrido.

La Figura A.2 es el resultado de normalizar los valores del RSS de la luz a lo largo del tiempo.

En la Figura A.3 se muestran los valores del RSS de la luz en el trayecto, restando el valor de la media aritmética e indicando con una línea gris el valor de la media de cada posición.

Las siguientes Figuras A.4a -A.7b son las correspondientes a los valores del RSS del *WiFi* indicando su frecuencia y su SSID. También se muestran las Figuras de los *box-plos* A.8a A.8b - A.11b de los datos y restando el valor de la media.

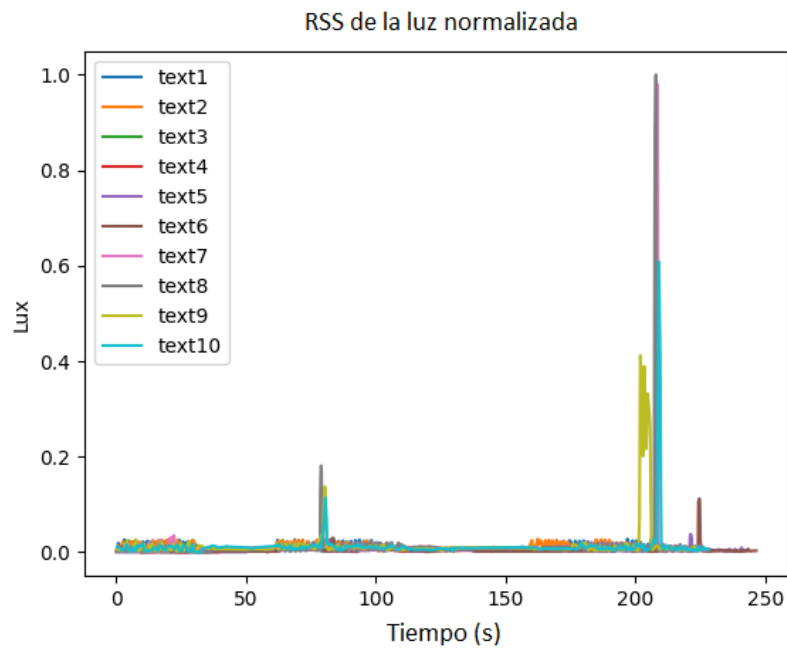


Figura A.2: Intensidad de luz normalizada recibida por el receptor por cada recorrido.

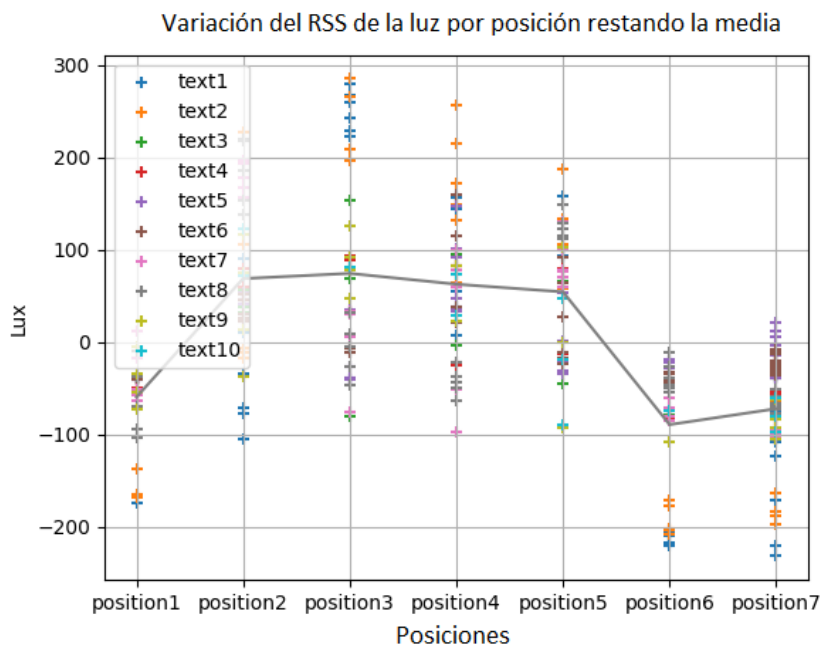
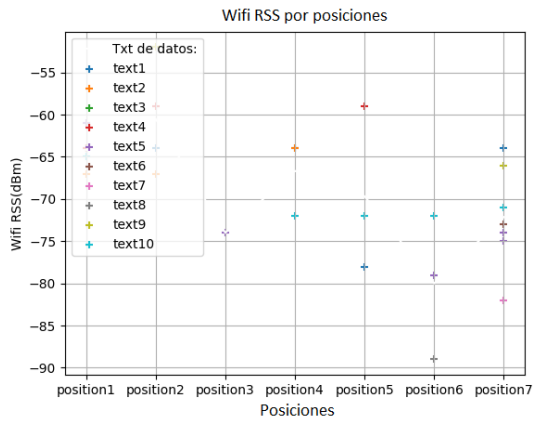
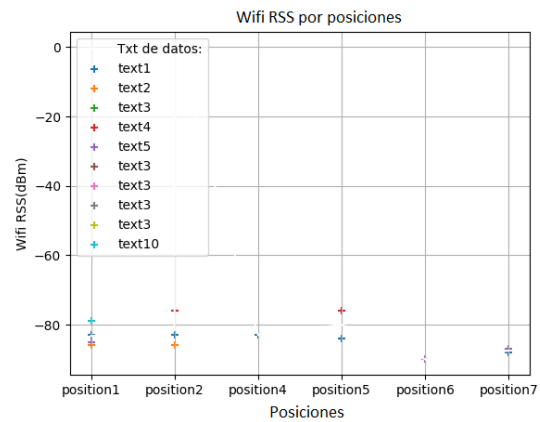


Figura A.3: Intensidad de luz recibida por el receptor por cada posición restando la media de valores por posición.

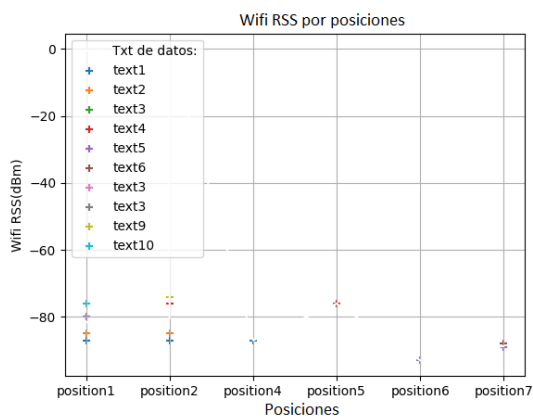


(a) *Received Signal Strength* del *WiFi* por posición y SSID2 con frecuencia 2447 MHz.

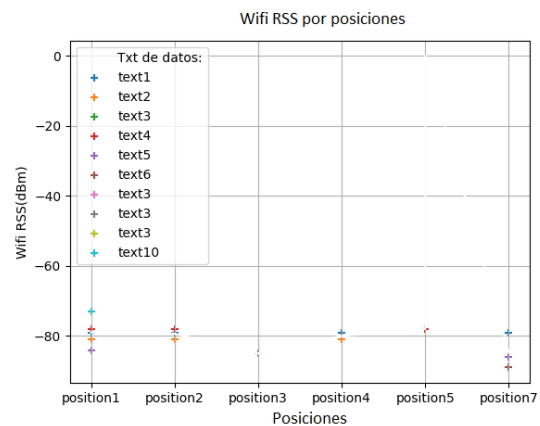


(b) *Received Signal Strength* del *WiFi* por posición y SSID3 con frecuencia 2422 MHz.

Figura A.4: Gráficas *box-plot* correspondientes a los valores del RSS del *WiFi*.

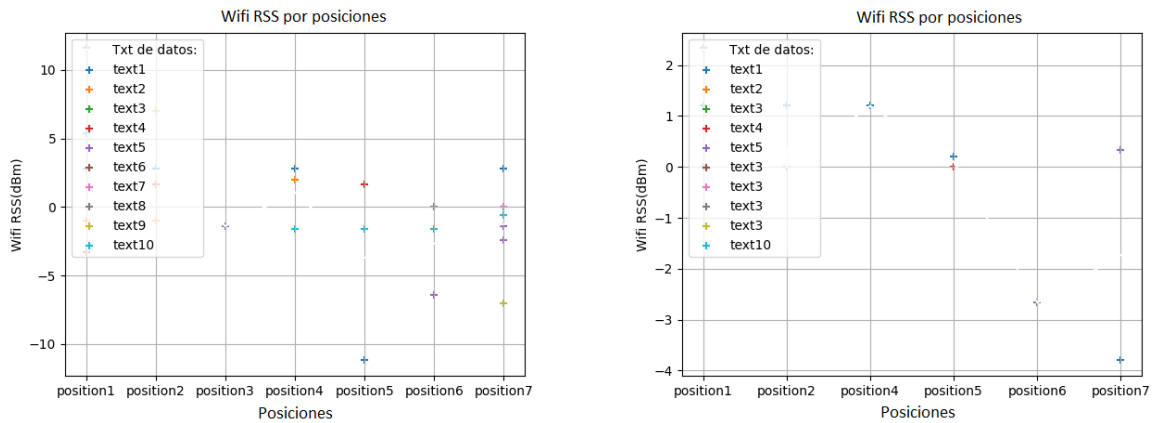


(a) *Received Signal Strength* del *WiFi* por posición y SSID4 con frecuencia 2472 MHz.



(b) *Received Signal Strength* del *WiFi* por posición y SSID5 con frecuencia 2437 MHz.

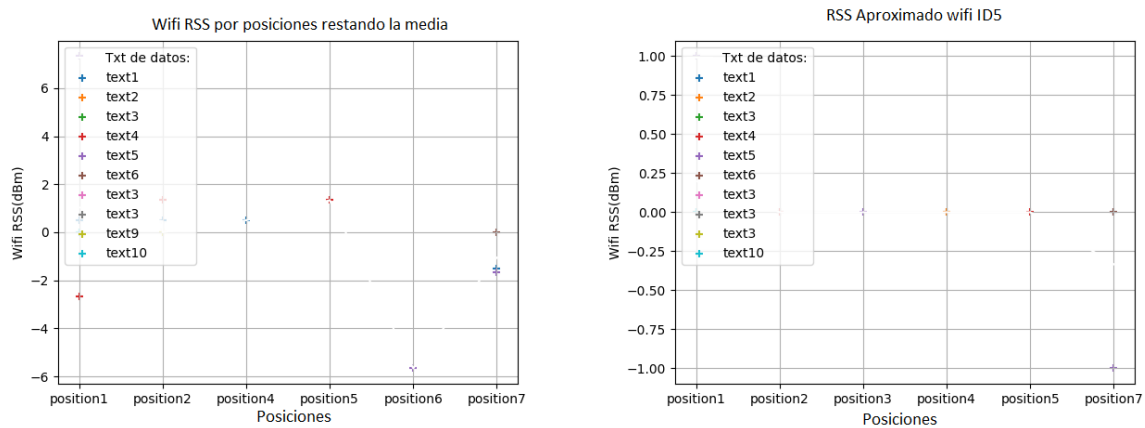
Figura A.5: Gráficas *box-plot* correspondientes a los valores del RSS del *WiFi*.



(a) *Received Signal Strength* del *WiFi* por posición y SSID2 con frecuencia 2447 MHz, restando el valor de su media por posición.

(b) *Received Signal Strength* del *WiFi* por posición y SSID3 con frecuencia 2422 MHz, restando el valor de su media por posición.

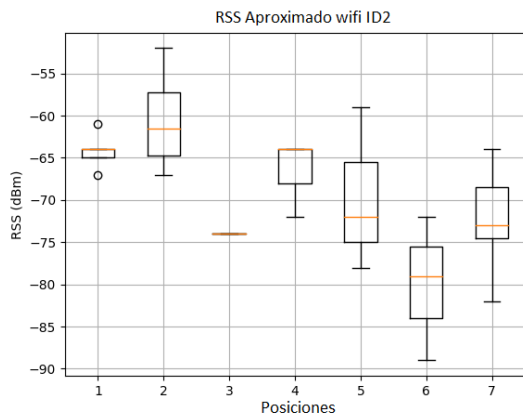
Figura A.6: Gráficas *box-plot* correspondientes a los valores del RSS del *WiFi* restando los valores de las medias.



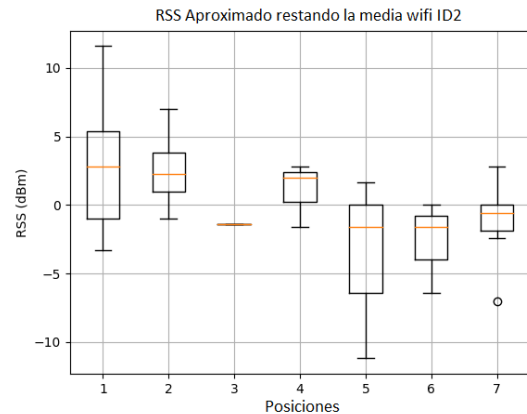
(a) *Received Signal Strength* del *WiFi* por posición y SSID4 con frecuencia 2472 MHz, restando el valor de su media por posición.

(b) *Received Signal Strength* del *WiFi* por posición y SSID5 con frecuencia 5560 MHz, restando el valor de su media por posición.

Figura A.7: Gráficas *box-plot* correspondientes a los valores del RSS del *WiFi* restando los valores de las medias.

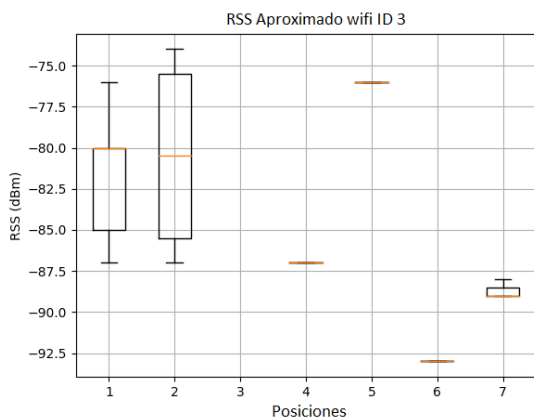


(a) *Box-plot* del *Received Signal Strength* del *WiFi* con SSID2 y con frecuencia 2447 MHz por cada posición del recorrido.

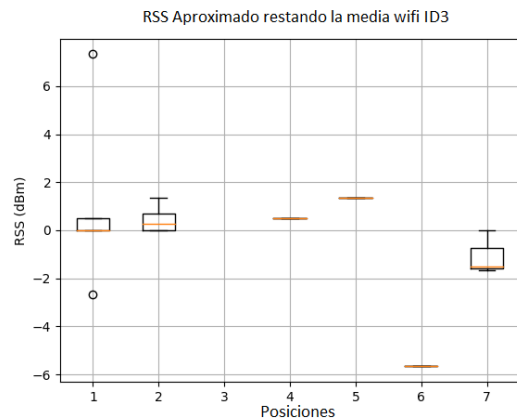


(b) *Box-plot* del *Received Signal Strength* del *WiFi* con SSID2 y con frecuencia 2447 MHz por cada posición del recorrido, restando el valor de la media del RSS de cada posición.

Figura A.8: Gráficas *box-plot* correspondientes a los valores del RSS del *WiFi* con frecuencia 2447 MHz y SSID2.

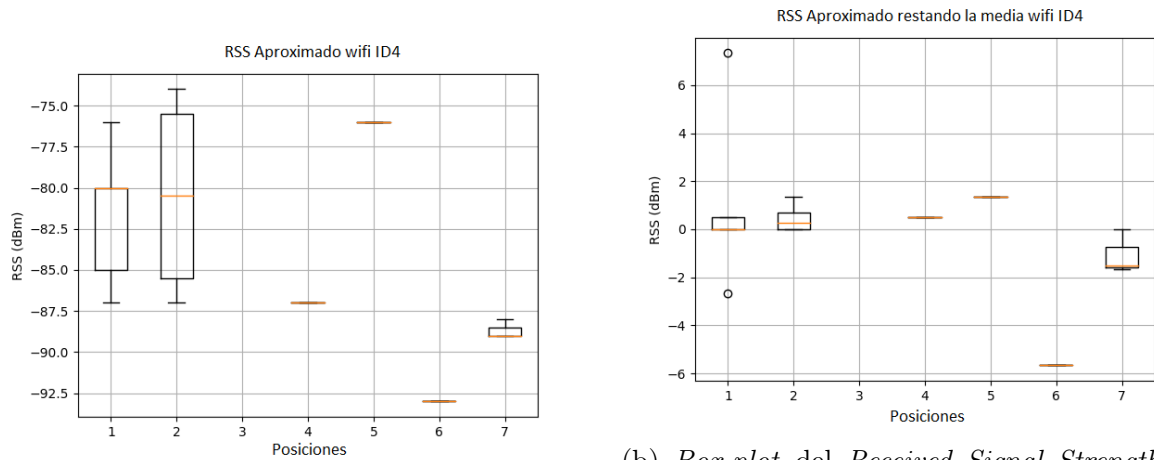


(a) *Box-plot* del *Received Signal Strength* del *WiFi* con SSID3 y con frecuencia 2422 MHz por cada posición del recorrido.



(b) *Box-plot* del *Received Signal Strength* del *WiFi* con SSID3 y con frecuencia 2422 MHz por cada posición del recorrido, restando el valor de la media del RSS de cada posición.

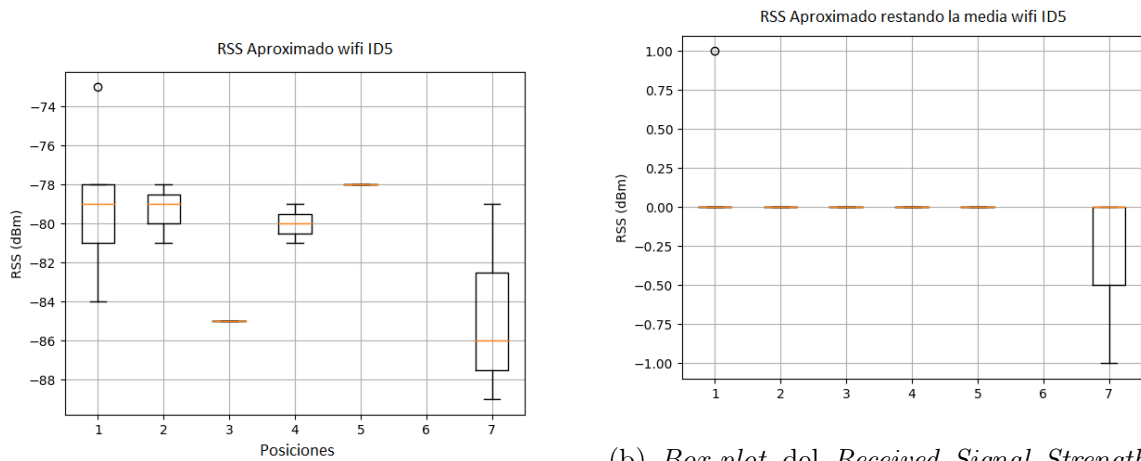
Figura A.9: Gráficas *box-plot* correspondientes a los valores del RSS del *WiFi* con frecuencia 2422 MHz y SSID3.



(a) *Box-plot* del *Received Signal Strength* del *WiFi* con SSID4 y con frecuencia 2472 MHz por cada posición del recorrido.

(b) *Box-plot* del *Received Signal Strength* del *WiFi* con SSID4 y con frecuencia 2472 MHz por cada posición del recorrido, restando el valor de la media del RSS de cada posición.

Figura A.10: Gráficas *box-plot* correspondientes a los valores del RSS del *WiFi* con frecuencia 2472 MHz y SSID4.



(a) *Box-plot* del *Received Signal Strength* del *WiFi* con SSID5 y con frecuencia 2437 MHz por cada posición del recorrido.

(b) *Box-plot* del *Received Signal Strength* del *WiFi* con SSID5 y con frecuencia 2437 MHz por cada posición del recorrido, restando el valor de la media del RSS de cada posición.

Figura A.11: Gráficas *box-plot* correspondientes a los valores del RSS del *WiFi* con frecuencia 2437 MHz y SSID5.

Bibliografía

- [1] Fabiola Mora and Walter Schupnik. El posicionamiento. *La guerra por un lugar en la mente del consumidor*. Lima: Amex, 2001.
- [2] David Andrés Castro Ruiz. Prototipo software de un sistema de posicionamiento en interiores para dispositivos móviles. 2014.
- [3] Davide Micheli, Andrea Delfini, Mario Marchetti, P Gianola, G Bertin, and R Diana. Measurements of the outdoor-to-indoor attenuation of mobile phone signal. *International Journal on Communications Antenna and Propagation (IRECAP)*, 4(6), 2014.
- [4] Elena Aparicio-Esteve, Willem Raes, Nobby Stevens, Jesús UreÁlvaro Hernández. Experimental evaluation of a machine learning-based RSS localization method using Gaussian processes and a quadrant photodiode. *Journal of Lightwave Technology*, 40(19):6388–6396, 2022.
- [5] Jingxue Bi, Meiqi Zhao, Guobiao Yao, Hongji Cao, Yougui Feng, Hu Jiang, and Dashuai Chai. Psovrpos: Wifi indoor positioning using svr optimized by pso. *Expert Systems with Applications*, 222:119778, 2023.
- [6] Alicia Martínez-Rebollar, Luis Antonio López-García, Hugo Estrada-Esquivel, Manuel Mejía Lavalle, and Javier Ortiz Hernández. Navegación en interiores utilizando la tecnología beacons. *Research in Computing Science*, 148:345–355, 2019.
- [7] Goran M Djuknic and Robert E Richton. Geolocation and assisted GPS. *Computer*, 34(2):123–125, 2001.
- [8] Frank Stephen Tromp Van Diggelen. *A-gps: Assisted gps, gnss, and sbas*. Artech house, 2009.
- [9] José Manuel Lanza Gutiérrez. *Metaheurísticas multiobjetivo para la resolución del problema del posicionamiento de nodos repetidores en redes de sensores inalámbricos*. PhD thesis, Universidad de Extremadura, 2015.

-
- [10] Xu Feng, Khuong An Nguyen, and Zhiyuan Luo. A survey of deep learning approaches for wifi-based indoor positioning. *Journal of Information and Telecommunication*, 6(2):163–216, 2022.
- [11] Zhaoyang Han, Ziyue Wang, Huakun Huang, Lingjun Zhao, Chunhua Su, et al. Wifi-based indoor positioning and communication: Empirical model and theoretical analysis. *Wireless Communications and Mobile Computing*, 2022, 2022.
- [12] Pooyan Shams Farahsari, Amirhossein Farahzadi, Javad Rezazadeh, and Alireza Bagheri. A survey on indoor positioning systems for iot-based applications. *IEEE Internet of Things Journal*, 9(10):7680–7699, 2022.
- [13] Huy Q Tran and Cheolkeun Ha. Machine learning in indoor visible light positioning systems: A review. *Neurocomputing*, 491:117–131, 2022.
- [14] Roger Alexander MartJos An indoor visible light positioning system for multi-cell networks. In *Photonics*, volume 9, page 146. MDPI, 2022.
- [15] Xianmeng Meng, Chaochuan Jia, Cuicui Cai, Fugui He, and Qing Wang. Indoor high-precision 3D positioning system based on visible-light communication using improved whale optimization algorithm. In *Photonics*, volume 9, page 93. MDPI, 2022.
- [16] Ren Liu, Zhonghua Liang, Kuo Yang, and Wei Li. Machine learning based visible light indoor positioning with single-led and single rotatable photo detector. *IEEE Photonics Journal*, 14(3):1–11, 2022.
- [17] Shuang Shang and Lixing Wang. Overview of wifi fingerprinting-based indoor positioning. *IET Communications*, 16(7):725–733, 2022.
- [18] Pramila P Shinde and Seema Shah. A review of machine learning and deep learning applications. In *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*, pages 1–6. IEEE, 2018.
- [19] Nitin Kumar Chauhan and Krishna Singh. A review on conventional machine learning vs deep learning. In *2018 International conference on computing, power and communication technologies (GUCON)*, pages 347–352. IEEE, 2018.
- [20] Johnson Kolluri, Dr Shaik Razia, and Soumya Ranjan Nayak. Text classification using machine learning and deep learning models. In *International conference on artificial intelligence in manufacturing & renewable energy (ICAIMRE)*, 2019.
- [21] CARLOS SOTO and CLAUDIA JIMÉNEZ. Aprendizaje supervisado para la discriminación y clasificación difusa. *Dyna*, 78(169):26–33, 2011.

- [22] Jesús Cáceres Tello and Servicios Informáticos. Reconocimiento de patrones y el aprendizaje no supervisado. *Universidad de Alcalá, Madrid*, 2007.
- [23] Yunior César Fonseca-Reyna, Yailen Martínez-Jiménez, and Ann Nowé. Aprendizaje reforzado aplicado a la programación de tareas bajo condiciones reales. *Ingeniería Industrial*, 39(1):36–45, 2018.
- [24] Michael P LaValley. Logistic regression. *Circulation*, 117(18):2395–2399, 2008.
- [25] Ward Hanson and Kipp Martin. Optimizing multinomial logit profit functions. *Management Science*, 42(7):992–1003, 1996.
- [26] Lior Rokach and Oded Maimon. Decision trees. *Data mining and knowledge discovery handbook*, pages 165–192, 2005.
- [27] Barry De Ville. Decision trees. *Wiley Interdisciplinary Reviews: Computational Statistics*, 5(6):448–455, 2013.
- [28] Yanli Liu, Yourong Wang, and Jian Zhang. New machine learning algorithm: Random Forest. In *Information Computing and Applications: Third International Conference, ICICA 2012, Chengde, China, September 14-16, 2012. Proceedings 3*, pages 246–252. Springer, 2012.
- [29] Clifton D Sutton. Classification and regression trees, bagging, and boosting. *Handbook of statistics*, 24:303–329, 2005.
- [30] Alessia Mammone, Marco Turchi, and Nello Cristianini. Support vector machines. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(3):283–289, 2009.
- [31] Joaquin Torres-Sospedra, Fernando Aranda Polo, Felipe Parralejo, Vladimir Bellavista Parent, Fernando Alvarez, Antoni Pérez-Navarro, Antonio R. Jimenez, and Fernando Seco. Datasets and Supporting Materials for the IPIN 2021 Competition Track 3 (Smartphone-based, off- site), June 2022.
- [32] Umit Sen, Yalin Evrim Yesilirmak, Irem Ozgur Bayman, Taner Arsan, Erdal Panayirci, and Nobby Stevens. 3D indoor positioning with spatial modulation for visible light communications. *Optics Communications*, 529:129091, 2023.
- [33] Yu-Chun Wu, Chi-Wai Chow, Yang Liu, Yun-Shen Lin, Chong-You Hong, Dong-Chang Lin, Shao-Hua Song, and Chien-Hung Yeh. Received-Signal-Strength (RSS) Based 3D Visible-Light-Positioning (VLP) System Using Kernel Ridge Regression Machine Learning Algorithm With Sigmoid Function Data Preprocessing Method. *IEEE Access*, 8:214269–214281, 2020.

-
- [34] Esperanza Manrique Rojas. Machine learning: análisis de lenguajes de programación y herramientas para desarrollo. *Revista Ibérica de Sistemas e Tecnologías de Información*, (E28):586–599, 2020.
- [35] Jorge López-Cano Ávila. TFG-2023 Jorge López-Cano Ávila. <https://github.com/JLopezcano/TFG-2023>, 2024. Online; accessed 10 May 2024.

