UNIVERSITAT POLITÈCNICA DE VALÈNCIA

_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de Telecomunicación

Caracterización del canal radio utilizando algoritmos avanzados de inteligencia artificial

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

AUTOR/A: Ardila Abellán, Ernesto

Tutor/a: Rubio Arjona, Lorenzo

CURSO ACADÉMICO: 2023/2024

Resumen

Este Trabajo de Fin de Grado (TFG) se centra en la caracterización y modelado de canales radio en entornos de interiores utilizando técnicas de inteligencia artificial (IA). Se han considerado dos enfoques diferentes: clusterización y clasificación.

El enfoque de clusterización se basa en la identificación de las contribuciones multicamino que llegan al receptor a través de mecanismos de reflexión, difracción y dispersión. Como resultado, se han obtenido los parámetros de un modelo de canal basado en líneas de retardo (TDL, *Tapped Delay Line*) de acuerdo al modelo Saleh-Valenzuela (SV). Los parámetros se han obtenido mediante el algoritmo K-means aplicado a medidas de canal en bandas de milimétricas susceptibles de desplegar las futuras redes inalámbricas 5G y 6G. En el proceso de clusterización se ha tenido en cuenta las características más relevantes de las contribuciones multicamino, como son amplitud, retardo de propagación y dirección de llegada al receptor.

El enfoque de clasificación está orientado a distinguir entre propagación con visión directa (LOS, *Line-Of-Sight*) y sin visión directa (NLOS, *Non-LOS*) basándose en diferentes características del perfil de retardo de potencia (PDP, *Power Delay Profile*). Se ha explorado y analizado una metodología basada en redes neuronales (NN, *Neural Networks*) y máquinas de vectores de soporte (SVM, *Support Vector Machines*) a partir de medidas de canal en dos entornos diferentes. Además, se han propuesto algunas mejoras para tener en cuenta los parámetros más significativos del PDP.

Resum

Este Treball de Fi de Grau (TFG) se centra en la caracterització i modelatge de canals radie en entorns d'interiors utilitzant tècniques d'intel·ligència artificial (IA). S'han considerat dos enfocaments diferents: clusterización i classificació.

L'enfocament de clusterización es basa en la identificació de les contribucions multicamí en el receptor degut a mecanismes de reflexió, difracció i dispersió. Com a resultat, s'han derivat els paràmetres d'un model de canal basat en línies de retard (TDL , *Tapped Delay Line*) basat en el model Saleh-Valenzuela (SV) a partir de mesures de canal realitzades en potencials bandes de freqüència d'ones mil·limètriques per al desplegament de futures xarxes sense fils 5G i 6G. L'algorisme K-means s'ha aplicat a les principals característiques de les components multicamí, com l'amplitud, el retard de propagació i la direcció d'arribada.

La classificació està orientada a distingir entre propagació amb visió directa (LOS, *Line-Of-Sight*) i sense visió directa (NLOS, *Non LOS*) basant-se en diferents característiques del Perfil de Retard de Potència (PDP, *Power Delay Profile*). S'ha explorat i analitzat una metodologia basada en Xarxes Neuronals (NN, *Neural Networks*) i Màquines de Vectors de Suport (SVM, *Support Vector Machines*) a partir de mesures de canal en dos entorns diferents. A més, s'han proposat algunes millores per a tindre en compte els principals paràmetres del PDP.

Abstract

This Final Project Degree is focused on the characterization and modeling of radio channels in indoor environments using artificial intelligence (AI) techniques. Two different approaches have

been considered: clustering and classification.

The clustering approach is based on the identification of the multipath contributions at the receiver due to reflection, diffraction and scattering mechanisms. As a result, the parameters of a Tapped Delay Line (TDL) channel model based on the Saleh-Valenzuela (SV) model have been derived from channel measurements performed at potential millimeter-wave frequency bands for the deployment of the future 5G and 6G wireless networks. The K-means algorithm has been applied from the main characteristics of the multipath components, such as the amplitude, propagation delay and direction of arrival.

The classification approach is oriented to distinguish between Line-of-Sight (LOS) and Non-LOS (NLOS) propagation conditions based on different features of the Power Delay Profile (PDP). A methodology based on Neural Networks (NN) and Support Vector Machines (SVM) has been explored and analyzed from channel measurements in two different environments. Some improvements have been proposed to take into account the main parameters of the PDP.

RESUMEN EJECUTIVO

La memoria del TFG del GTIST debe desarrollar en el texto los siguientes conceptos, debidamente justificados y discutidos, centrados en el ámbito de la IT

CONCEPT (ABET)	**CONCEPTO (traducción)**	**¿Cumple? (S/N)**	**¿Dónde? (páginas)**
1. IDENTIFY:	1. IDENTIFICAR:		
1.1. Problem statement and opportunity	1.1. Planteamiento del problema y oportunidad	S	2
1.2. Constraints (standards, codes, needs, requirements & specifications)	1.2. Toma en consideración de los condicionantes (normas técnicas y regulación, necesidades, requisitos y especificaciones)	S	23-26,27-29, 47-51
1.3. Setting of goals	1.3. Establecimiento de objetivos	S	2-7
2. FORMULATE:	2. FORMULAR:		
2.1. Creative solution generation (analysis)	2.1. Generación de soluciones creativas (análisis)	S	29-36, 44-46, 51-54, 61-65
2.2. Evaluation of multiple solutions and decision-making (synthesis)	2.2. Evaluación de múltiples soluciones y toma de decisiones (síntesis)	S	36-41, 54-61, 66-77
3. SOLVE:	3. RESOLVER:		
3.1. Fulfilment of goals	3.1. Evaluación del cumplimiento de objetivos	S	79-81
3.2. Overall impact and significance (contributions and practical recommendations)	3.2. Evaluación del impacto global y alcance (contribuciones y recomendaciones prácticas)	S	81-83

A mis padres, que siempre han estado presentes en cualquier situación, aportando cariño, confianza y haciéndome creer de lo que soy capaz.
A mi pareja, demostrándome lo que es acompañar para siempre a alguien y apoyándome en todos los ámbitos.
Al resto de mi familia, hermana, sobrinas, abuelos, tíos, que sus palabras de orgullo hacia mí siempre han sido un aliciente para continuar luchando.
Y por último a mi tutor, por la oportunidad y cercanía mostrada a lo largo de los meses.
Gracias a todos, de corazón.

Índice general

Índice de figuras

Índice de tablas

Listado de siglas empleadas

AI Artificial Intelligence. 2

APDP Average Power Delay Profile. 8, 29

CIR Channel Impulse Response. 23

CPI Ciudad Politécnica de la Innovación. 24, 25

ETSIT Escuela Técnica de Ingenieros de Telecomunicación. 27

GRE Grupo de Radiación Electromagnética. 24

IA Inteligencia Artificial. 1, 2, 5, 6, 23, 27, 47, 50, 79–83

IFFT Inverse Fast Fourier Transform. 28

IoT Internet of Things. 47

iTEAM Instituto de Telecomunicaciones y Aplicaciones Multimedia. 5, 8, 24, 26

k-NN K-Nearest Neighbors. 51

LOS Line Of Sight. 2, 3, 5, 8, 9, 11, 25, 30–33, 35, 37–39, 44, 45, 47, 49, 50, 53, 61–63, 66–73, 75, 81

ML Machine Learning. 2, 3, 6, 12–15, 17, 19, 21, 83

MPC Multipath Component. 2, 3, 9, 10, 30, 37, 42, 48, 49, 80, 81

NLOS Non Line Of Sight. 2, 3, 5, 8, 9, 11, 25, 30, 34–37, 39–41, 46, 47, 49, 50, 53, 61–63, 66–73, 75, 80, 81

NLP Natural Language Processing. 21

NN Neural Network. 2, 3, 5, 11, 12, 48, 51, 61–64, 66, 68, 70, 72, 73, 75, 77, 80, 82

ODS Objetivos de Desarrollo Sostenible. 2, 3, 7, 83

ONU Organización de las Naciones Unidas. 79

PDP Power Delay Profile. 1–3, 8, 9, 11, 12, 23, 28, 29, 42, 43, 47–50, 52, 55, 66, 80

QoS Quality Of Service. 1

SAGE Space-Alternating Generalized Expectation-maximization. 1, 8, 9, 11, 23, 29, 32, 33, 35, 36, 38–41, 47

SHAP Shapley Additive Explanations. 66, 67, 69, 71, 73, 80

SMOTE Synthetic Minority Over- sampling Technique. 77

SNR Signal to Noise Ratio. 1

SV Saleh-Valenzuela. 2, 3, 5, 6, 8, 11, 12, 30, 41–46, 79–81

SVM Support Vector Machine. 2, 3, 5, 6, 8, 9, 17, 18, 48, 51–55, 59–61, 66, 80–82

TDL Tapped Delay Line. 2, 3

TFG Trabajo Fin de Grado. 1–3, 5–7, 79, 81–83

TIC Tecnologías de la Información y la Comunicación. 79

ULA Uniform Longitudinal Array. 25

UPV Universidad Politécnica de Valencia. 8, 24–26

URA Uniform Rectangular Array. 25, 29, 33

VNA Vector Network Analyzer. 23, 25

Capítulo 1

Introducción y objetivos

1.1. Introducción

Las telecomunicaciones hoy en día forman parte de la vida cotidiana y cada vez estarán más presentes gracias a los continuos avances desarrollados por la industria [1]. Estos avances, ahora con tecnologías como las comunicaciones espaciales, 5G, 6G, etc., son una continuación de las mejoras que el ser humano ha implementado a lo largo de la historia con un solo fin: mejorar la vida mediante unas comunicaciones más rápidas, eficientes y compatibles con el exponencial crecimiento poblacional y con el uso de datos [2].

Queda muy lejos, por tanto, el hito histórico de *Guglielmo Marconi* de transmitir la primera señal radio, primero a través del canal de la Mancha y años después entre ambos lados del Atlántico [3]. Este hecho fue muy discutido por los científicos de entonces, al no poder asegurar de una forma teórica que lo conseguido por *Marconi* fuese posible. Sin embargo, dos siglos después se conoce por qué se pudo producir, qué ayudó y qué dificultó esa comunicación, y esto en parte es gracias al estudio e investigación del canal radio.

Otro avance que en los últimos tiempos acapara noticias, tertulias e incluso los temas de conversación es la inteligencia artificial (IA). La IA ha irrumpido en todos los marcos posibles: mediático, político, económico, etc., y como todas, tiene sus detractores y sus adoradores; pero lo que está claro es que, con un buen y responsable uso, puede agigantar los pasos en la investigación de muchos campos, entre las que se encuentran las telecomunicaciones. Por ello, en este Trabajo Final de Grado (TFG) se han empleado nuevos algoritmos de IA aplicados al estudio de la caracterización y modelado del canal radio.

En el TFG se parte de unas medidas en frecuencia en un entorno *indoor* a las que se les aplicará herramientas como la transformada rápida inversa de Fourier (IFFT, *Inverse Fast Fourier Transform*) para obtener el perfil potencia-retardo (PDP, *Power Delay Profile*). En el PDP se reflejan las múltiples contribuciones (MPC, *Multipath Component*), consecuencia del efecto multicamino, el cual produce una dispersión temporal que resulta en una posible interferencia entre símbolos y desvanecimientos de la señal recibida. Estos inconvenientes son variantes con el tiempo y deben ser abordados obligatoriamente si se quiere mantener un buen nivel de relación señal a ruido (SNR, *Signal-to-Noise Ratio*) y de calidad del servicio (QoS, *Quality of Service*) en la comunicación.

Además se hace uso del algoritmo SAGE (*Space-Alternating Generalized Expectation-maximization*)

que permite disponer de mayores datos, como los ángulos de llegada de las MPCs. Estos datos contribuirán a la implementación de algoritmos de IA para las dos vías de actuación en las que se divide el proyecto: la clusterización y la clasificación.

1.2. Planteamiento y oportunidad

En el contexto actual, las telecomunicaciones juegan un papel fundamental en el desarrollo y avance de la sociedad moderna. La creciente demanda de conectividad y el auge de nuevas tecnologías, requieren una caracterización precisa y eficiente del canal radio. En este TFG se aborda esta necesidad mediante la utilización de técnicas avanzadas de IA para la caracterización y clasificación de señales en entornos *indoor*.

El uso de algoritmos de aprendizaje automático (ML, *Machine Learning*), como K-means, redes neuronales (NN, *Neural Networks*) y máquinas de soporte vectorial (SVM, *Support Vector Machines*), ofrece una ventaja significativa para mejorar la comprensión del comportamiento del canal radio. La caracterización precisa de estos canales no solo permitirá optimizar la transmisión de datos, sino que también contribuirá a la creación de modelos estándar que puedan ser aplicados en diferentes entornos y tecnologías.

El proyecto se divide en dos enfoques principales: la clusterización y la clasificación. La clusterización, mediante el uso del algoritmo K-means, agrupa las MPCs recibidas basándose en parámetros como potencia, retardo y ángulos de llegada. Esto no solo permite entender mejor el entorno de propagación, sino que también facilita la creación de un modelo Saleh-Valenzuela (SV) estándar. Por otro lado, en la clasificación se utiliza SVM y NN para categorizar diferentes situaciones de visibilidad entre antenas, como paso a la optimización en la recepción de señales.

Por tanto, en este TFG se discute cómo aplicar técnicas de IA en un campo crucial para el desarrollo de las telecomunicaciones. La implementación y comparación de diferentes algoritmos no solo aportará un avance académico y científico, sino que también tendrá un impacto directo en la industria, mejorando la calidad y eficiencia de las comunicaciones inalámbricas.

Además, este proyecto se alinea con varios Objetivos de Desarrollo Sostenible (ODS), tales como la Industria, Innovación e Infraestructura y Acción por el Clima, que se explican en el Capítulo 7. Al mejorar la eficiencia energética de las redes de telecomunicación y optimizar el uso de recursos, este trabajo no solo avanza en el campo técnico, sino que también contribuye al bienestar y sostenibilidad global.

1.3. Objetivos

Los objetivos de este TFG están orientados a la caracterización del canal radio a través de las dos vías previamente comentadas: la clasificación y la clusterización.

Con la clusterización, se empezará distinguiendo qué parámetros (potencia, retardo, acimut y elevación) son óptimos combinar para agrupar las MPCs, y tras ello, obtener los parámetros de un modelo basado en el modelo SV, con el que estandarizar el comportamiento del canal en entornos similares. Esto permitirá exportar las conclusiones del proyecto a otros ámbitos en los que también sea necesario analizar el impacto del canal, pudiendo suponer un entorno similar al de este trabajo

y conllevando un ahorro temporal y de recursos importante.

El segundo objetivo será el de la implementación de dos algoritmos que permitan distinguir entre situaciones de visibilidad radioeléctrica (LOS, *Line-Of-Sight*) o sin visibilidad (NLOS, *Non LOS*). Por ende se estudiará qué parámetros del PDP son los más influyentes en las decisiones, además de comparar el funcionamiento de ambas tecnologías, enfrentando resultados y analizando los puntos clave de cada método. Todo ello con herramientas como la matriz de confusión [4], diagrama de pesos, etc.

Después del desarrollo de ambas vías, se tendrá la capacidad de modelar el entorno, lo que facilitará la tarea de mitigar las consecuencias de la emisión de señales en el canal radio y permitirá ahorrar recursos en futuros proyectos.

1.4. Distribución de la memoria

La memoria está estructurada en 7 capítulos:

- **Capítulo 2: Metodología**
  - Se describe la distribución de la memoria y la organización de las tareas realizadas durante el TFG, detallando los procedimientos y métodos utilizados.
- **Capítulo 3: Aspectos técnicos**
  - Se presentan las principales características del canal radio, el modelo SV y los algoritmos de ML empleados: K-means, SVM y NN.
- **Capítulo 4: Descripción de medidas**
  - Se presentan las medidas en el canal radio utilizadas en el TFG, describiendo sus características y el contexto en el que se realizaron.
- **Capítulo 5: Clusterización**
  - Se explica el procesamiento seguido en el tratamiento de las medidas y la implementación del algoritmo K-means para agrupar las MPCs y obtener los parámetros del modelo SV.
- **Capítulo 6: Clasificación**
  - Se detalla el uso de algoritmos de clasificación, como SVM y NN, incluyendo el ajuste de hiperparámetros y la evaluación de resultados.
- **Capítulo 7: Conclusiones, trabajo futuro y contribuciones del proyecto**
  - Se resumen los resultados obtenidos, las principales contribuciones del TFG y se proponen líneas futuras de investigación. A su vez, se hace referencia a los ODS relacionados con el TFG.

Capítulo 2

Metodología

Este segundo punto de la memoria tiene como objetivo mostrar el trabajo realizado, siguiendo una cronología veraz que permita reflejar el trabajo hecho a lo largo del TFG. También se detallarán las tareas, procedimientos y métodos utilizados con el fin de conseguir los objetivos descritos previamente.

2.1. Gestión del proyecto

La memoria está compuesta por 7 capítulos distribuidos por sus propios objetivos, asegurando así la correcta comprensión por parte del lector.

Si se analiza la trama lógica de cualquier proyecto, redacción, artículo, etc., se reconocen 3 partes principales: introducción, cuerpo argumentativo y conclusión. Es por ello por lo que la memoria seguirá este mismo esquema, en el que se comenzará introduciendo el ámbito teórico del proyecto para propiciar el conocimiento de los aspectos más técnicos, detallando las características, métricas y sus modelos. Posteriormente se describirán las medidas, aclarando puntos como el entorno y la información que proporcionarán.

La parte central del TFG se centrará en aplicar el algoritmo de clusterización a los datos recopilados durante las campañas de mediciones, tanto en el laboratorio como en la sala de becarios del Instituto de Telecomunicaciones y Aplicaciones Multimedia (iTEAM). Se comenzará utilizando el algoritmo K-means, combinando retardo y amplitud junto con los ángulos de llegada, que añaden información esencial para obtener los parámetros de modelos de canal como el modelo SV.

Además, en esta fase, se utilizarán algoritmos de IA para la clasificación como SVM y NN. Se probarán diferentes configuraciones de SVM, ajustando parámetros como el valor de C y los *kernels* utilizados, para optimizar la clasificación de las medidas obtenidas. Respecto a las NN, se diseñarán y ajustarán modelos cambiando hiperparámetros como el número de neuronas en la capa oculta y la tasa de aprendizaje, evaluando su desempeño en la clasificación de las muestras LOS y NLOS.

Como conclusión, se resumirá el análisis realizado a la vez que se proponen mejoras futuras en pro de la eficiencia de los diseños y determinando las mejores opciones para alcanzar los objetivos propuestos.

2.2. Organización de tareas

En esta sección se presenta la planificación realizada durante el TFG, detallándose en la Figura 2.1. Cada tarea, numerada del 1 al 11, se distribuye a lo largo de los meses de noviembre a junio, destacando el periodo de ejecución de cada una y proyectando el progreso del trabajo realizado, facilitando así el seguimiento de las actividades y sus plazos.

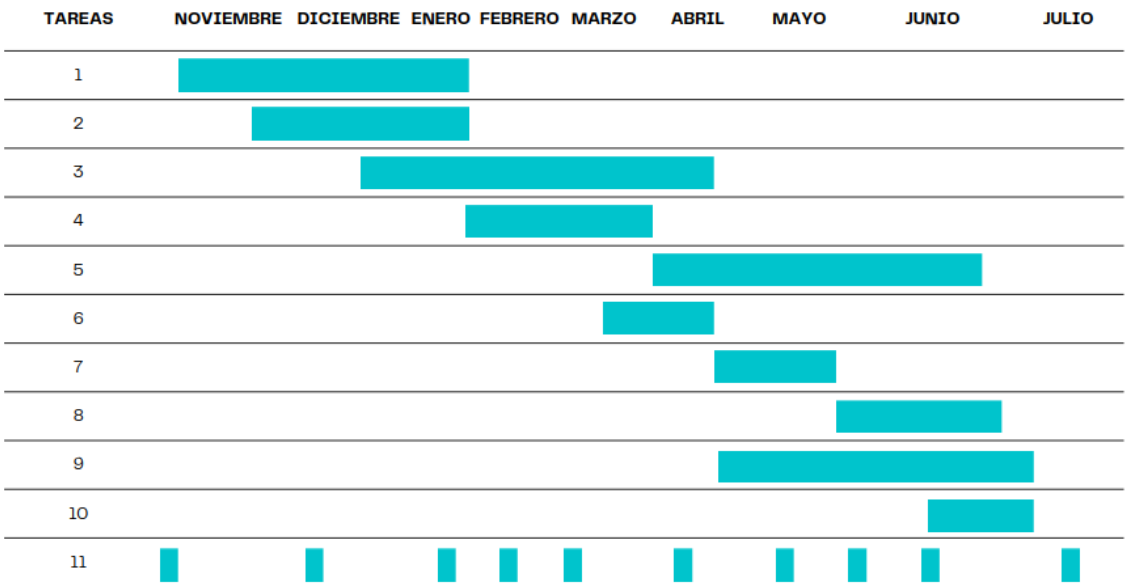

Figura 2.1: Tareas realizadas durante el proyecto.

A continuación, se explicará en detalle cada tarea, especificando el alcance y garantizando una comprensión exhaustiva de su contribución al desarrollo global del proyecto.

1. Documentación acerca del canal radio y sus líneas de estudio.
2. Programación del procesamiento de las medidas del canal radio.
3. Documentación sobre algoritmos de inteligencia artificial.
4. Programación del procesamiento de las medidas, adaptando el *dataframe* para los algoritmos sobre clasificación.
5. Pruebas y evaluación de resultados recogidos por los algoritmos de IA aplicados a la clasificación.
6. Programación del algoritmo de K-means.
7. Pruebas y evaluación de resultados recogidos por el algoritmo de IA aplicado a la clusterización.
8. Extracción de los parámetros del modelo SV.
9. Escritura de la memoria.

10. Preparación de la presentación para la defensa.
11. Tutorías personalizadas.

Capítulo 3

Aspectos técnicos

3.1. Canal radio

El canal radio se define como el medio físico a través del cual un transmisor y un receptor establecen una comunicación mediante ondas de radio [5]. Sin embargo, esta transmisión se verá afectada por las consecuencias de propagar una señal por el canal, destacando así la importancia de su estudio en las telecomunicaciones.

Es vital conocer los cambios producidos en la señal transmitida y contrarrestarlos para recibir así la señal más pura posible. Todo esto pese a las cualidades variantes en el tiempo del canal, lo que convierte su estudio en una difícil tarea.

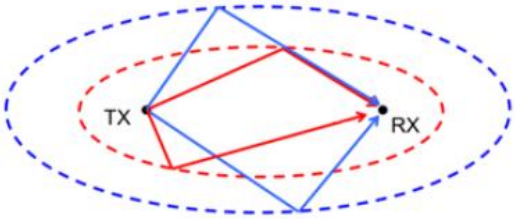

Figura 3.1: Reflexiones transmisor-receptor [5].

En la Figura 3.1 se esquematiza un ejemplo de transmisión por un medio como el aire y en un entorno interior o *indoor*. Con este entorno la señal total recibida estará formada por un conjunto de las contribuciones directa, reflejadas, difractadas, etc., lo que se denominará como contribuciones multicamino o MPCs.

Para su caracterización se definirá al canal temporal como $h(t, \tau)$. Suponiendo la señal transmitida como $z(t)$, entonces la señal recibida quedaría representada según la ecuación (3.1) [5]:

$$w(t) = \int_{-\infty}^{\infty} z(t-\tau)h(t,\tau)\,d\tau, \tag{3.1}$$

donde τ es el retardo con el que se reciben las contribuciones, $z(t)$ es la señal retardada, y $h(t, \tau)$ es el efecto del canal afectándole a la señal.

Este efecto de la propagación de las ondas, mostrado en la Figura 3.1, puede suponer:

1. **Interferencia entre símbolos**
   Las muestras reflejadas o difractadas de la señal original tendrán un retardo diferente y la suma del total de esa dispersión, en ocasiones puede ser mayor al periodo de transmisión de señales. Esto conllevaría que en el receptor se combinaran contribuciones de pulsos diferentes, tal y como se muestra en la Figura 3.2.

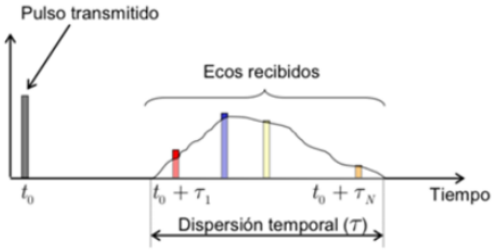

Figura 3.2: Pulsos de ecos [5].

   Una solución, si se mantienen los mismos elementos de medición, sería reducir la velocidad de transmisión, medida en bps. Como ejemplo estaría la utilización de una modulación menos eficiente, pero a su vez más robusta. La teoría de la eficiencia espectral de las diferentes modulaciones se evidencia en el Teorema de Shannon, mostrado en la Figura 3.3.

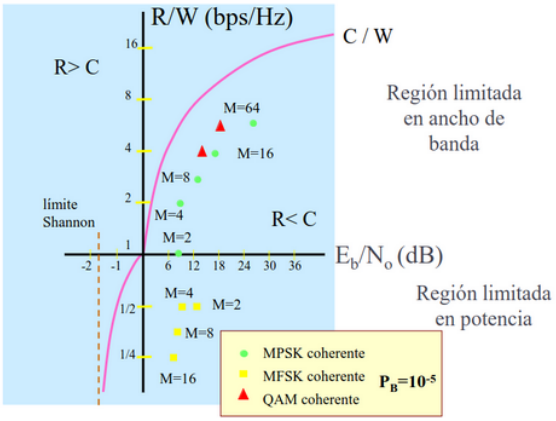

Figura 3.3: Límite de Shannon [6].

2. **Desvanecimientos**
   Atendiendo a las fases con las que llegan las MPCs, se identificarán sumas en fase o contrafase, lo que podrá desencadenar en desvanecimientos importantes de la potencia. Para ello se proponen soluciones como la diversidad o de nuevo el ajuste de modulación, entre otras.

3. **Atenuación**
   Además de recorrer un mayor espacio, lo que conlleva ya de por sí mayores pérdidas, la reflexión o difracción de una señal con elementos urbanos, naturales, etc., reduce la potencia de la onda, por lo que un aumento de la potencia emitida puede ser necesario en ciertos casos.

4. **Ruido / Interferencias**
   Las muestras residuales del multicamino podrán ser tratadas o rechazadas con filtros, facilitando la correlación final.

Con el objetivo de plasmar cómo afecta el efecto multicamino sobre las medidas, se muestra una imagen clara a través del gráfico denominado PDP, mostrado en la Figura 3.4 con valores lineales y en la Figura 3.5 en cifras logarítmicas, representando ambos la distribución temporal de la energía. Es decir, cómo se recibió la señal directa y sus ecos con sus retardos correspondientes.

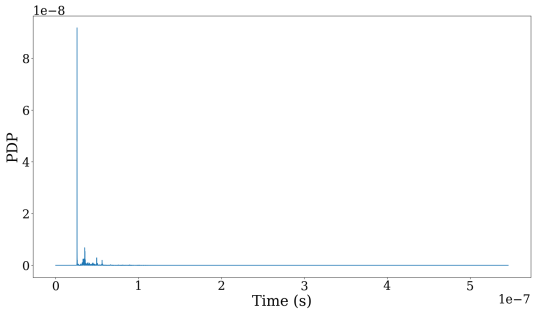

Figura 3.4: PDP en unidades lineales.

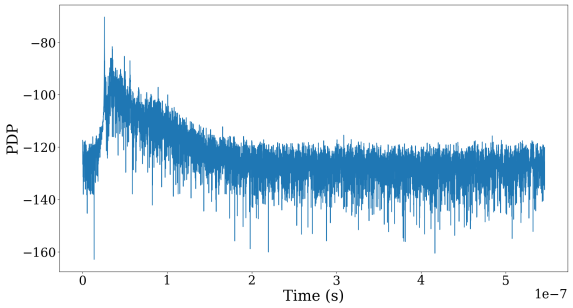

Figura 3.5: PDP en unidades logarítmicas.

De cara a describir el entorno y caracterizar las medidas del canal radio, el PDP limitaría la información simplemente a potencia y retardo, relegando del análisis datos como los ángulos de llegada de las contribuciones. No obstante, herramientas como SAGE permiten obtener la estimación con una gran precisión del canal mediante potencia, retardo, elevación y acimut.

3.2. Modelo Saleh-Valenzuela

En el ámbito de las telecomunicaciones, este método de análisis es uno de los más extendidos al permitir, con gran rigor, la modelización del canal radio tanto en entornos *indoor* como *outdoor*, a través de las distribuciones temporales y de amplitud que siguen los clústeres de cada PDP [7].

Como se explicará en lineas posteriores, las contribuciones recibidas y reflejadas en el PDP se agruparán atendiendo a ciertas características mediante las que se definirá el comportamiento del canal.

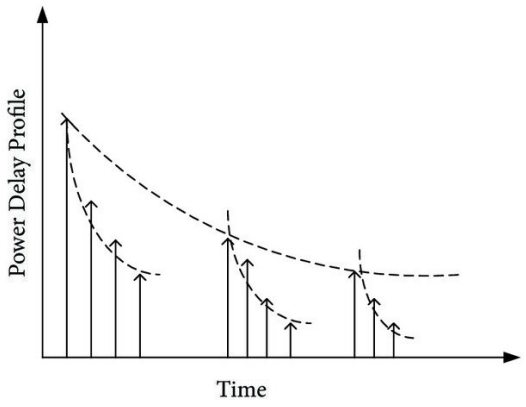

Figura 3.6: PDP en unidades lineales según el modelo SV [8].

En la Figura 3.6 se pueden plasmar los parámetros propios del algoritmo SV. De ellos, λ y Λ hacen referencia al retardo, cuyo conjunto de datos sigue una distribución de *Poisson*; los otros dos, γ y Γ, se refieren a la potencia y su decaimiento con el tiempo.

1. λ **:** Informa del ratio medio de tiempo entre muestras pertenecientes al mismo clúster. Esto es el retardo de propagación medio.
2. Λ **:** Representa la media de los retardos totales de cada clúster, desde su primera contribución hasta la última. Físicamente se le denomina retardo de los clústeres
3. γ **:** Siguiendo esta distribución se obtendrá el factor por las que las contribuciones de un mismo clúster decaen en potencia con el retardo.
4. Γ **:** Mismo significado físico que el anterior, pero en este caso contando con las primeras contribuciones de cada clúster del PDP, resultando en el decaimiento medio.

3.3. Algoritmos de Machine Learning (ML)

3.3.1. K-means

K-means es uno de los algoritmos de aprendizaje automático más célebres para la clasificación de datos [9]. Distinguiendo entre sus diferentes tipos, K-means se sitúa en el grupo de mecanismos basados en aprendizaje no supervisado, al no necesitar que los datos que se someten a entrenamiento estén previamente categorizados, sino que el propio modelo es capaz de interpretar las relaciones existentes entre el conjunto de datos y a partir de ahí, los clasifica.

En cuanto al funcionamiento del algoritmo K-means, se podrían destacar 3 pasos principales, de los cuales 2, más concretamente los pasos 2 y 3, se repetirán en bucle hasta alcanzar la optimización del problema, basada en una minimización de las distancias entre los datos.

1. **Inicialización**
   El modelo asigna, dentro del rango de los datos, tantos centroides como número de clústeres

se hayan definido previamente. Esta inicialización tiene varias formas de asignación de los centroides dependiendo del modelo de K-means que se esté utilizando, ya que con los años se han desarrollado modelos más complejos en los que ya no se realiza este proceso de manera aleatoria, sino que se puede establecer su ubicación tanto manualmente como pedirle al algoritmo que previamente encuentre las mejores coordenadas para situarlos, como es el modelo K-means++ [10]. En este proyecto se ha decidido utilizar la variante estándar, la cual está representada su asignación inicial de centroides en la Figura 3.7.

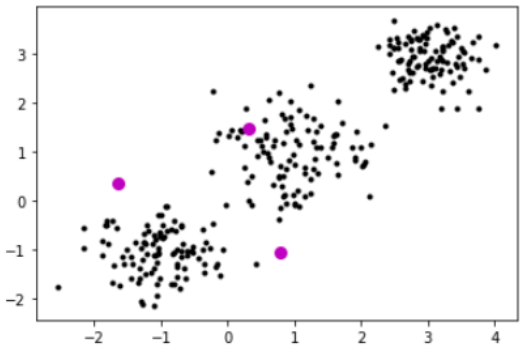

Figura 3.7: Inicialización de centroides en algoritmo K-means [11].

2. **Agrupación por centroides**
   Una vez los centroides se asignan y tienen definidos sus coordenadas en el mapa de datos, el algoritmo realiza las operaciones matemáticas para calcular las distancias euclídeas entre cada punto y los centroides, agrupando estos datos con los clústeres cuya distancia sea menor, tal y como se muestra en la Figura 3.8:

$$\min_{S} E(\mu_i) = \min_{S} \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2. \tag{3.2}$$

En la ecuación (3.2) se representa la fórmula que sigue K-means para la optimización, en la que se dispondría del conjunto de datos proveniente de las medidas (x_1, x_2, x_n) a los que se les calcula las distancias respecto de su centroide asignado μ_i minimizándolas y creando los clústeres $S = \{S_1, S_2, S_k\}$ [11].

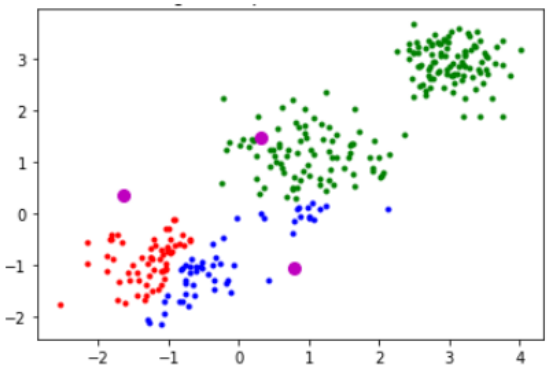

Figura 3.8: Agrupación de datos en algoritmo K-means [11].

3. **Corrección centroides**
   La primera posición asignada a los centroides no es la definitiva, sino que tras ella y después de la primera agrupación, se determinará como nuevo centroide el promedio de los datos de cada clúster, intentando reducir al máximo posible el error de asignación. Este último paso queda representado en la Figura 3.9.

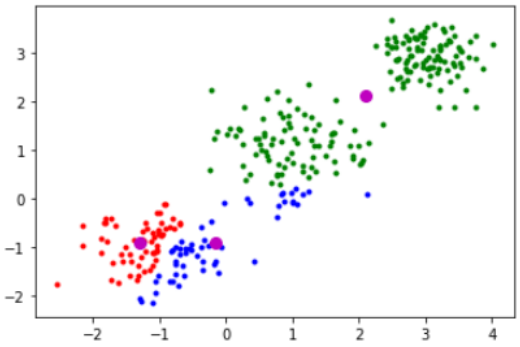

Figura 3.9: Optimización centroides en algoritmo K-means [11].

Para poder explicar en detalle los pasos que K-means sigue, primero se ha supuesto conocer el número de clústeres óptimos, denominado como el número k, para el conjunto de datos, pero tan importante es diseñar y escoger bien el algoritmo de ML a utilizar, en este caso con K-means, como escoger k correctamente. Para determinar este valor, existen varias métricas o métodos que ayudan, aunque siempre será necesaria la visión crítica del investigador debido a que estos algoritmos basados en términos matemáticos pueden obviar relaciones, características, etc., importantes al clasificar los datos y que pueden no quedar correctamente reflejados en los clústeres.

Las métricas que se han utilizado en este proyecto son las siguientes:

1. **Método del codo**
   El método del codo consiste en la evaluación de la inercia, siendo esta la suma de todas las distancias entre el centroide y los puntos de su mismo clúster, todas ellas elevadas al cuadrado, tal y como se muestra en la ecuación (3.3) [12].

$$WSS = \sum_{i=1}^{n_k} \|X_{ik} - C_k\|^2, \tag{3.3}$$

donde:

- WSS representa la suma de las distancias cuadradas dentro de los clústeres.
- X_{ik} es el vector de datos i-ésimo en el clúster k.
- C_k es el centroide del clúster k.
- n_k es el número de datos en el clúster k.

Sería trivial reparar en que, a mayor número de clústeres, se obtendría un valor de inercia menor. Por ello, si se buscase una reducción drástica y continua de la inercia, bastaría con aumentar k; pero el objetivo es encontrar un número óptimo de clústeres con los que se pueda definir objetivamente el entorno de las mediciones, simplemente observando los datos. Un ejemplo de la gráfica obtenida en el método del codo queda representada en la Figura 3.10.

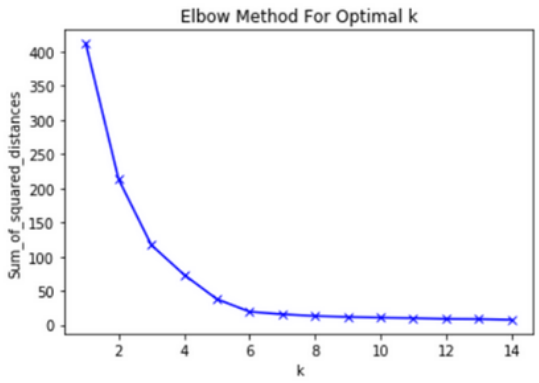

Figura 3.10: Método del codo [13].

Para averiguar el k óptimo, se analizará la gráfica con este método, que define la zona de inflexión como la más probable para encontrarlo, a modo de primera estimación. La lógica que sigue es tomar como aceptable el aumento de clústeres mientras que la pendiente de la gráfica produzca cambios significativos; a partir de entonces, supondría una cuestión de ineficiencia el aumentar el número de grupos al no tener mejoras en la inercia.

2. **Índice de Calinski-Harabasz**
   Este índice, representado en la Figura 3.11, permitirá evaluar la calidad de los clústeres, comparando la dispersión de los datos dentro de un mismo clúster (distancias intra-clúster) y los demás clústeres (inter-clúster). Con un valor alto de Calinski-Harabasz, se podría hablar de una agrupación densa dentro de la misma y separada de las demás.

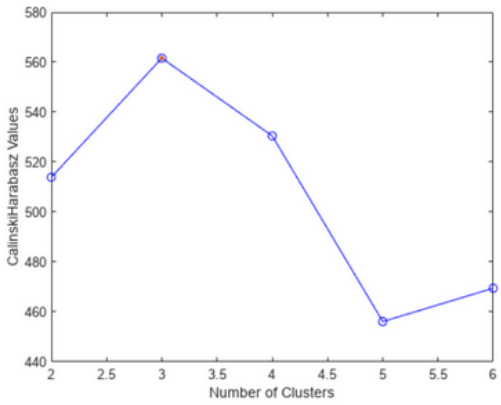

Figura 3.11: Métrica de calidad del índice Calinski-Harabasz [14].

A continuación en la ecuación (3.4) [12], se definirá el índice Calinski-Harabasz como:

$$CH = \frac{\frac{BSS}{K-1}}{\frac{WSS}{N-L}}, \tag{3.4}$$

donde:

- *BSS* representa las distancias entre clústeres,con factores multiplicando como el número de clústeres k.
- N es el número total de datos.

3. **Índice de Silhouette**
   Con esta última métrica, cuyos valores oscilan entre 1 y –1, se conseguirá plasmar en una gráfica cómo de parecidos son los datos de un mismo clúster enfrente de los de otros clústeres. El número de clústeres que aporte valores mayores que 0.5 en el índice significarían una muy buena opción a determinarse como k. Un ejemplo del índice de Silhouette se representa en la Figura 3.12.

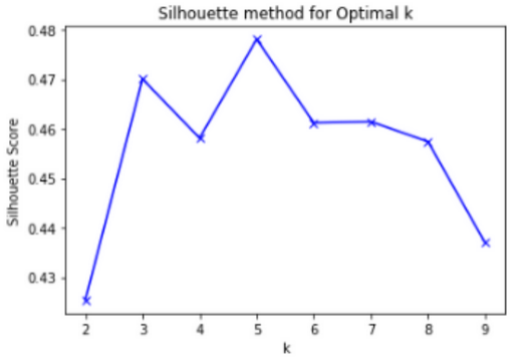

Figura 3.12: Métrica de calidad del índice Silhouette [13].

Matemáticamente se podría deducir el índice de Silhouette a través de la ecuación (3.5).

$$\text{Coeficiente Silhouette} = \frac{(x-y)}{\text{máx}(x,y)}, \tag{3.5}$$

donde:

- y es la media de la distancia intra-clúster.
- x es la media de la distancia del clúster más cercano.

3.3.2. Máquina de Soporte Vectorial (SVM)

SVM es un algoritmo, esta vez de aprendizaje supervisado, utilizado mayormente en estudios de clasificación o regresión.

El método de funcionamiento de SVM consiste en establecer en el rango de datos de entrada un hiperplano de separación que permita distinguir entre las diferentes clasificaciones que se le pida, como se muestra en la Figura 3.13.

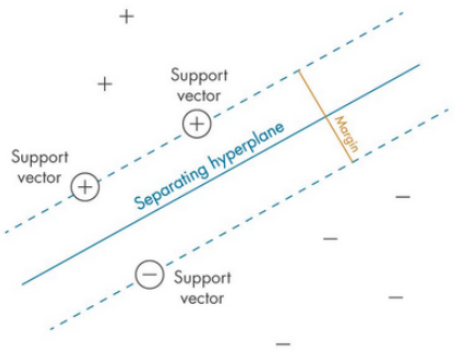

Figura 3.13: Frontera de vectores de soporte en algoritmo SVM [15].

En la Figura 3.13 se muestra una simple representación de cómo trabajaría este algoritmo para marcar una frontera entre los datos. Entre ambas clases se dejaría un espacio de margen en el que se permite un máximo número de muestras dentro de él y que estaría marcado por los vectores de soporte, siendo estos los datos que estén posicionados en la línea separadora del propio margen. El objetivo de este hiperplano es separar al 100 % de precisión los datos, pero también conseguirlo de una manera óptima, para lo que sería necesario conseguir el margen más amplio posible. Esto aseguraría un mejor funcionamiento del algoritmo a la hora del testear con nuevos datos desconocidos hasta el momento.

En el mundo del aprendizaje automático, no se pueden generalizar las soluciones debido a la cantidad de variantes distintas en cada problema o proyecto, que es infinita. SVM, gracias a la versatilidad en su funcionamiento, dispone de un parámetro, conocido como C, que permite regular la aceptación de fallos con la anchura del margen, pudiendo establecer un margen muy grande con admitancia a fallos, o uno muy pequeño sin tanta tolerancia.

Dentro de SVM además se distinguirán varios tipos, según cómo se procesen los datos a través de los diferentes *kernels* disponibles. Estos son funciones, como las representadas en las ecuaciones (3.6), (3.7), (3.8) y (3.9), que manejan los datos de entrada del algoritmo para hacerlo más sencillo a la hora de tratarlos, consiguiendo así una mayor eficacia. Por ello, cada tipo dará una máquina de vectores diferente:

1. Lineal
$$K(x_1, x_2) = x_1^\top x_2. \tag{3.6}$$
2. Polinómica
$$K(x_1, x_2) = (x_1^\top x_2 + 1)^\rho. \tag{3.7}$$
3. Sigmoide
$$K(x_1, x_2) = \tanh(\beta_0 x_1^\top x_2 + \beta_1). \tag{3.8}$$
4. *RBF* o gaussiana
$$K(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\right). \tag{3.9}$$

donde:

- $K(x_1, x_2)$ es la función *kernel* que mide la similitud entre los vectores x_1 y x_2.
- x_1 y x_2 son los vectores de características.
- $x_1^T x_2$ es el producto interno de x_1 y x_2.
- ρ es el grado del polinomio en el *kernel* polinómico.
- β_0 y β_1 son parámetros del *kernel* sigmoide.
- $\|x_1 - x_2\|^2$ es la distancia euclidiana al cuadrado entre los vectores x_1 y x_2.
- σ es el parámetro del *kernel RBF* que controla el ancho de la función gaussiana.

Estos *kernels* [15] o funciones, permiten que relaciones entre los datos difíciles de ver a simple vista puedan ser observables y separadas por el hiperplano, por ejemplo, reduciendo una clasificación entre varias clases (no binaria), a una binaria, mucho más sencilla de manejar.

3.3.3. Redes Neuronales

Las redes neuronales pueden ser definidas como un método de aprendizaje profundo, preparado para reconocer relaciones y patrones, tomar decisiones, prever datos o situaciones, etc., tal y como lo haría un cerebro de un humano, uniendo conexiones y creando un entendimiento de las mismas. La ecuación que ejecutan las neuronas para calcular su valor, se muestra en la ecuación (3.10).

$$z = w_1 \cdot x_1 + w_2 \cdot x_2 + b, \tag{3.10}$$

donde:

- z es la salida de la neurona.
- w_1 y w_2 son los pesos sinápticos correspondientes a cada entrada.
- x_1 y x_2 son las entradas de la neurona.
- b es el sesgo (bias) de la neurona.

A partir de unos datos correctamente categorizados, por lo que se estaría hablando de nuevo de un sistema de aprendizaje supervisado como con SVM, la red neuronal se entrena aprendiendo de los errores e intentando optimizar su aprendizaje modificando los datos de entrada, mezclándolos, haciendo operaciones entre ellos, etc. Estas operaciones, representadas en las ecuaciones (3.11), (3.12) y (3.13), son conocidas como funciones de activación y están representadas por las neuronas de la red, las cuales tendrán un peso asignado, el cual define así la importancia de esa neurona en cuanto al resultado final. Además, contarán con el sesgo, que influirá en el procesamiento de los datos a las neuronas:

1. Sigmoide:
$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{3.11}$$
2. ReLU (Rectified Linear Unit):
$$\text{ReLU}(x) = \text{máx}(0, x). \tag{3.12}$$
3. Tanh (Tangente Hiperbólica):
$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}. \tag{3.13}$$

donde:

- $\sigma(x)$ es la función sigmoide, que produce una salida entre 0 y 1.
- ReLU(x) es la función ReLU, que produce una salida igual a x si x es mayor que 0, de lo contrario produce 0.
- tanh(x) es la función tangente hiperbólica, que produce una salida entre -1 y 1.
- x es la entrada a la función de activación.
- e es la base del logaritmo natural, aproximadamente igual a 2.71828.

Ambos valores, los pesos y el sesgo, se calculan y se actualizan automáticamente en el entrenamiento de la red gracias a algoritmos como el descenso del gradiente, que permitirá durante las épocas (número de veces que el algoritmo realiza un bucle buscando nuevos parámetros) minimizar el coste [16], representado en la ecuación (3.14) y siendo este la diferencia entre los valores predichos y los reales.

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2, \tag{3.14}$$

donde m es el número de épocas, y_i son los valores de las medidas, y $\hat{y}_i$ representa las predicciones.

La disposición de las neuronas es un punto muy importante a la hora de diseñar la red. En la Figura 3.14 se distinguirán 4 capas:

1. Capa de entrada o *input layer*, siendo esta comúnmente el mismo número de neuronas que características o *features* se esté trabajando.
2. Dos capas ocultas, o *hidden layers*, con sus funciones de activación que permitirán obtener relaciones no lineales y por tanto, encontrar conexiones más complejas.
3. Capa de salida u *output layer*, que representará el resultado de la red neuronal.

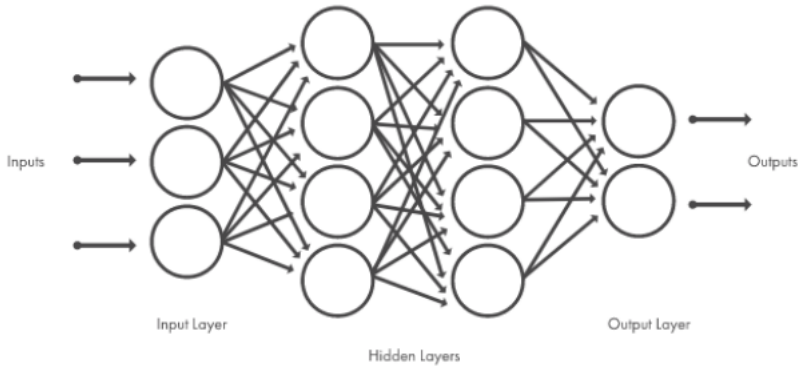

Figura 3.14: Esquema sencillo de una red neuronal [17].

Otro punto de posible optimización de las redes neuronales es su tasa de aprendizaje o *learning rate*, visualmente definido como el tamaño de los pasos que da la red para encontrar en la función de coste el mínimo global, a la misma vez que se modifican los pesos y sesgos, calculados a través de las ecuaciones (3.15) y (3.16) [16]:

$$w := w - \alpha\frac{\partial J(w,b)}{\partial w} \tag{3.15}$$

$$b := b - \alpha\frac{\partial J(w,b)}{\partial b} \tag{3.16}$$

donde w es el peso de las neuronas, b es el bias, y α representa el *learning rate*.

El mal ajuste de estos parámetros puede hacer que el sistema se encuentre con situaciones de no aprendizaje de los patrones, ya que memoriza o se ajusta demasiado a ciertos datos y luego no puede extrapolar ese aprendizaje a datos de prueba no vistos anteriormente, a lo que se denomina como overfitting. También pueden surgir problemas de convergencia, producidos cuando se tiene un *learning rate* grande que impide encontrar el punto de inflexión más bajo. Por el contrario,

si es pequeño se enfrentará a problemas de eficiencia computacional, al estar dando pasos muy pequeños a lo largo de la función.

La falta de datos es un problema al entrenar el algoritmo, por lo que otro inconveniente sería la necesidad de una gran tamaño del conjunto de entrenamiento, y preferentemente con clases equilibradas en cuanto al número de muestras. A modo de ejemplo del equilibrio necesario, podría imaginarse un estudio de clasificación entre camisetas o camisas. Entrenar este modelo con un 88 % de entradas siendo camisetas y el restante camisas, podría establecer un sesgo que no convendría si se desea conseguir el mayor número de aciertos posible.

Por otra parte y gracias a la versatilidad que ofrecen estos algoritmos, se tienen infinidad de herramientas para poder lidiar con estos inconvenientes. Algunos ejemplos serían el *dropout*, en el que con el paso de las épocas se "apagan" ciertas neuronas estratégicas, evitando así la memorización pura de los datos; o el *EarlyStopping*, en el que se pararía el entrenamiento de la red neuronal una vez sus validaciones no mejoren en un rango de épocas, ahorrando así recursos computacionales.

En relación con los tipos de redes neuronales, hay que aclarar que esta clasificación es general, ya que el diseño de una red neuronal es tan amplio y adaptable a cada proyecto, que sería complicado definir todas y cada una de las redes disponibles. Un resumen general de estos tipos [18][19] sería:

1. **Red Neuronal general**
   Están formadas por capas de neuronas conectadas entre sí. Son las más básicas para la búsqueda de esos patrones. Al ser la estructura general, dependiendo del tipo de objetivo y datos que se manejen, podrán ser las más eficientes o no.

2. **Red Neuronal convolucional**
   Este tipo de redes está muy extendido en el reconocimiento de imágenes y objetos. Para ello, maneja varios tipos de capas con una función diferente cada una.

3. **Red Neuronal recurrente**
   Las redes neuronales recurrentes se usan para aprender lenguaje, como NLP (*Natural Language Processing*), usando secuencias de datos o series temporales. Su principal característica es la influencia de resultados previos en los futuros, teniendo una retroalimentación que ayuda a mejorar el proceso de aprendizaje. Otro hecho diferencial con las anteriores es que todas las capas tienen los mismos pesos, aunque estos también cambian con la optimización del gradiente.

Capítulo 4

Descripción de las medidas

Con el fin de caracterizar el canal, se han utilizado medidas de propagación obtenidas en anteriores investigaciones [9], las cuales han facilitado la tarea de procesamiento para su posterior clusterización y clasificación con algoritmos de IA.

4.1. Primeras medidas

4.1.1. Características

Las primeras medidas utilizadas se obtuvieron con una sonda de canal implementada en el dominio de la frecuencia a través de un analizador de redes vectorial (VNA, *Vector Network Analizer*). Se utilizaron antenas omnidireccionales en el plano horizontal y con polarización lineal vertical. El analizador de redes medía el parámetro de scattering *S21(f)*, equivalente a la función de transferencia compleja del canal, *H(f)*. La respuesta impulsional del canal (CIR, Channel Impulse Response), indicada por $h(\tau)$, se obtiene a partir de la transformada inversa de Fourier de *S21(f)*, y el PDP se obtiene a partir de la CIR mediante la ecuación (4.1):

$$\text{PDP}(\tau) = |h(\tau)|^2 \tag{4.1}$$

El parámetro *S21* realmente indica la atenuación del canal en la banda medida (*SPAN* en el VNA), al relacionar el nivel de señal recibido en función del nivel transmitido, por lo que el PDP así calculado es adimensional pero proporcional, en un factor de escalado igual a la potencia recibida, a la densidad de potencia en la variable de retardo.

Las señales se emitieron en la banda de frecuencias milimétricas, desde 25 GHz a 40 GHz más concretamente, teniendo usos como el desarrollo de tecnologías 5G, tan extendidas hoy en día, como además en radares, escáneres, etc.

Para poder recopilar la mayor cantidad de información posible, las medidas se han procesado además con el algoritmo SAGE, permitiendo modelar y estimar los ángulos de acimut y elevación con los que se han recibido las medidas en la antena situada en el array.

En la Tabla 4.1, quedan representados los parámetros principales de las primeras medidas.

Posiciones del array URA	12x12
Número de puntos	8192
BW (Ancho de banda)	25 - 40 GHz
Separación entre posiciones	3.04 mm
Frecuencia de muestreo	1.831 MHz

Tabla 4.1: Parámetros principales de las primeras medidas.

Dentro de la banda de frecuencia utilizada, a su vez se ha dividido en 4 subandas. Esto conllevará un posible análisis más específico, en el que sea trivial identificar las peculiaridades de cada rango. Por ello, quedan definidas cada subanda con los parámetros de la Tabla 4.2.

B1	25 - 27.5 GHz	1365 muestras	1-1365
B2	27.5 - 29.5 GHz	1092 muestras	1366-2457
B3	31.8 - 33.4 GHz	874 muestras	3714-4587
B4	37 - 40 GHz	1639 muestras	6554-8192

Tabla 4.2: Bandas de frecuencia en las medidas.

4.1.2. Contexto de las medidas

Estos primeros datos se obtuvieron gracias a las medidas realizadas en la CPI (Ciudad Politécnica de la Innovación), en la UPV (Universidad Politécnica de Valencia). En este edificio se podrá encontrar la sala donde se midió, la cual forma parte del GRE (Grupo de Radiación Electromagnética), grupo de investigación componente del iTEAM. El plano del entorno se observa en la Figura 4.1.

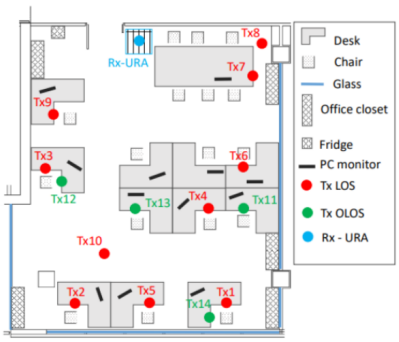

Figura 4.1: Mapa sala de becarios iTEAM (UPV) [9].

Como se puede apreciar en la Figura 4.1, la antena transmisora se ha ido desplazando mientras que el receptor ha quedado estático. De esta forma se han obtenido hasta 14 mediciones, pudiendo

identificar mejor los obstáculos de la sala que han hecho que 10 medidas tuvieran visión directa o LOS entre la antena transmisora y la receptora, y 4 sin visión directa o NLOS. Estas medidas no solo han servido para tener estas dos situaciones de visión directa o no, sino que a través de ellas se conseguirán identificar qué elementos han causado reflexiones, refracciones, mayores pérdidas, etc. y así poder modelizar el entorno de medidas.

4.2. Segundas medidas

4.2.1. Características

En este segundo conjunto de datos [20], de nuevo proyectados de investigaciones pasadas, el esquema ha sido muy similar al anterior, con los mismos elementos, pero con ciertos detalles diferentes. En cuanto a los equipos utilizados se sigue disponiendo de dos antenas, receptora y transmisora, junto con el VNA.

Una diferencia que se destaca respecto a las otras medidas sería que también en el transmisor se tiene un *array* por el que se ha ido desplazando la antena, por lo que quedaría un esquema con un *array* URA de 7x7 en el receptor combinado con un *array* longitudinal uniforme o ULA (*Uniform Longitudinal Array*) de 1x10 en el transmisor. Esto significan 490 medidas por puesto en la sala.

En la Tabla 4.3, quedan representados los parámetros principales de las segundas medidas.

Posiciones del array ULA	1x10
Posiciones del array URA	7x7
Número de puntos	8192
BW (Ancho de banda)	24 - 40 GHz
Separación entre posiciones	3.04 mm
Frecuencia de muestreo	1.953 MHz

Tabla 4.3: Parámetros principales de las segundas medidas.

Otro parámetro distinto sería el rango de frecuencias, siendo ahora de 24 GHz a 40 GHz, abarcando así un rango de 16 GHz, en vez de 15 GHz. Al tener el mismo número de muestras, 8192, se ha simplificado mucho el procesado de datos, aunque la frecuencia de muestreo se ha visto alterada ligeramente.

4.2.2. Contexto de las medidas

La sala, representada en la Figura 4.2, en las que se han tomado estas medidas también es diferente, aunque perteneciente al mismo edificio que las anteriores, en la CPI de la UPV.

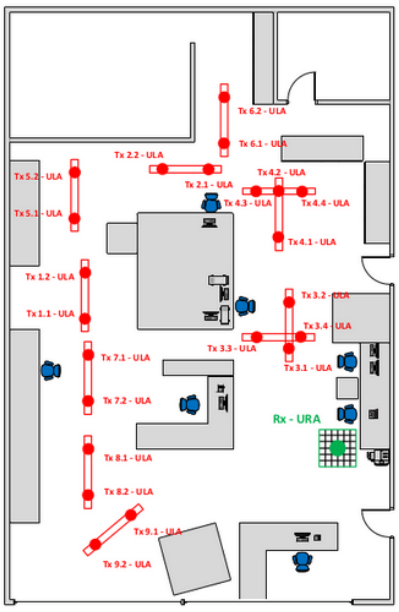

Figura 4.2: Mapa de laboratorio iTEAM (UPV) [20].

Capítulo 5

Clusterización

Como ya se viene indicando en páginas previas, uno de los objetivos principales del proyecto es estandarizar el entorno donde se realizaron las mediciones, pudiendo permitir a agentes externos a esta investigación aprovechar la información expuesta para el conocimiento del comportamiento del canal en escenarios similares.

La programación de este capítulo, que como ya se ha indicado ha sido a través del lenguaje *Python*, ha requerido del uso de múltiples bibliotecas públicas que han facilitado tanto el procesamiento de las medidas como la implementación del algoritmo K-means. Fueron dos motivos por los que se eligió *Python* y no Matlab, tan extendido en la ETSIT (Escuela Técnica Superior de Ingenieros de Telecomunicación), para el proyecto: aprendizaje y facilidad.

Según la Universidad ORT de Uruguay [21], *Python* es uno de los lenguajes de programación más ampliamente utilizados, gracias en parte a su versatilidad en el diseño para múltiples propósitos, pero destacando especialmente en análisis de datos, IA, etc. Además, una de las características, como ya se ha comentado, es su facilidad; con menos de 7 librerías importadas, ha sido posible el diseño del proyecto. Estas son las recogidas en la Tabla 5.1.

Librerías utilizadas
Scipy
Numpy
Matplotlib
kmeans
sklearn
pandas

Tabla 5.1: Librerías de *Python* empleadas.

En base a estas herramientas se ha implementado el algoritmo de clusterización K-means, aunque previamente han sido necesarios varios procesos de manejo de datos.

5.1. Procesamiento de medidas

Las medidas analizadas en este trabajo de investigación, con las características mostradas en el Capítulo 4, han tenido que ser procesadas para adaptarlas a las exigencias del algoritmo de clusterización y sus necesidades.

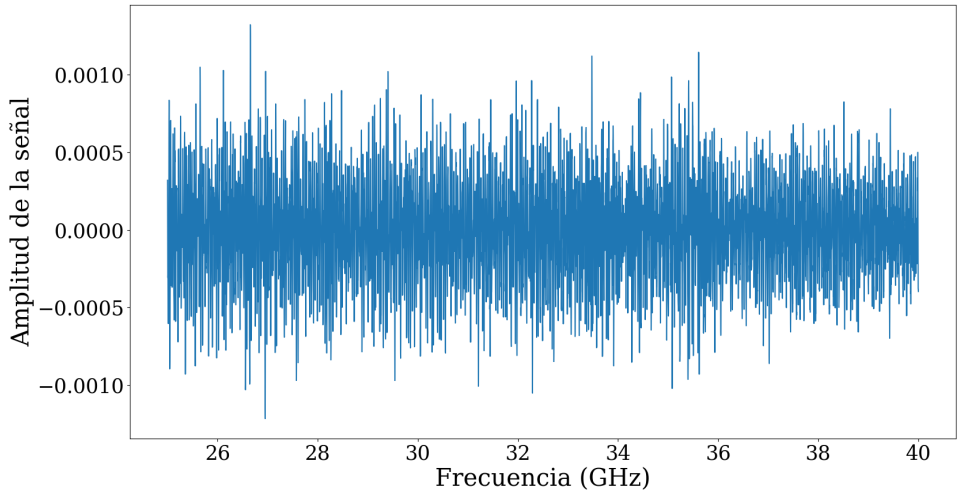

Figura 5.1: Visualización Amplitud-Frecuencia.

En la Figura 5.1 se reflejan los datos de las medidas en bruto, sin ningún tipo de procesamiento. Estos datos vienen en matrices, en los cuales los datos en frecuencia y en amplitud están organizados por columnas.

Como se puede observar, de la Figura 5.1 no es trivial sacar conclusión alguna, ya que su amplitud sólo indica cuánto contribuye cada frecuencia, por lo que conviene transformarlo al dominio tiempo aplicándole la IFFT.

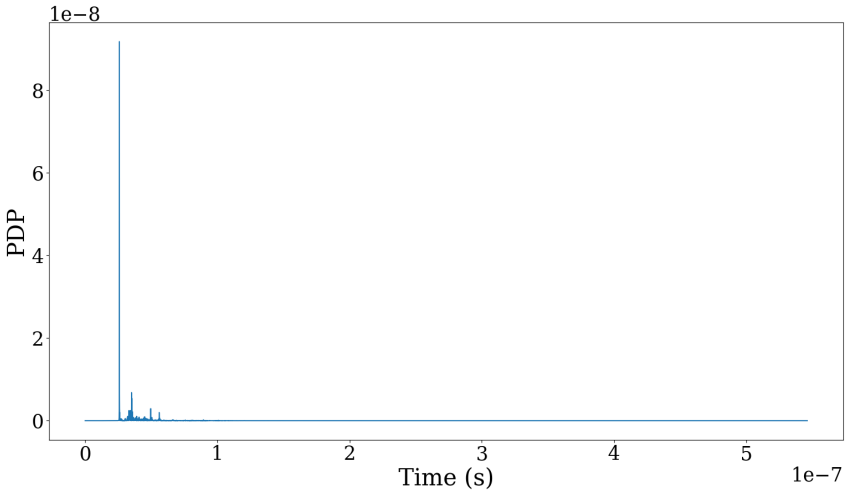

Figura 5.2: PDP en unidades lineales.

En las Figuras 5.2 y 5.3 sí que se puede ya visualizar más nítidamente la señal recibida con sus múltiples contribuciones, tanto en escala lineal como en logarítmica.

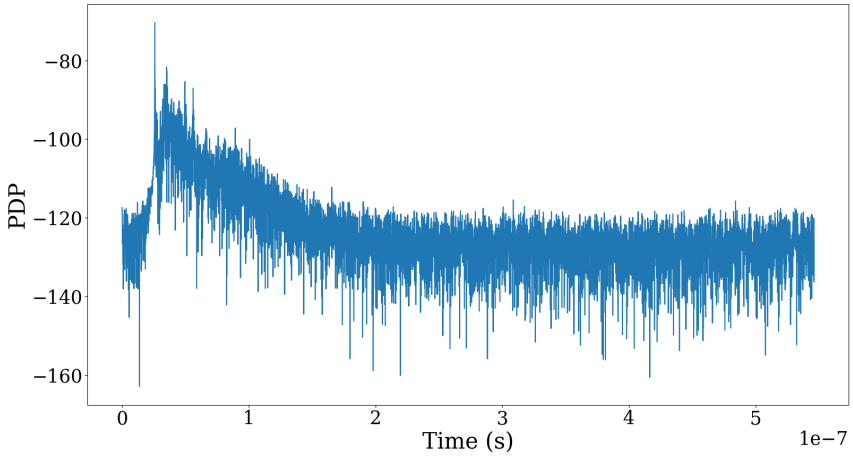

Figura 5.3: PDP en unidades logarítmicas.

Habiendo llevado a cabo estos pasos, se obtendrá el APDP (*Average Power Delay Profile)* de cada posición del URA, calculando el promedio de las 144 medidas y pudiendo así mostrar una gráfica más limpia, sin tantos desvanecimientos, como se contempla en la Figura 5.4.

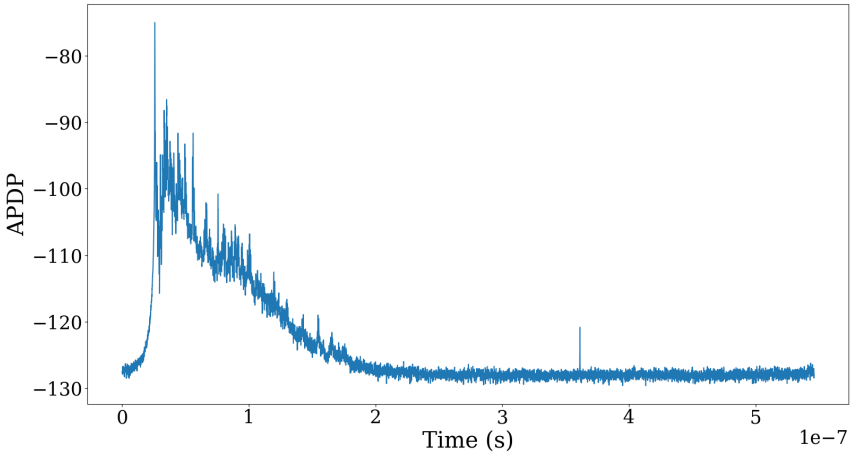

Figura 5.4: APDP de medida LOS_POS_1.

Hasta el momento, exclusivamente se dispone de la información expuesta en los gráficos anteriores: amplitud de potencia y retardo, lo que deja información sin conocer, como los ángulos de llegada al URA. Es por esto por lo que ejecutando SAGE se obtendrá esa información necesaria gracias a su estimación de la señal y sus atributos. Siendo así y escogiendo cualquiera de las 4 bandas de frecuencia, explicadas en el Capítulo 4, se estará en disposición de clusterizar los datos obtenidos.

Destacar por último que, precisamente por la necesidad de utilización de SAGE, en esta primera parte del proyecto exclusivamente se han utilizado las primeras medidas, al no haber procesado con este algoritmo las segundas.

5.2. Algoritmos de clusterización

La clusterización es la acción de agrupar, en este caso las MPCs de las medidas realizadas, atendiendo a una serie de factores de análisis previamente escogidos. Las muestras del mismo clúster deberán de tener una cierta similitud en esos factores comparada con la de otros clústeres, calculando la proximidad a través de operaciones matemáticas como las distancias euclídeas.

Las implementaciones para clusterizar son diversas, que dependiendo del objetivo final y del tipo de datos, se escogerán para tal fin. En este caso, K-means gracias al aprendizaje no supervisado, agrupará los datos de las MPCs para su posterior parametrización con el modelo SV.

5.2.1. K-means

Con la posibilidad que ofrece K-means de poder clusterizar datos con más de 2 variables, primero se tratará de conocer qué tipos de datos devuelven una agrupación óptima y tras ello, proceder a la clusterización de los mismos.

A modo de prueba, se estudiarán la primera subanda de frecuencias B1 de la primera posición LOS y la primera subanda B1 de la cuarta posición NLOS , basando ambas su clusterización en los datos de amplitud, retardo, acimut y elevación. Para ello, y previo a la ejecución de K-means, será necesario conocer el número de clústeres óptimos para el conjunto de datos, por lo que se tomarán en consideración las gráficas referenciadas en el Capítulo 3 sobre los aspectos técnicos. Estas gráficas aportarán una primera estimación del número k.

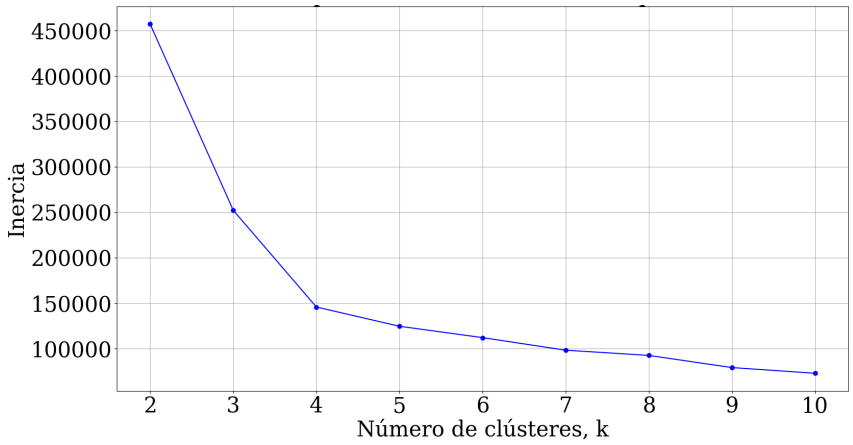

Figura 5.5: Gráfica del método del codo para la banda B1 de la posición 1 de LOS.

Siguiendo la estimación de la métrica del codo, mostrada en la Figura 5.5, la inercia tiene como punto de inflexión de la curva en 4 clústeres, donde la pendiente negativa comienza paulatinamente a moderarse.

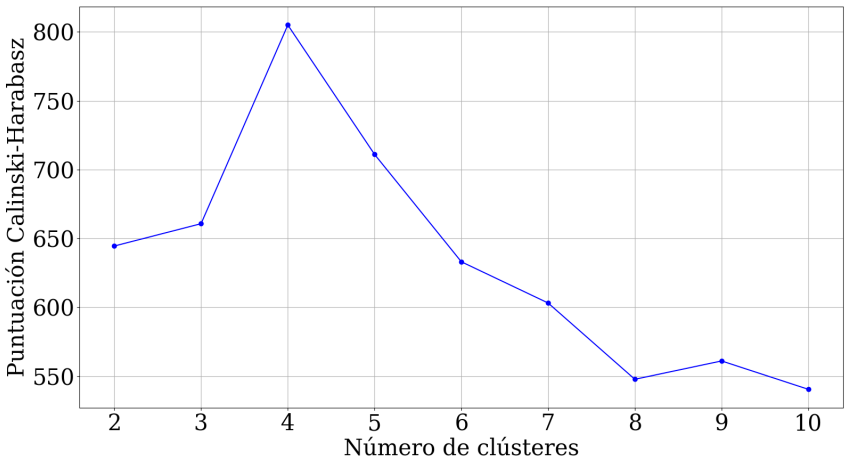

Figura 5.6: Gráfica del índice Calinski-Harabasz para la banda B1 de la posición 1 de LOS.

Según el índice Calinski-Harabasz, reflejado en la Figura 5.6, 4 también representa el número óptimo de clústeres en los que agrupar las muestras.

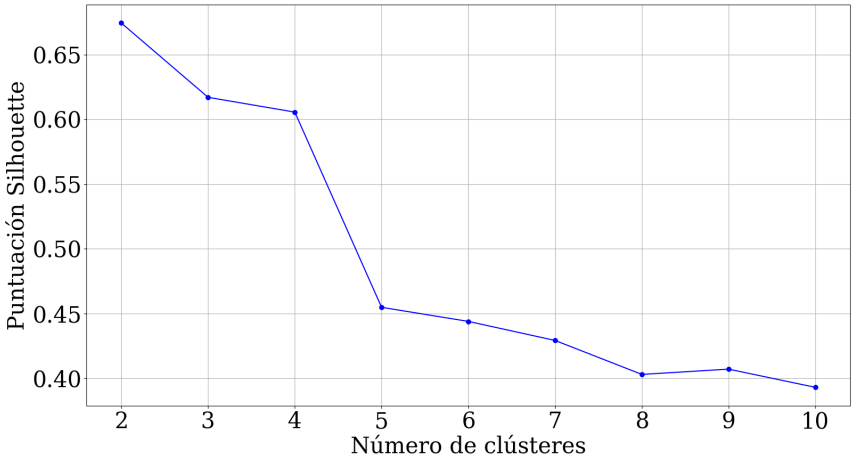

Figura 5.7: Gráfica del índice Silhouette para la banda B1 de la posición 1 de LOS.

Si se toma en consideración la puntuación ofrecida por el último índice utilizado, el índice Silhouette, descrito en la Figura 5.7, 4 también sería una opción pertinente para la ocasión, al estar por encima de 0.5 en la puntuación.

Por consiguiente, tras haber definido gracias a las métricas el número k en el que agrupar las muestras, el resultado plasmado en un gráfico 3D quedaría tal y como se representa en la Figura 5.8.

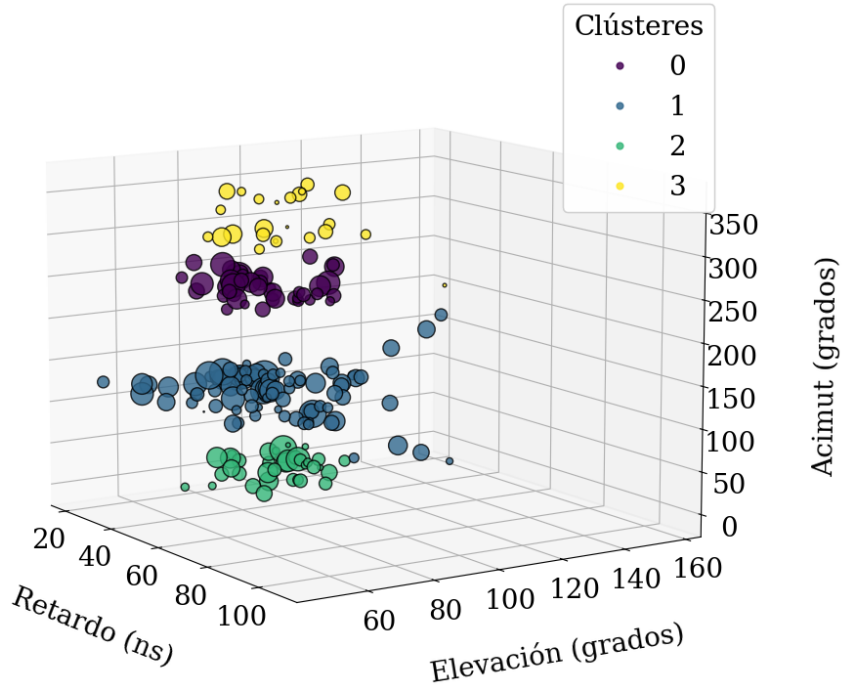

Figura 5.8: Gráfica 3D de los datos SAGE clusterizados con K-means para la banda B1 de la posición 1 de LOS con k=4.

Como primer objetivo de la rama de clusterización del proyecto se tiene que concretar qué informaciones de las muestras recibidas son más sensibles para poder agrupar de una forma óptima los datos. Es por esto por lo que se representarán en gráficas 2D los diferentes tipos de datos de las muestras obtenidos por SAGE, pudiendo definirlos de una manera más sencilla e incluso visual.

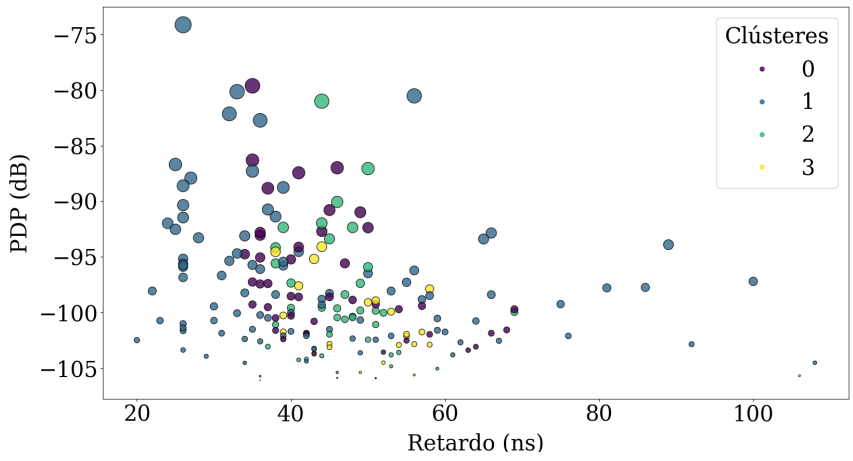

Figura 5.9: Gráfica Amplitud vs. Retardo de los datos SAGE clusterizados con K-means para la banda B1 de la posición 1 de LOS con k=4.

En la Figura 5.9 se representan las muestras clusterizadas por K-means en una gráfica amplitud-retardo. Siguiendo el esquema de colores, poder concluir u obtener resultados claros es complicado.

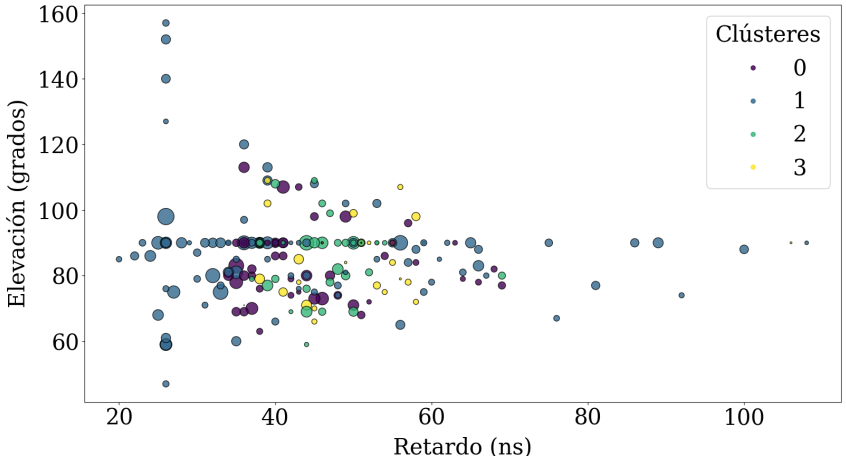

Figura 5.10: Gráfica Elevación vs. Retardo de los datos SAGE clusterizados con K-means para la banda B1 de la posición 1 de LOS con k=4.

Modificando los ejes de la gráfica y mostrando ahora el ángulo de elevación respecto al retardo en la Figura 5.10, se percibe lo mismo que en la anterior Figura 5.9, y es la imposibilidad de detectar un patrón en la clusterización.

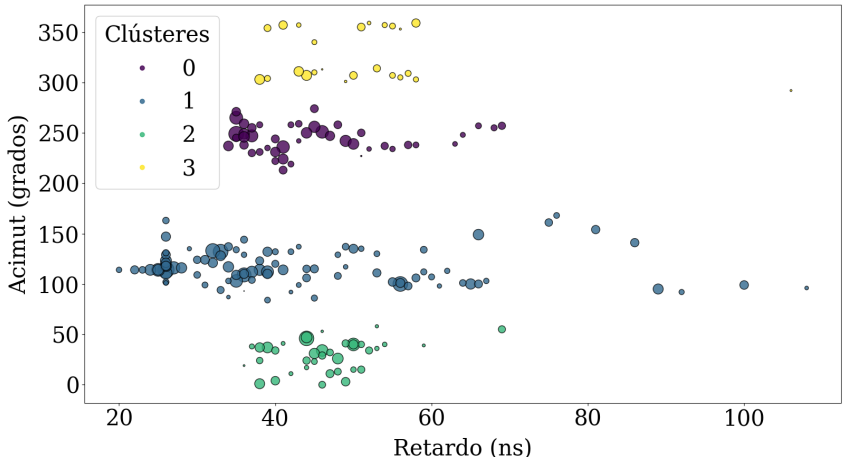

Figura 5.11: Gráfica Acimut vs. Retardo de los datos SAGE clusterizados con K-means para la banda B1 de la posición 1 de LOS con k=4.

Para finalizar con las informaciones de las muestras, se representa el ángulo de acimut con el retardo en la Figura 5.11, teniendo ahora sí una agrupación clara y visual. Esto lo que lleva a entender es la importancia del ángulo de llegada acimut a la hora de clusterizar las muestras recibidas por URA.

Con el fin de comprobar la conclusión anterior, se llevará a cabo el mismo procedimiento con los clústeres de una posición en la que las antenas transmisora y receptora carecían de visión directa.

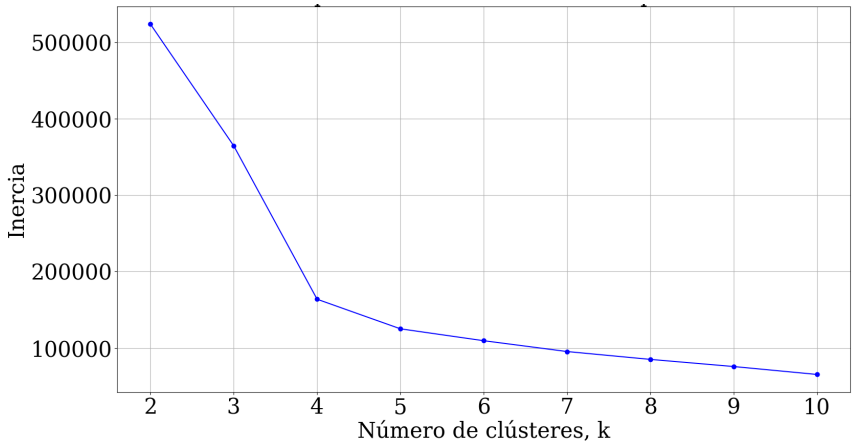

Figura 5.12: Gráfica del método del codo para la banda B1 de la posición 4 de NLOS.

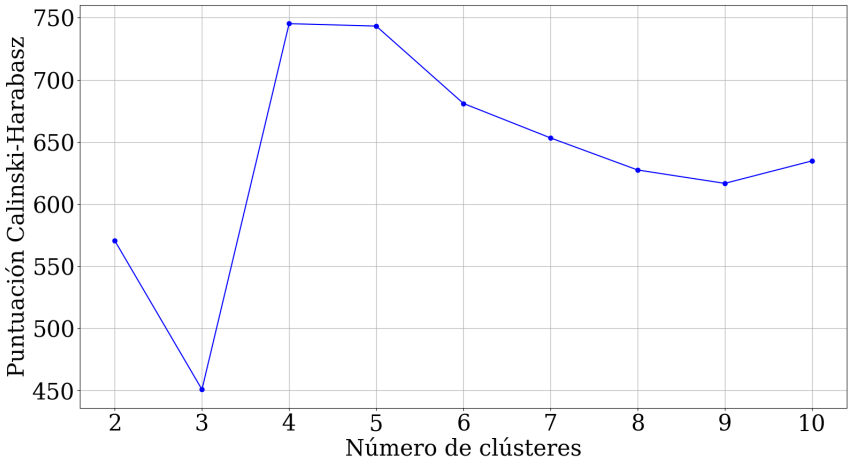

Figura 5.13: Gráfica del índice Calinski-Harabasz para la banda B1 de la posición 4 de NLOS.

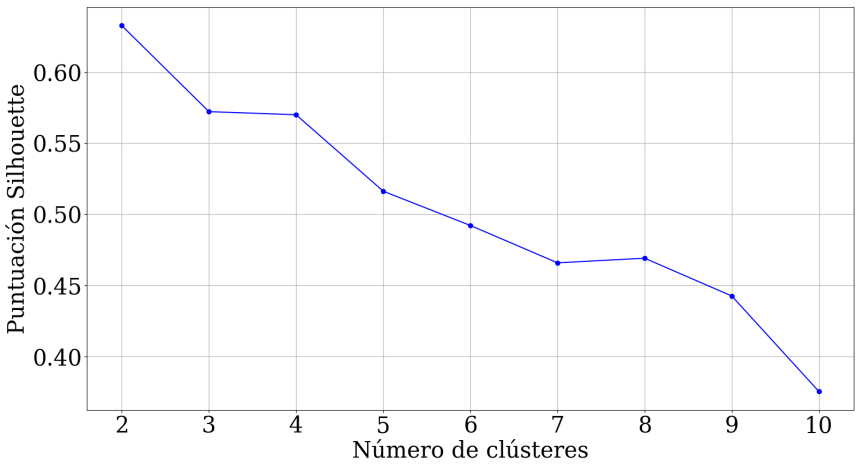

Figura 5.14: Gráfica del índice Silhouette para la banda B1 de la posición 4 de NLOS.

Obteniendo las gráficas de las métricas similares al ejemplo de LOS, representadas en las Figuras 5.12, 5.13 y 5.14, se seguirá el mismo criterio para determinar que 4 vuelve a ser una de las cifras candidatas a k óptimo.

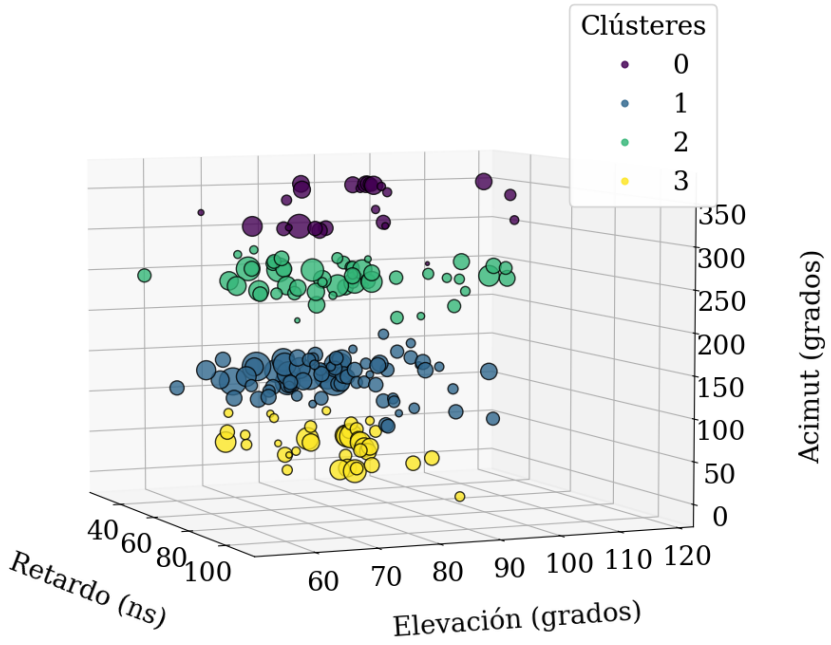

Figura 5.15: Gráfica 3D de los datos SAGE clusterizados con K-means para la banda B1 de la posición 4 de NLOS con k=4.

Tras la representación en 3D en la Figura 5.15, se volverá a desglosar la clusterización con gráficas 2D revisando las diversas informaciones aportadas por SAGE, representadas en las Figuras 5.16, 5.17 y 5.18.

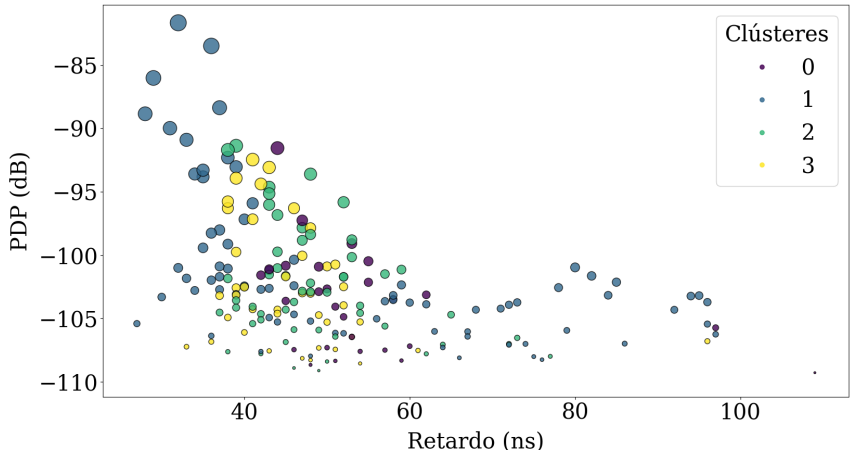

Figura 5.16: Gráfica Amplitud vs. Retardo de los datos SAGE clusterizados con K-means para la banda B1 de la posición 4 de NLOS con k=4.

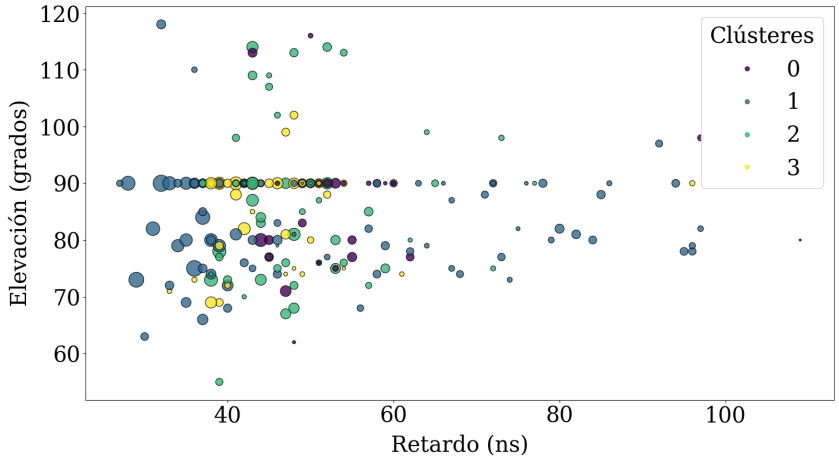

Figura 5.17: Gráfica Elevación vs. Retardo de los datos SAGE clusterizados con K-means para la banda B1 de la posición 4 de NLOS con k=4.

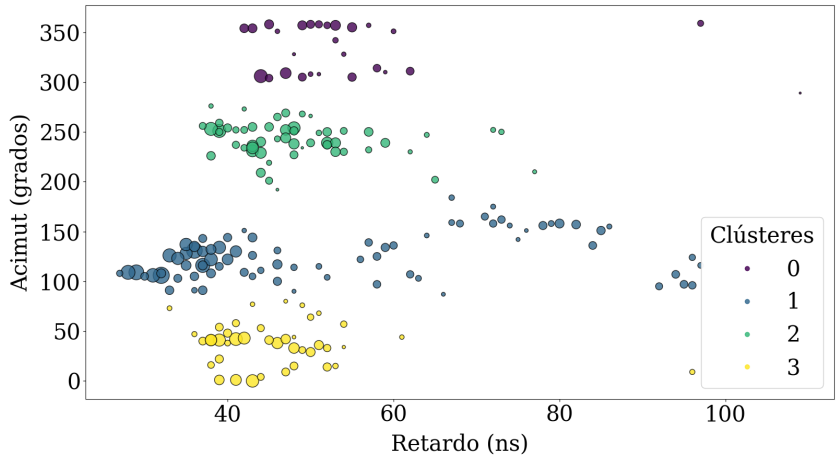

Figura 5.18: Gráfica Acimut vs. Retardo de los datos SAGE clusterizados con K-means para la banda B1 de la posición 4 de NLOScon k=4.

Se puede reafirmar que el ángulo de acimut aporta la información con mayor influencia en la clusterización de los datos medidos por parte de K-means.

Toda esta evaluación anterior sobre el peso de las informaciones ha estado bajo la consideración de que el número k óptimo es el devuelto por las métricas e índices utilizados, pero nada más lejos de los objetivos establecidos. Con la finalidad de poder modelizar el entorno de mediciones, se ha de intentar conseguir la mayor precisión posible, siempre en conjunción con la eficiencia en la clusterización de los datos.

Si se observan las gráficas de las Figuras 5.8 y 5.15 se podría comentar que la clusterización habría realizado un agrupamiento correcto, pero si se revisa más nítidamente la información clusterizada en gráficas como las Figuras 5.11 y 5.18 se detectan muestras agrupadas en un mismo clúster, pero con una diferencia en el ángulo de acimut recibido mayor de 90 grados.

Número de clúster	Rango (grados)
1	61
2	84
3	58
4	67

Tabla 5.2: Rango de ángulos correspondiente a cada clúster para la banda B1 de la posición 1 de LOS con k=4.

Número de clúster	Rango (grados)
1	70
2	97
3	84
4	80

Tabla 5.3: Rango de ángulos correspondiente a cada clúster para la banda B1 de la posición 4 de NLOS con k=4.

Con los valores mostrados tanto en la Tabla 5.2 como en la Tabla 5.3 , poder sacar conclusión alguna de cómo el entorno está afectando a la señal transmitida y estar en disposición de realizar cambios para mejorar la comunicación se vuelve difícil. Para un análisis profundo de qué objetos están reflejando mayores MPCs, en qué situaciones, etc., será necesario poder reducir los rangos, aportando así una mayor precisión.

Para comenzar con la revisión del parámetro k, se tendrán que volver a tener en cuenta las métricas, ya que, pese a no escoger el k óptimo que indican, sí que se eligirá dentro del intervalo que cumple las expectativas. Tomando en consideración un $k \in \{5, 6, 7\}$, siendo estos valores aceptables, se resuelve como mejor opción k=7.

La determinación de 7 como k óptimo tiene como base no solo las métricas, sino también los intervalos de las muestras de los clústeres y la visión crítica tras observar tanto la Figura 5.19 como el gráfico polar de la Figura 5.21, donde se representarán las muestras según el acimut y el retardo.

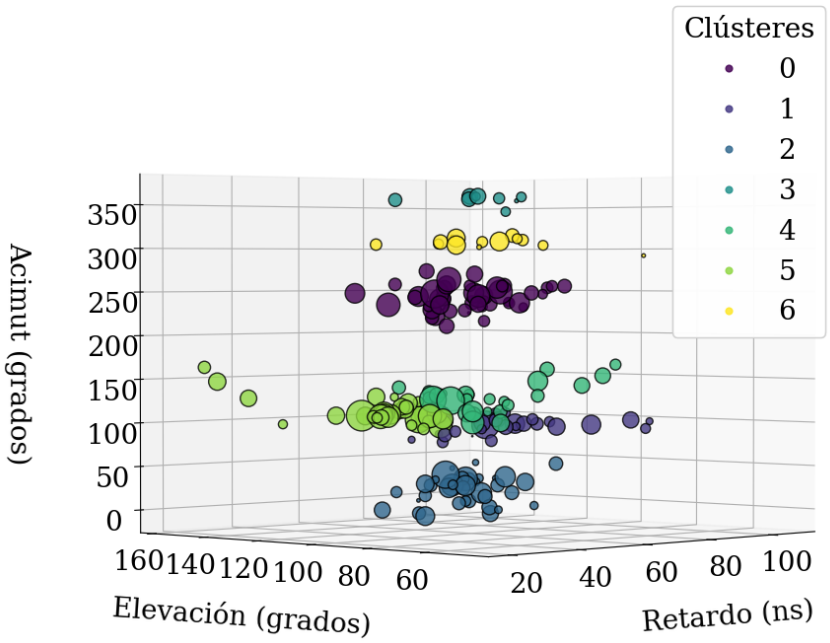

Figura 5.19: Gráfica 3D de los datos SAGE clusterizados con K-means para la banda B1 de la posición 1 de LOS con k=7.

Respecto al espectro en grados perteneciente a cada clúster, se detecta fácilmente cómo al aumentar el número de clústeres se reducen sensiblemente estos rangos mostrados en la Tabla 5.4, acaparando como máximo un rango de 62 grados y como mínimo 19, lo que aportará una precisión notablemente mayor.

Número de clúster	Rango (grados)
1	61
2	29
3	58
4	19
5	59
6	62
7	22

Tabla 5.4: Rango de ángulos correspondiente a cada clúster para la banda B1 de la posición 1 de LOS con k=7.

Tanto en la Figura 5.20 como en la Figura 5.21 se muestran las direcciones con las que se han recibido las muestras en el plano horizontal, además de su retardo. Si se pusiesen estas muestras en un plano real de la sala de becarios donde se midió, podría aportar información muy valiosa a la hora de identificar elementos que entorpezcan de alguna manera el objetivo de conseguir una

comunicación estable, segura y eficiente.

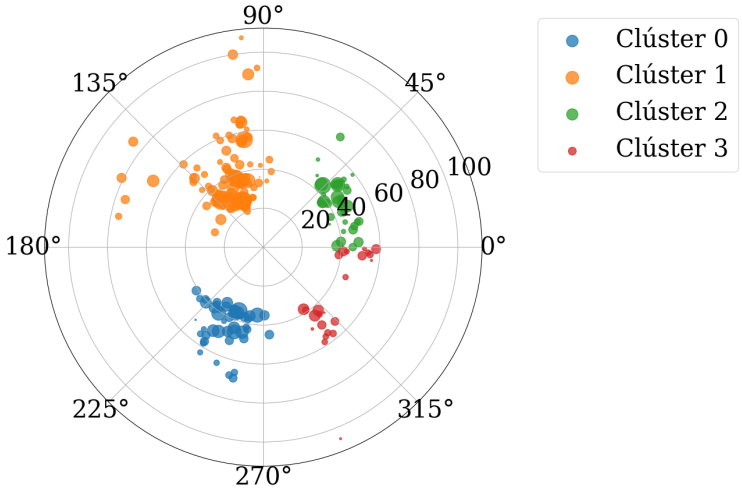

Figura 5.20: Gráfico polar de los datos SAGE clusterizados con K-means para la banda B1 de la posición 1 de LOS con k=4.

Comparando ambas gráficas polares, a simple vista se reconocen los cambios en la clusterización, en la que K-means ha tenido más en cuenta el retardo en la Figura 5.21 que en la Figura 5.20. Además, los límites en cuanto a ángulo también se han estrechado significativamente, separando muestras que anteriormente estaban agrupadas en un mismo clúster cuando muy posiblemente los caminos recorridos hasta alcanzar la antena receptora fueran muy diferentes.

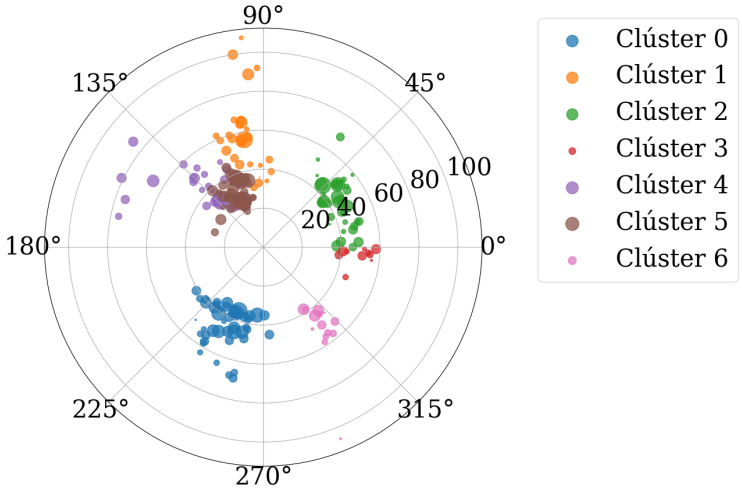

Figura 5.21: Gráfico polar de los datos SAGE clusterizados con K-means para la banda B1 de la posición 1 de LOS con k=7.

Con el caso de ejemplo de NLOS se obtendrían resultados paralelos. De nuevo k óptimo correspondería con 7, ya que de otra manera no se conseguiría una agrupación con límites reducidos en cuanto al acimut.

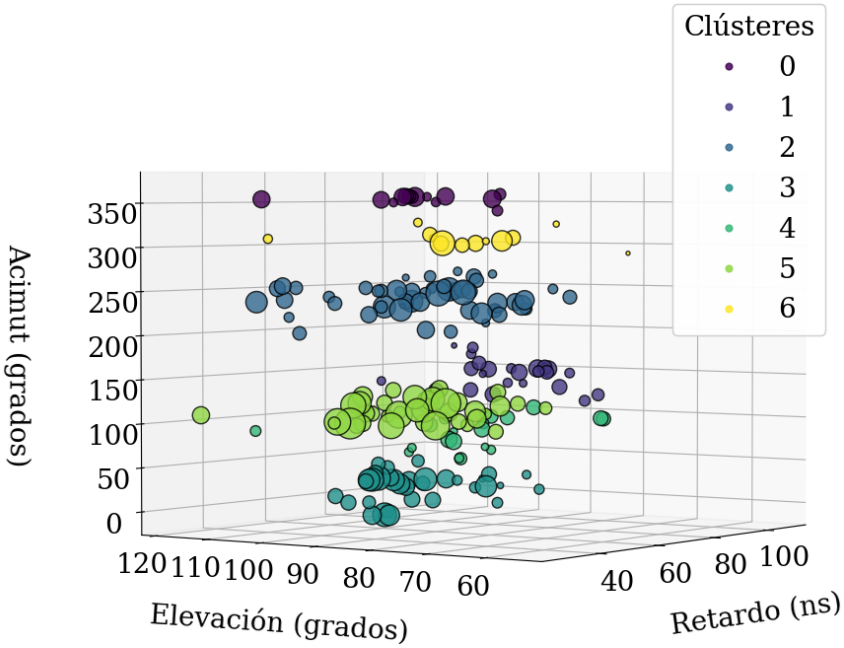

Figura 5.22: Gráfica 3D de los datos SAGE clusterizados con K-means para la banda B1 de la posición 4 de NLOS con k=7.

Si se enfrentan las Figuras 5.19 y 5.22, se diferencian en que en la segunda situación, la correspondiente con NLOS , se agrupan mejor las muestras al tener menor superposición entre los clústeres que en la Figura 5.19. La superposición es un peaje que responde al aumento del número de grupos, lo que significa que lo eficiente en cuanto a agrupaciones y reducción de distancias no tiene por qué coincidir con lo eficiente para los objetivos perseguidos, como ocurre en el presente estudio.

Número de clúster	Rango (grados)
1	17
2	76
3	75
4	58
5	43
6	41
7	39

Tabla 5.5: Rango de ángulos correspondiente a cada clúster para la banda B1 de la posición 4 de NLOS con k=7.

Tras notar una reducción notable en los rangos mostrados en la Tabla 5.5, se aportan los gráficos polares de nuevo para poder analizar de una manera más realista y visual las direcciones de llegada de las contribuciones, viendo el cambio al aumentar de 4 a 7 el número de clústeres.

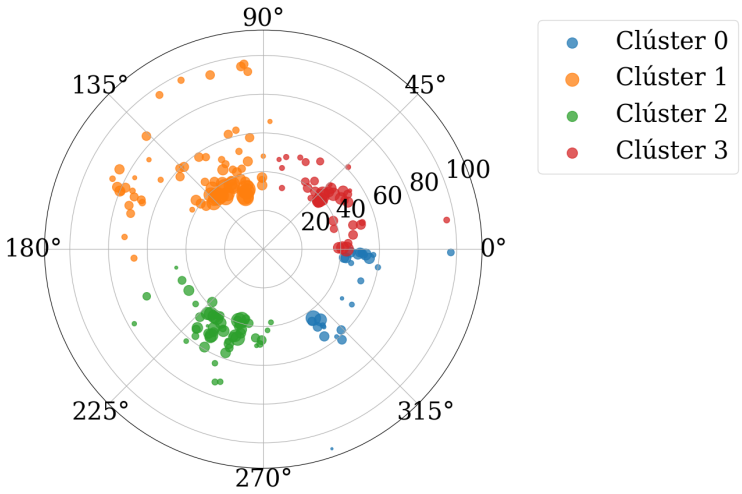

Figura 5.23: Gráfico polar de los datos SAGE clusterizados con K-means para la banda B1 de la posición 4 de NLOS con k=4.

En la Figura 5.23 se encuentra el clúster de color naranja, que abarca casi 1/4 del gráfico, mientras que en la Figura 5.24 se divide entre el clúster naranja, marrón y lila. El clúster morado también se divide, formando el clúster morado, disponiendo de muestras recibidas entorno a 315 grados y el azulado, situado cerca de los 0 grados.

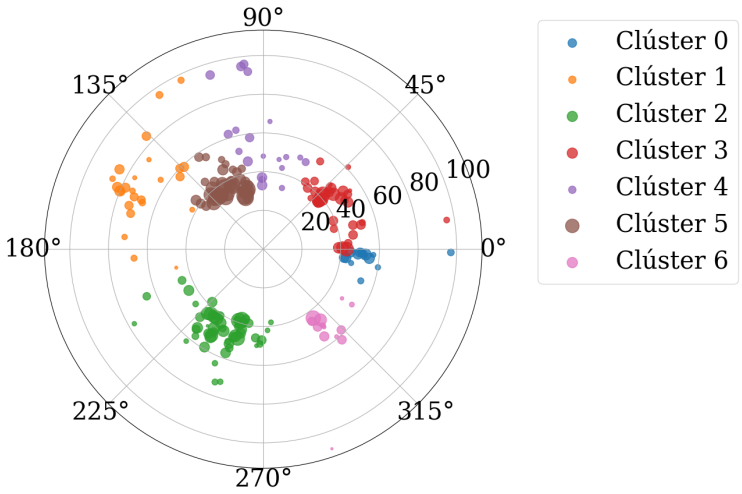

Figura 5.24: Gráfico polar de los datos SAGE clusterizados con K-means para la banda B1 de la posición 4 de NLOS con k=7.

5.3. Parametrización del entorno

Para alcanzar la caracterización del entorno, un paso fundamental es su parametrización a través de los términos que ofrece el modelo SV y que se pueden revisar en el capítulo teórico de la memoria.

Esto es un proceso fundamental en el diseño, análisis y optimización de sistemas de comunicaciones, ya que gracias a conocer las características del canal a través de sus parámetros, permite diseñar estrategias que maximicen la eficiencia y la robustez de la transmisión.

5.3.1. Modelo Saleh-Valenzuela

Con el fin de establecer los parámetros del modelo SV, primero se han tenido que estudiar los ajustes que mejor pueden definir los clústeres resultantes del PDP.

En cuanto a las variables temporales, en el apartado teórico se ha explicado cómo siguen una distribución de Poisson definida en la ecuación (5.1):

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}, \tag{5.1}$$

donde:

- $P(X = k)$ es la probabilidad de que ocurran k eventos en el intervalo de tiempo.
- λ es la tasa promedio de eventos que ocurren en el intervalo de tiempo.
- k es el número de eventos que queremos calcular la probabilidad.

Por tanto, esta distribución define que las MPCs son recibidas con una tasa promedio constante y de manera independiente entre sí.

Además, en el modelo SV también se define el decaimiento de las amplitudes de las muestras recibidas. La distribución γ del decaimiento de las MPCs en el PDP se modela en la ecuación (5.2):

$$P_h(\tau) = \alpha_0 e^{-\frac{\tau}{\gamma}}, \tag{5.2}$$

donde:

- $P_h(\tau)$ es la potencia del canal en función del retardo τ.
- α_0 es la potencia inicial del canal.
- τ es el retardo.
- γ es el parámetro de decaimiento, que representa la constante de tiempo del decaimiento exponencial.

Para representarlo en decibelios (dB), se sigue la ecuación (5.3):

$$10 \log_{10} P_h(\tau) = P_h(\tau)_{dB}. \tag{5.3}$$

Se desarrollan las ecuaciones (5.4) y (5.5):

$$P_h(\tau)_{dB} = 10\log_{10}\alpha_0 + 10\log_{10} e^{-\frac{\tau}{\gamma}}. \tag{5.4}$$

$$P_h(\tau)_{dB} = 10\log_{10}\alpha_0 - \tau\frac{1}{\gamma}10\log_{10} e. \tag{5.5}$$

Lo que se puede simplificar como en la ecuación (5.6):

$$P_h(\tau)_{dB} = A + B\tau, \tag{5.6}$$

donde están presentes los términos definidos en las ecuaciones (5.7) y (5.8):

$$A = 10\log_{10}\alpha_0, \tag{5.7}$$

$$\alpha_0 = 10^{\frac{A}{10}}, \tag{5.8}$$

y en las ecuaciones (5.9) y (5.10):

$$B = -\frac{1}{\gamma}10\log_{10} e, \tag{5.9}$$

$$\gamma = -\frac{1}{B}10\log_{10} e. \tag{5.10}$$

Estos cálculos demuestran que si para parametrizar el entorno con SV se utilizan las amplitudes del PDP en unidades lineales, el ajuste que se debe implementar es el exponencial; pero si se manejan amplitudes en unidades logarítmicas, desarrollando las ecuaciones previas se llega a la conclusión de un ajuste polinómico de orden 1 o ajuste lineal, como se muestra en la Figura 5.25.

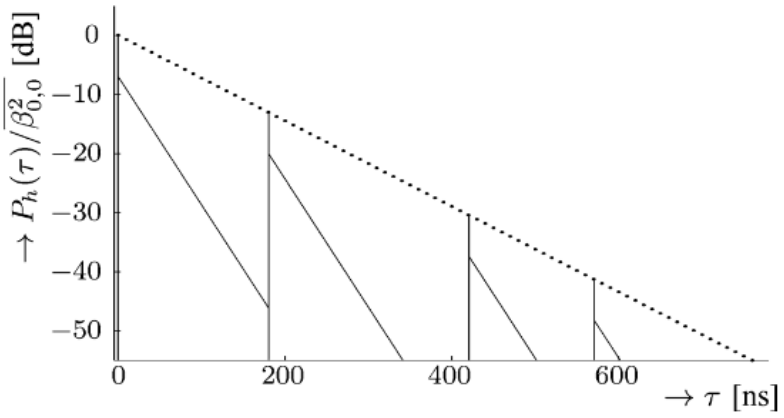

Figura 5.25: Ajuste lineal para el decaimiento de las amplitudes en el PDP [7].

En consecuencia, y siguiendo la ecuación (5.6), se obtendrán los parámetros A y B con el ajuste lineal para extrapolarlos posteriormente en la ecuación (5.8) y la ecuación (5.10). Con estos términos, se estará en disposición de los valores de la distribución γ definida en la ecuación (5.2).

El resultado de la parametrización con el modelo SV se representa en la Tabla 5.6, Tabla 5.7 y Tabla 5.9, mostrándose sus valores medios y desviaciones en la Tabla 5.8 y Tabla 5.10.

Banda	Posición	γ	Γ	λ	Λ
B1	LOS2	0.339	20.409	0.604	0.176
	LOS3	72.036	22.651	0.557	0.175
	LOS4	85.994	15.222	0.588	0.238
	LOS5	41.754	8.313	0.577	0.294
	LOS6	38.676	43.542	0.598	0.160
	LOS8	218.577	44.477	0.640	0.194
	LOS10	114.152	29.894	0.387	0.499
B2	LOS1	18.979	33.611	0.903	0.127
	LOS2	36.785	58.694	0.619	0.166
	LOS3	67.561	21.309	0.481	0.194
	LOS5	22.057	6.808	0.823	0.400
	LOS7	71.972	32.727	0.807	0.169
	LOS8	45.631	128.978	0.850	0.129

Tabla 5.6: Parámetros modelo SV para Bandas B1 y B2 en situaciones LOS.

Banda	Posición	γ	Γ	λ	Λ
B3	LOS2	32.704	46.224	0.762	0.149
	LOS3	64.427	44.212	0.474	0.212
	LOS4	18.285	27.111	0.619	0.210
	LOS5	24.956	11.741	0.750	0.333
	LOS6	14.68	26.02	0.37	0.16
	LOS8	41.911	220.675	0.801	0.140
	LOS10	13.626	41.950	0.556	0.242
B4	LOS1	37.272	43.149	0.656	0.125
	LOS2	110.732	103.800	0.773	0.170
	LOS4	19.697	23.057	0.692	0.294
	LOS5	35.553	14.657	0.830	0.583
	LOS6	23.996	722.843	0.753	0.122
	LOS8	18.628	68.906	0.510	0.138
	LOS10	354.661	36.700	0.574	0.294

Tabla 5.7: Parámetros del modelo SV para Bandas B3 y B4 en situaciones LOS.

	γ	Γ	λ	Λ
Promedio	60.95	70.284	0.65	0.22
Desviación típica	73.563	137.76	0.14	0.11

Tabla 5.8: Promedio y desviación típica de los parámetros modelo SV para situaciones LOS.

Banda	Posición	γ	Γ	λ	Λ
B1	NLOS1	35.062	75.046	0.565	0.117
	NLOS3	41.650	60.804	0.423	0.368
	NLOS4	0.921	13.705	0.616	0.179
B2	NLOS1	173.969	36.142	0.583	0.242
	NLOS2	51.918	106.867	0.573	0.360
	NLOS3	40.307	26.479	0.469	0.333
	NLOS4	726.945	20.083	0.812	0.179
B3	NLOS1	112.837	36.965	0.6443	0.218
	NLOS3	60.326	71.471	0.421	0.135
	NLOS4	140.314	14.661	0.662	0.205
B4	NLOS1	31.449	71.526	0.633	0.173
	NLOS2	268.471	87.861	0.570	0.235
	NLOS3	4.529	38.977	0.687	0.409
	NLOS4	8.820	23.763	1.020	0.194

Tabla 5.9: Parámetros del modelo SV para las 4 bandas en situaciones NLOS.

	γ	Γ	λ	Λ
Promedio	121.25	48.88	0.62	0.24
Desviación típica	190.02	29.75	0.15	0.09

Tabla 5.10: Promedio y Desviación Típica de los Parámetros del Modelo SV para situaciones NLOS.

A la hora de analizar los resultados del modelo SV, y al estar tratando con cálculos estadísticos, se han tenido que realizar unos ajustes con el fin de no falsear las tendencias. Esta medida ha consistido en la eliminación de los datos de ciertas posiciones que proporcionaban puntos impropios. Estos se alejaban del sentido común al devolver una pendiente positiva con el retardo, por lo que se ha decidido prescindir de ellos.

El origen de estos puntos incorrectos son producidos por el efecto de *aliasing*, que ocurre cuando antes de la contribución principal se reciben componentes de elevada amplitud que irían en retardos grandes, y que se mezclan en la traza siguiente. Este efecto se da por un número insuficiente de puntos en la traza, o lo que es lo mismo, baja resolución en frecuencia. Dependiendo de la banda, se tienen más o menos puntos, por lo que se corrige diseñando la campaña de medidas para bandas específicas.

Capítulo 6

Clasificación

En las telecomunicaciones, debido principalmente al gran campo de servicios que ofrecen, la eficiencia es un requisito fundamental para cualquier diseño. Además, no solo basta con la satisfacción de la demanda, sino que también se exige una respuesta rápida, en tiempo real, sin cortes, con calidad extrema y respetuosa con el medio ambiente. Por ello, la optimización del sistema en términos de potencia y retardo que ofrezca una fiabilidad y calidad suficiente se ha convertido en el paso a determinar por cualquier ingeniero.

Para ello, seguir conociendo las características del entorno de mediciones es el principal objetivo del proyecto, explorando y entendiendo cualquier característica, detalle o estado del canal y de los elementos de medida. Es por esto por lo que la segunda parte del trabajo está centrada en el diseño de algoritmos de IA que permitan diferenciar situaciones de LOS y NLOS a través del PDP.

Distinguir si se está ante dos antenas con visión directa o no, tiene una vital importancia en aplicaciones como las recientes tecnologías 5G y 6G, geolocalización, IoT (*Internet of Things*) y múltiples usos más.

6.1. Procesamiento de medidas

Al contrario que en el Capítulo 5, para implementar los algoritmos de clasificación sí que se han tomado en consideración todas las medidas, tanto las primeras como las segundas, ante la no necesidad de un preprocesamiento con SAGE.

Esta suma de medidas totales han seguido además unas pautas diferentes a la hora de procesarlas, aunque los primeros pasos son comunes con el apartado de Clusterización, siendo la primera tarea la extracción del PDP. No obstante, esta recopilación de datos se ha procesado salvando las distancias de que las medidas parten de parámetros diferentes como el rango de frecuencias, entre otros.

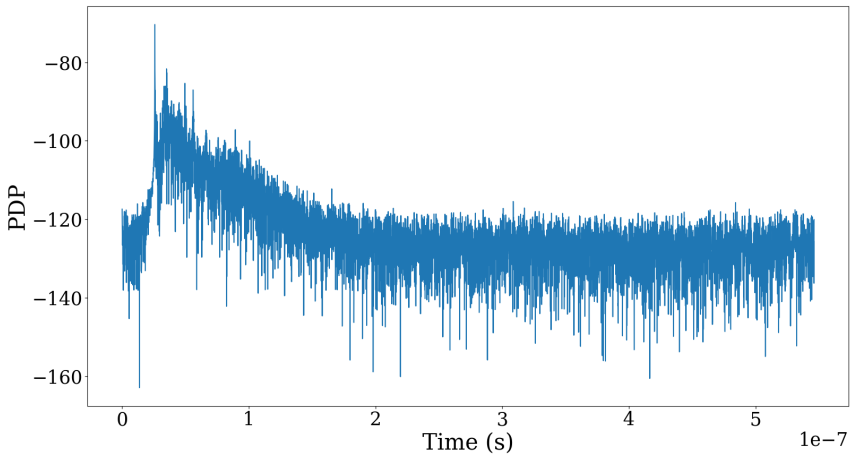

Figura 6.1: PDP.

El PDP representado en la Figura 6.1, tal y como está explicado en los aspectos técnicos, es una representación de las MPCs en potencia respecto al retardo con el que han sido recibidas. No obstante, para la programación y diseño de la NN y SVM, no será suficiente con esos datos, ya que para encontrar los patrones y las relaciones entre las muestras, se debe disponer de una cantidad suficiente de datos para su entrenamiento y el posterior proceso de testeo. Debido a esta falta de información, se extraerán numerosas características físicas propias del PDP, como son las expuestas en la Tabla 6.1.

Nombre Físico	**Significado Físico**
RMS Delay Spread	Mide la dispersión temporal de la señal. Un mayor valor indica una mayor dispersión de la energía de la señal en el tiempo.
Mean Delay	Indica el retardo medio de la señal.
Maximum Excess Delay	Indica el máximo retardo de la energía de un umbral, quedándose así con las MPCs significativas.
Número de Picos Sobre el Umbral	Número de picos del PDP que exceden un umbral específico.
Level Crossing Rate	Mide cuántas veces el PDP cruza un umbral dado, indicando la variabilidad y fluctuación de la potencia en el tiempo.
Amplitud del Primer Pico	Indica la mayor potencia de pico recibida de las MPCs.
Retardo del Primer Pico	Indica el retardo de la componente mayor en potencia.
Relación Primer a Segundo Pico	Da una idea de lo dominante que es la mayor componente en potencia comparado con el segundo mayor.
Área Bajo la Curva	Representa la energía total recibida, considerando la dispersión temporal de la señal.
Asimetría	La asimetría indica si la energía está más concentrada a la izquierda o a la derecha del pico principal.
Curtosis	La curtosis mide la acumulación de la distribución, indicando la presencia de trayectorias con retardos extremos.
Distribución de Energía 1	Proporciona una distribución de la energía en función del tiempo de retardo, mostrando cómo la energía de la señal se distribuye en diferentes intervalos de retardo.
Distribución de Energía 2	Proporciona una segunda distribución de la energía en función del tiempo de retardo, mostrando cómo la energía de la señal se distribuye en diferentes intervalos de retardo.

Tabla 6.1: Descripción de las características del PDP y su significado físico.

Posteriormente a la obtención del conjunto de *features*, se continuará con el procesamiento de los datos, pero ahora con la categorización. En la parte teórica del proyecto se explica que al tratarse de algoritmos de aprendizaje supervisado las muestras de entrenamiento deben estar correctamente marcadas si son LOS o NLOS . Para facilidad del algoritmo, directamente se ha marcado con notación binaria la clasificación, siendo “1” para LOS y “0” con NLOS . En la Figura 6.2 queda representado el *dataframe* de entrenamiento.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	rms_ds	mean_delay	max_ed	num_peaks_over_threshold	lcr	first_peak_amplitude	peak_delay	second_peak_ratio	auc	skewness_value	kurtosis_value	energy_dist_1	energy_dist_2	classification
2	3.633852287736489e-08	3.566401581574841 6e-08	8.906666666666667e-08	121	130	-73.04643856789804	2.413333333333334e-08	15.32936757449809	8.712688355324542e-18	83.13710811292957	7234.617796673723	0.9097251896017446	0.07380472183182461	1
3	3.77697648617737e-08	3.612339951874721 3e-08	7.033333333333334e-08	124	124	-73.38821191015685	2.413333333333334e-08	15.51659974414640 6	8.447798584625701e-18	80.49238996031768	6860.206879533196	0.905276452504946	0.0775229784819846	0

Figura 6.2: Ejemplo *dataframe* para entrenamiento.

Adicionalmente, para probar la eficacia de los algoritmos se deberá de poner a ensayo con nuevos datos no vistos hasta el momento y sin que estén categorizados, sino que sea el propio programa el que certifique si se está ante un PDP representativo de LOS o de NLOS, tal y como se muestra en la Figura 6.3.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	rms_ds	mean_delay	max_ed	num_peaks_over_threshold	lcr	first_peak_amplitude	peak_delay	second_peak_ratio	auc	skewness_value	kurtosis_value	energy_dist_1	energy_dist_2
2	3.309067184097108e-08	3.770112536283406e-08	6.606666666666666e-08	215	204	-73.34038258536539	2.586666666666667e-08	6.077489646925251	1.5035368519453896e-17	58.825819750845376	3652.8482698518646	0.9048189686253466	0.08066045424119399
3	3.1869581450983643e-08	3.780610227550887e-08	8.913333333333334e-08	218	188	-72.38640986905989	2.586666666666667e-08	6.227024428776688	1.5495780701211662e-17	65.19936631747751	4675.514446731285	0.9098202611631625	0.0764578834243335

Figura 6.3: Ejemplo de *dataframe* para prueba.

Debido a su gran versatilidad, los algoritmos basados en IA son capaces de ayudar a solucionar u optimizar múltiples problemas, actividades, procesos, etc., lo que desemboca en una desmesurada variedad de maneras de programarlos. Por tal razón, se ha decidido el uso de un *software* llamado *Orange Data Mining*. Este programa es utilizado para el análisis de datos complejos sin necesidad de haber programado previamente nada. Sus principales funciones son:

1. **Preprocesamiento de Datos**
   Incluye limpieza y preparación de datos, transformaciones y normalizaciones.

2. **Visualización de Datos**
   Gráficos interactivos como histogramas, diagramas de dispersión, diagramas de caja y gráficos de barras. Visualización de mapas de calor, árboles de decisión y redes neuronales.

3. **Análisis de Datos**
   Análisis exploratorio de datos, identificación de patrones y tendencias en los datos, clusterización y reducción de dimensionalidad.

4. **Modelado Predictivo**
   Creación y validación de modelos predictivos, algoritmos de clasificación y regresión, métodos de evaluación de modelos como validación cruzada y matrices de confusión.

5. **Minería de Textos**
   Análisis de texto y minería de textos, extracción de características de documentos y análisis de contenido.

6. **Integración y Extensibilidad**
   Integración con otros lenguajes y herramientas de programación como *Python*, posibilidad de añadir *plugins* y *scripts* personalizados ampliando su funcionalidad.

7. **Automatización**
   Creación de flujos de trabajo automatizados para el análisis de datos, permite arrastrar y soltar widgets para construir pipelines de análisis de datos sin escribir código.

Tales cualidades del *software* ayudarán en una primera estimación de cómo se comportarían dependiendo de qué algoritmos y con qué parametros cada uno ante los datos de entrenamiento,

poniéndolos a su vez a prueba con los datos de evaluación. Para tal fin, se ha diseñado el esquema mostrado en la Figura 6.4.

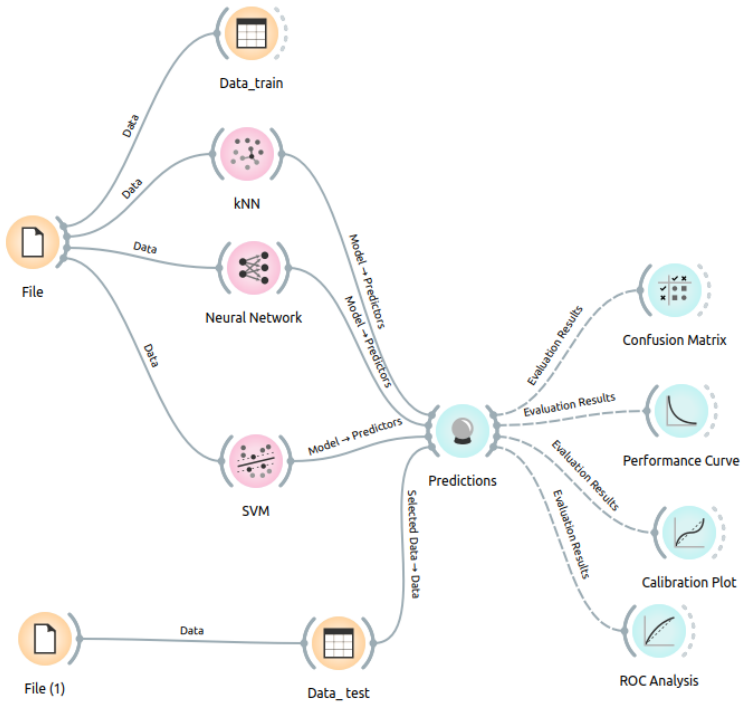

Figura 6.4: Esquema del diseño en Orange Data Mining.

Los resultados que arrojaron fueron realmente buenos para los dos algoritmos que se han finalmente programado, en los que el porcentaje de acierto se acercaba al 98 %. Sin embargo, el tercer método, denominado k-NN (*K-Nearest Neighbors*) no resultaría ser el más eficaz.

6.2. Algoritmos para Clasificación

Tal y como se viene destacando a lo largo de la memoria, los dos algoritmos empleados para la clasificación son SVM y las NN.

Además, una vez aplicados ambos procesos, se realizará una comparación entre ambos con el fin de determinar las ventajas y desventajas de cada uno, terminando por la conclusión de la mejor opción para el trabajo. Para dicha razón se han utilizado métodos como la matriz de confusión, diagrama de pesos, etc.

6.2.1. SVM

A la hora de programar SVM se debe encontrar el margen de decisión óptimo, dependiente del tipo de *kernel* utilizado y del valor del hiperparámetro C, que recordando lo anteriormente explicado,

relaciona el ancho del margen de seguridad entre ambas clases y el número de muestras clasificadas incorrectamente. En este caso y acorde con el *feedback* proporcionado por *Orange Data Mining* los parámetros han sido los proyectados en la Tabla 6.2.

Kernel	Lineal
Parámetro C	0.01

Tabla 6.2: Características SVM.

Una vez ejecutado el código de *Python* de la máquina de vectores, comenzará el análisis de los datos y de las predicciones propuestas por el programa. Primero, en la Figura 6.5 se representa el mapa de calor que permitirá visualizar la importancia relativa de cada parámetro obtenido del PDP en la predicción del modelo SVM, con una escala de colores que oscila entre 0 y 1 y donde los tonos azulados representan valores de coeficientes más bajos mientras que los tonos rojizos valores más altos.

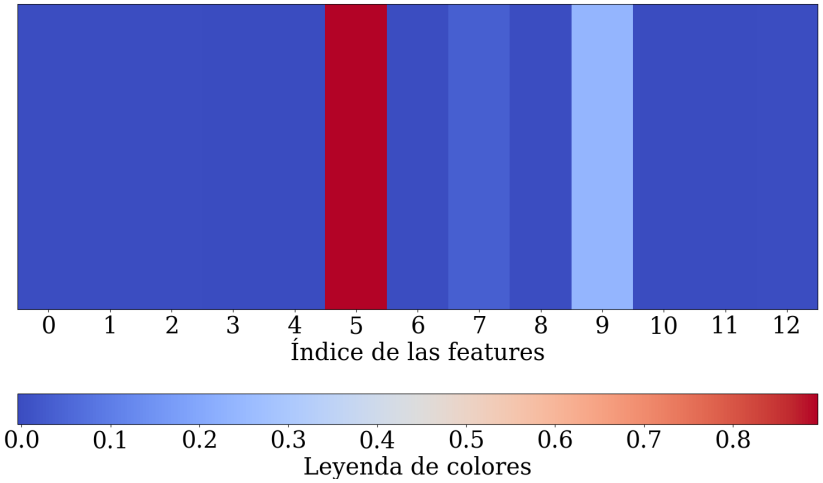

Figura 6.5: Mapa de calor de las *features* de SVM.

Al evaluar el gráfico de la Figura 6.5, se detecta fácilmente que la característica enumerada como 5 tiene un coeficiente significativamente más alto, representado en color rojo. Esto sugiere que dicha característica tiene un impacto considerablemente mayor en la predicción del modelo SVM en comparación con las demás, lo que implica una alta determinación en el proceso de clasificación. Siguiendo la numeración de la Tabla 6.1, esta *feature* correspondería con la métrica que define la amplitud del pico más alto.

En contraste, las características ubicadas en los índices 0, 1, 2, 3, 4, 6, 8, 10, 11 y 12 presentan coeficientes relativamente bajos, definidos por tonos parecidos de azul e indicando su menor influencia sobre las decisiones. Los parámetros en los índices 9 (Asimetría) y 7 (Relación primer-segundo pico) sí que parecen tener un aumento moderado en su coeficiente, representados por tonos más claros y expresando así una importancia intermedia en el modelo.

El rango del conjunto de los coeficientes varía de aproximadamente 0.1 a 0.9, lo que indica una gran dispersión en los valores asignados a cada característica. Esta variación apunta a que no todas contribuirán de manera equitativa a las predicciones, como ya se ha explicado detalladamente.

Esta información podría ser muy útil para llevar a cabo una selección de *features* con el fin de reducir la dimensionalidad del modelo y su complejidad computacional, enfocándose exclusivamente en aquellas con coeficientes más altos.

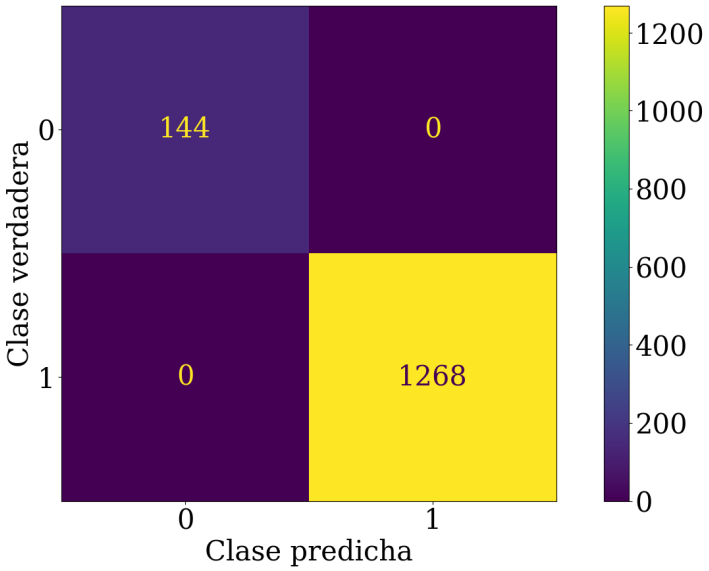

Figura 6.6: Matriz de confusión de SVM.

La imagen presentada en la Figura 6.6 es lo denominado matriz de confusión, utilizada para evaluar el rendimiento de un modelo de clasificación. En este caso, la matriz de confusión dispone de dos clases: la clase 0 (NLOS) y la clase 1 (LOS). Los elementos en la matriz representan el número de predicciones correctas e incorrectas hechas por el modelo comparadas con los valores verdaderos de las etiquetas.

Se observa que el modelo predice correctamente 144 veces la clase 0, además de clasificar correctamente 1268 veces la clase 1. Por tanto, se afirma que el algoritmo no produce error alguno en las resultados, es decir, no hay falsos positivos ni falsos negativos.

En las métricas derivadas de esta matriz de confusión quedan reflejados:

1. **Precisión (*Precision*)**: La precisión indica la proporción de verdaderos positivos entre el total de muestras clasificadas como positivas por el modelo.

2. **Exhaustividad (*Recall*)**: El *recall*, comúnmente conocido como sensibilidad, mide la proporción de verdaderos positivos entre el total de muestras verdaderamente positivas.

3. **Puntuación F1 (*F1-Score*)**: La puntuación F1 representa la media armónica de la precisión y el *recall*. Esta métrica ofrece un equilibrio entre ambos, proporcionando una medida única de rendimiento que considera tanto la exactitud del modelo en las predicciones positivas como su capacidad para identificar todas las instancias positivas.

4. **Soporte (*Support*)**: El soporte se refiere al número de muestras reales de cada clase en el conjunto de datos. Añade información sobre la distribución de las clases.

5. **Exactitud (*Accuracy*)**: La exactitud global del modelo mide la proporción de predicciones correctas entre el total de instancias. Sin embargo, con datos desbalanceados, puede no llegar a ser del todo representativa del funcionamiento del modelo.

6. **Promedio Macro (*Macro Avg*)**: La media de las métricas de precisión, *recall* y *F1-score* calculadas de manera independiente para cada clase y luego promediadas. Esta métrica trata a todas las clases por igual, sin importar su proporción en el conjunto de datos, siendo práctica en situaciones donde las clases tienen diferentes tamaños, como es el caso.

7. **Promedio Ponderado (*Weighted Avg*)**: La media ponderada de las métricas de precisión, *recall* y *F1-score*, donde el peso de cada clase se basa en su proporción en el conjunto de datos, considerando así la distribución real de las clases.

6.2.1.1. Ajuste de hiperparámetros

Se identifica fácilmente que los resultados arrojados por SVM con los hiperparámetros escogidos inicialmente podrían definirse como excelentes, si no fuera porque en los algoritmos de aprendizaje automático el hecho de tener un porcentaje de acierto del 100 % puede indicar *overfitting* y poco aprendizaje, siendo poco común en la práctica. La tendencia por tanto es en su mayoría a tener algún grado de equivocación.

En búsqueda de la opción que mejor se ajuste a los datos, se llevará a cabo un amplio estudio del comportamiento del algoritmo programado con el cambio de los hiperparámetros que lo definen: el *kernel* utilizado y el parámetro C.

Esta revisión consistirá en un primer análisis de las gráficas de validación [22], las cuales habiéndose establecido el *kernel* y dependiendo del número C, indicarán qué tan preciso es el entrenamiento (representando el aprendizaje con muestras vistas en el propio entrenamiento) y la prueba con validación cruzada (con datos extraídos aleatoriamente del conjunto de entrenamiento). A continuación se mostrarán en una tabla las métricas más significativas que ofrece SVM para valores más concretos de C, pudiendo aportar una perspectiva más exacta del comportamiento del método. Como últimos pasos, y tras el análisis de la gráfica y de la tabla anteriores, se tratará de escoger la o las opciones más adecuadas, aportando posteriormente la curva de aprendizaje [23] del modelo seleccionado, que representará la evolución de la puntuación de SVM dependiendo del número de muestras de entrenamiento y facilitando la información para poder optimizar posteriormente el diseño en tiempo y computación.

En ambas gráficas, la de validación y de aprendizaje, se representarán valores medios con la línea más destacada y sus desviaciones típicas, con el sombreado a lo largo de la curva. Destacar que ambos valores se situarán entre valores de 0 y 1.

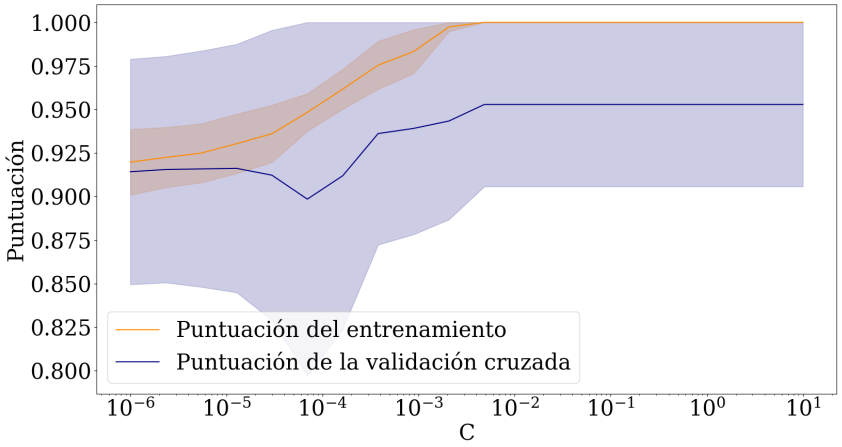

Figura 6.7: Curva de validación con *kernel* lineal.

C	Precision	Recall	F1-Score	Support
0.0001	0.99	0.99	0.99	1412
0.001	0.99	0.99	0.99	1412
0.01	1.0	1.0	1.0	1412
0.1	1.0	1.0	1.0	1412
1.0	1.0	1.0	1.0	1412
10.0	1.0	1.0	1.0	1412
100.0	1.0	1.0	1.0	1412

Tabla 6.3: Resultados de SVM para el *kernel* lineal.

Estudiando el rendimiento de SVM con un *kernel* lineal, mostrado en la Figura 6.7 y Tabla 6.3, la perfección sigue destacando en los resultados, al menos en 5 de los 7 valores de prueba. Una vez descartados estos valores ”perfectos” como medida de seguridad frente al *overfitting*, las opciones sobrantes representan una ligera mejoría en el funcionamiento del algoritmo.

Sí que cabría destacar que con un valor de C=0.0001, tal y como se muestra en la Tabla 6.3, se obtienen fallos en ambas clases, que aunque en pequeña cantidad, ocurre al contrario que en el modelo con C=0.001, donde exclusivamente falla en la predicción de una clase, mostrando un posible sesgo. Además, con C=0.0001 el diagrama de pesos en el que se refleja la importancia de cada *feature* indica la participación de más características del PDP en las decisiones, lo que supondría un punto interesante de investigación para futuros trabajos.

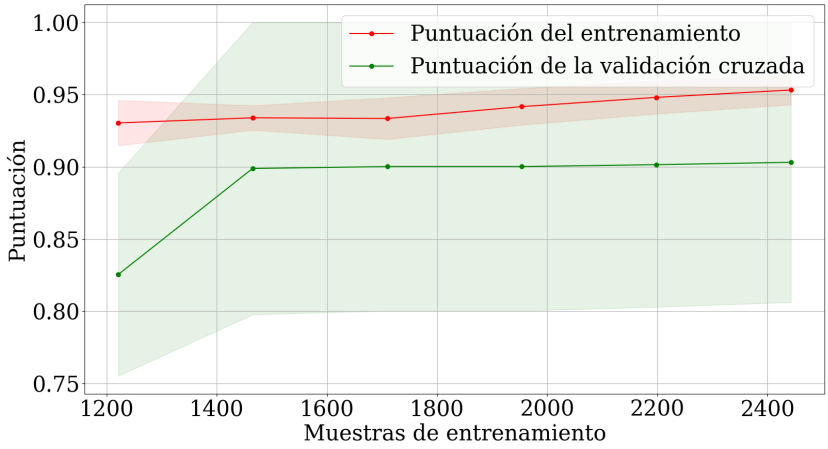

Figura 6.8: Mapa de calor de las *features* con C=0.0001.

Gracias a la Figura 6.8 se afirma que a partir de 1500 muestras de entrenamiento, la mejora no es significativa, lo que podría determinar un punto para futuras mejoras en busca de la eficiencia computacional.

Al haber modificado los hiperparámetros, otra gráfica que también depende del número C utilizado es el mapa de calor de las *features*. En la Figura 6.9 se referencian unos pesos que ahora son distintos a los establecidos por los hiperparámetros por defecto. Estos pesos tienen la novedad de dotar de mayor importancia a más características que los expresados en la Figura 6.5.

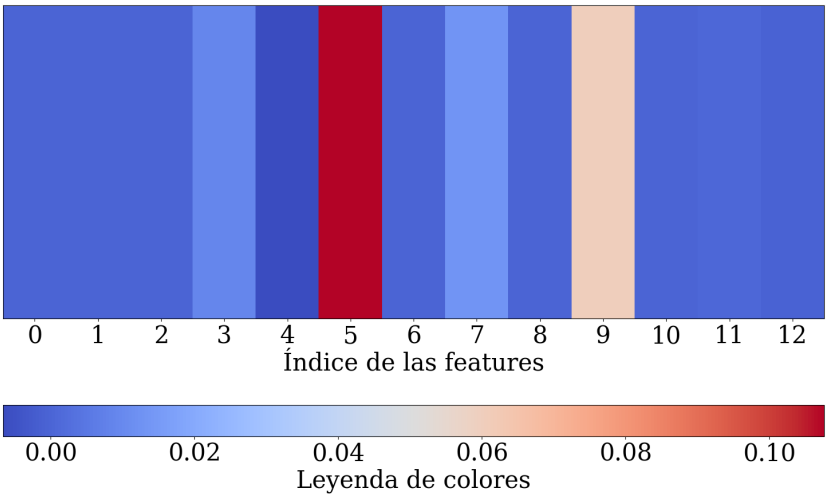

Figura 6.9: Mapa de calor de las *features* con C=0.0001.

Continuando con el siguiente tipo de *kernel*, en este caso el polinómico, se obtendrán los siguientes resultados representados en la Figura 6.10 y en la Tabla 6.4:

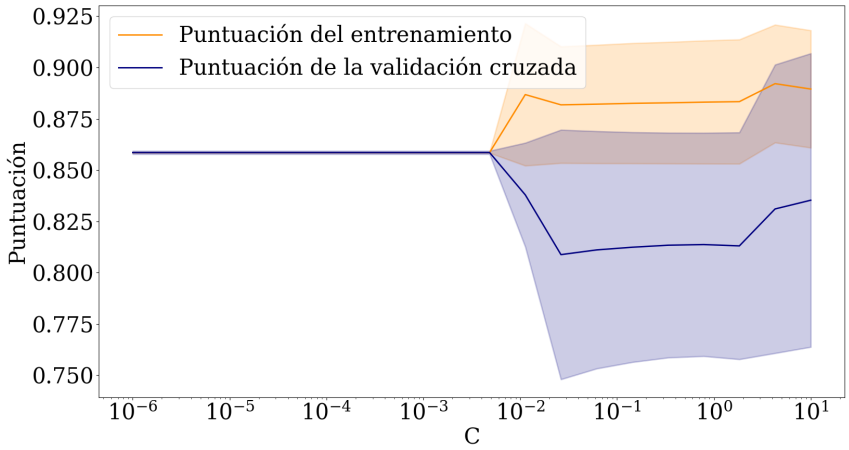

Figura 6.10: Curva de validación con *kernel* polinómico.

C	Precision	Recall	F1-Score	Support
0.0001	0.81	0.90	0.85	1412
0.001	0.81	0.90	0.85	1412
0.01	0.81	0.90	0.85	1412
0.1	0.97	0.96	0.96	1412
1.0	0.97	0.96	0.96	1412
10.0	0.96	0.96	0.96	1412
100.0	0.96	0.96	0.96	1412

Tabla 6.4: Resultados de SVM para el *kernel* polinómico.

En esta modalidad de *kernel* polinómico se ve un fenómeno extraño que para futuras líneas sería interesante estudiar más en profundidad, y es que con el crecimiento del hiperparámetro C y según la curva de validación, se reducen las expectativas de puntuación en la validación cruzada, con una extracción de datos de entrenamiento para probar. Sin embargo, a la hora de la evaluación real con otras muestras independientes, las métricas mejoran, tal y como se muestra en la Tabla 6.4.

Dejando este hecho interesante como punto de estudio futuro y prestando atención a las métricas más exactas de la tabla, el mejor resultado se recogería con el parámetro C=10, que resulta en unos fallos equilibrados para ambas clases, siendo un 5.55 % para la clase 0 y un 4.41 % en la clase 1.

Las primeras 3 opciones de la Tabla 6.4 ni se contemplan como posibles soluciones, ya que devuelven unos resultados que pueden ser calificados como pésimos al estar fallando en el 100 % de las predicciones sobre la clase 0.

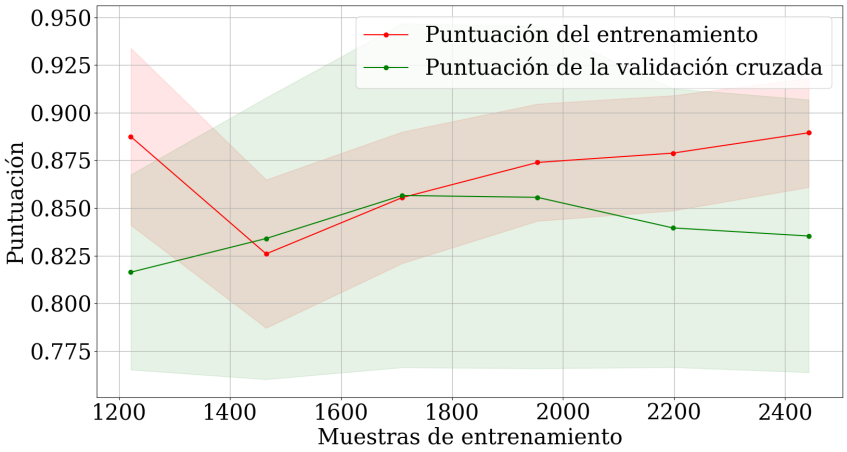

Figura 6.11: Curva de aprendizaje con *kernel* polinómico y C=10.

En la Figura 6.11 entre 1800 y 2000 muestras se recogen los mejores resultados si se busca un balance entre la puntuación del entrenamiento y la de la validación cruzada.

Prosiguiendo con el estudio, ahora se pondrán a prueba también modelos con el *kernel RBF*, los cuales devolverán unos resultados diferentes a los vistos anteriormente con otros diseños. Estos están representados en la Figura 6.12 y en la Tabla 6.5.

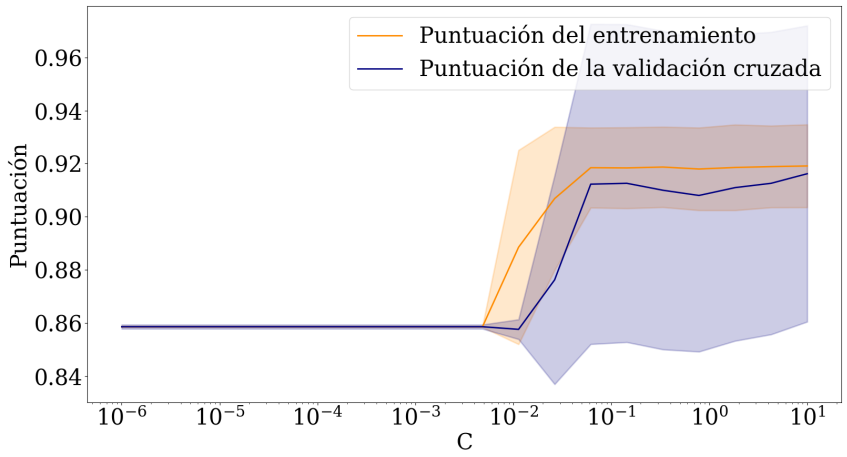

Figura 6.12: Curva de validación con *kernel RBF*.

C	Precision	Recall	F1-Score	Support
0.0001	0.81	0.9	0.85	1412
0.001	0.81	0.90	0.85	1412
0.01	0.81	0.90	0.85	1412
0.1	0.95	0.95	0.95	1412
1.0	0.95	0.95	0.95	1412
10.0	0.95	0.95	0.95	1412
100.0	0.95	0.95	0.95	1412

Tabla 6.5: Resultados de SVM para el *kernel RBF*.

Habiendo indagado más en detalle sobre este tipo de SVM con un *kernel RBF* se encuentra que ninguna de las opciones sería conveniente para el objetivo, debido al sesgo demostrado en los resultados. Estos diseños presentan un sesgo total en las tres primeras opciones de la tabla y parcial en las restantes.

En cuanto al sesgo total se hace referencia a la situación en la que falla completamente en las predicciones de una clase, mientras que en el parcial se refiere a un desbalanceo en el porcentaje de fallos en una clase respecto de la otra.

A modo de ejemplo para poder entender la situación, con un número C=0.1, se tendría un 43.75 % de errores en la clase 0 y un 0.78 % en la clase 1. Con un C=1, la clase 0 devuelve unos fallos representando el 48.6 % y la 1 un 0.39 %.

En resumen, el *kernel RBF* no es una solución óptima para esta situación, aunque aclarando que con más o menos datos, o diferentes, sí que podría ser una buena opción. Por ello, no se adjuntará la curva de aprendizaje.

Los resultados arrojados por los modelos con un *kernel* sigmoide están representados en la Figura 6.13 y en la Tabla 6.6.

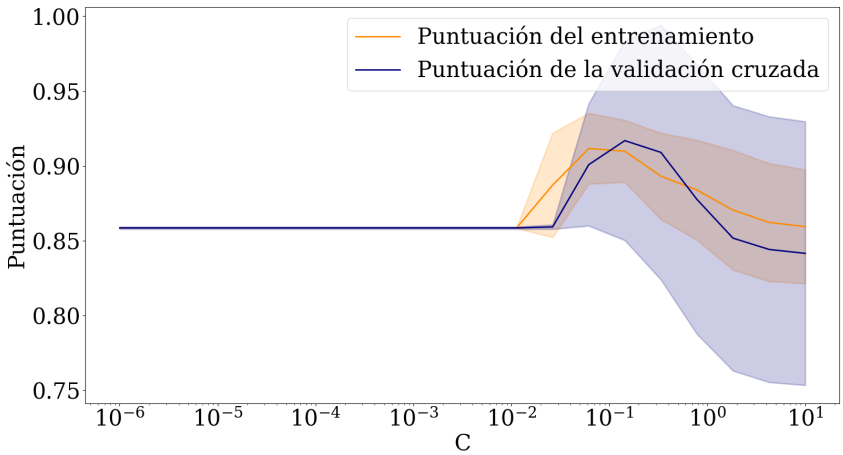

Figura 6.13: Curva de validación con *kernel* sigmoide.

C	Precision	Recall	F1-Score	Support
0.0001	0.81	0.9	0.85	1412
0.001	0.81	0.9	0.85	1412
0.01	0.81	0.9	0.85	1412
0.1	0.95	0.95	0.95	1412
1.0	0.96	0.96	0.96	1412
10.0	0.95	0.95	0.95	1412
100.0	0.95	0.95	0.95	1412

Tabla 6.6: Resultados de SVM para el *kernel* sigmoide.

En esta última prueba se descartan los 3 primeros valores de C (0.0001 / 0.001 / 0.01) al seguir los mismos criterios que en anteriores *kernels*, ya que cuando la precisión ronda el valor de 0.8 se obtienen un 100 % de fallos de la clase 0, demostrando un porcentaje de error significativo además de un sesgo importante.

Analizando los valores del resto de opciones, se reconocen resultados con un alto grado de similitud, por lo que se tendrá que interiorizar más en las predicciones para poder distinguir la mejor opción. Para este objetivo, se han vuelto a examinar las matrices de confusión de cada opción de valores del parámetro C, viendo a su vez no el número de fallos de cada clase, sino el porcentaje de cada una.

Los resultados arrojados indican que la mejor opción de SVM es la parametrizada con un valor de C=10, gracias al cual se tendrían un 7.63 % de errores en la clase 0 y un 4.9 % en la clase 1, dando los resultados más equilibrados y teniendo en cuenta el tamaño de las muestras.

Como añadido, en la Figura 6.14 se representa la que sería la curva de aprendizaje tras haber escogido el parámetro C óptimo.

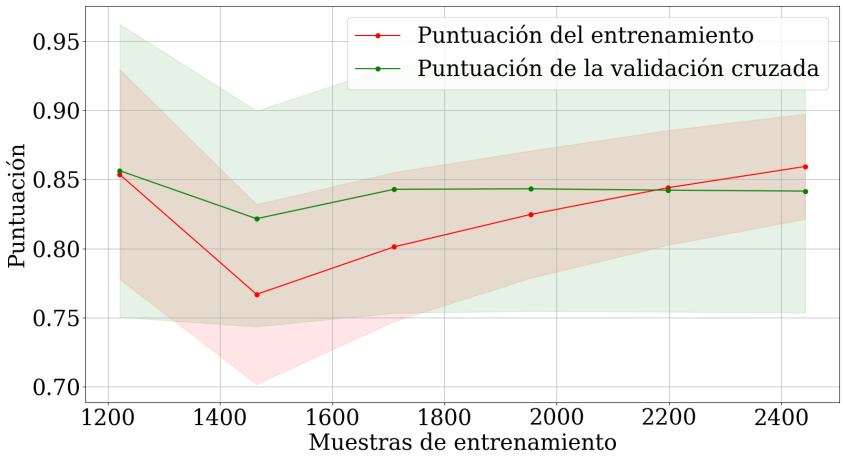

Figura 6.14: Curva de aprendizaje con *kernel* sigmoide y C=10.

Siguiendo las tendencias en anteriores modelos, con un *kernel* sigmoide también se puede concretar que cantidades mayores de 2000 datos en la toma de muestras son las más eficaces, pero las mejoras a partir de tal número no conllevan aumentos distinguidos en la métrica. Además, a partir de esa cifra se reducen los márgenes establecidos por la desviación típica, lo que puede aportar una cierta seguridad respecto al correcto funcionamiento del sistema.

6.2.2. Redes Neuronales

El diseño de las redes neuronales ha conllevado mayor tiempo tanto computacional como de documentación para poder crear una herramienta eficaz y adaptada a los datos. Como añadido, se parte de la cuestión de que por lo general este tipo de algoritmos suelen requerir una gran cantidad de muestras, mucho mayor a la que se dispone para este estudio.

Al igual que en la optimización de SVM, se ha pretendido probar diferentes configuraciones de redes neuronales para obtener así una visión más general del funcionamiento de estas y ver sus puntos de mejora para futuras líneas. En este caso, los hiperparámetros modificados han sido el número de neuronas de la capa oculta y el *learning rate*, tal y como se muestra en la Tabla 6.7.

Rango de neuronas	8	10	15	20	25	32
Learning Rate	0.001	0.002	0.003	0.004	0.005	

Tabla 6.7: Rango hiperparámetros NN.

El motivo de comenzar el diseño y su proceso de ajuste con un determinado número de capas ocultas, en este caso una, viene dado por la primera visión general que aporta de nuevo *Orange Data Mining*. Respecto a la cantidad de neuronas de la capa de entrada o *input layer*, corresponde al número de *features*, que coinciden con las utilizadas en SVM. La capa de salida u *output layer* dispone de una sola neurona, al tratarse de un problema de clasificación binaria.

Respecto a los datos de entrenamiento, hay que destacar un desbalance significativo. Este desbalance se manifiesta en una mayor cantidad de muestras pertenecientes a la clase LOS en comparación con la clase NLOS , pudiendo tener un impacto considerable en el rendimiento del modelo.

6.2.2.1. Ajuste de hiperparámetros

Las distintas pruebas o entrenamiento de los modelos, tal y como se comenta previamente, ha supuesto un elevado gasto temporal para el procesamiento computacional de las redes, por lo que el tratamiento de los datos ha sido extremadamente riguroso con el fin de evitar repeticiones innecesarias del entrenamiento.

Además, con el fin de aumentar el espectro del estudio se han establecido dos conjuntos de muestras diferentes para el proceso de entrenamiento de las NN. El primero de ellos constará exclusivamente de datos provenientes de las medidas clasificadas en el Capítulo 4 como ”primeras medidas” y conformadas por un 72.22 % de muestras pertenecientes a LOS y un 27.78 % a NLOS . No obstante, para el segundo grupo de datos se mantendrán ambos porcentajes pero se añadirá una función comúnmente utilizada en las redes neuronales. Todo ello añadido a que la evaluación se

realizará con una mezcla de muestras de las recopiladas en la sala de becarios, así como las medidas en el laboratorio; añadiendo así dificultad al poner a prueba el algoritmo ante medidas en entornos nunca vistos previamente.

- **Entrenamiento con primeras medidas**

  Los modelos diseñados con las siguientes opciones de hiperparámetros, han devuelto unos resultados como los representados en las Tablas 6.8, 6.9, 6.10, 6.11, 6.12 y 6.13.

Learning Rate	Porcentaje de error NLOS	Porcentaje de error LOS
0.001	100 %	0 %
0.002	100 %	0 %
0.003	100 %	0 %
0.004	100 %	0 %
0.005	100 %	0 %

Tabla 6.8: Resultados ajuste de NN para 8 neuronas.

Learning Rate	Porcentaje de error NLOS	Porcentaje de error LOS
0.001	7.6 %	0.9 %
0.002	100 %	0 %
0.003	100 %	0 %
0.004	100 %	0 %
0.005	100 %	0 %

Tabla 6.9: Resultados ajuste de NN para 10 neuronas.

Learning Rate	Porcentaje de error NLOS	Porcentaje de error LOS
0.001	0 %	7.3 %
0.002	100 %	0 %
0.003	100 %	0 %
0.004	100 %	0 %
0.005	100 %	0 %

Tabla 6.10: Resultados ajuste de NN para 15 neuronas.

Learning Rate	Porcentaje de error NLOS	Porcentaje de error LOS
0.001	0 %	11.9 %
0.002	100 %	0 %
0.003	100 %	0 %
0.004	100 %	0 %
0.005	100 %	0 %

Tabla 6.11: Resultados ajuste de NN para 20 neuronas.

Learning Rate	Porcentaje de error NLOS	Porcentaje de error LOS
0.001	0 %	9.54 %
0.002	100 %	0 %
0.003	3.4 %	7.9 %
0.004	100 %	0 %
0.005	100 %	0 %

Tabla 6.12: Resultados ajuste de NN para 25 neuronas.

Learning Rate	Porcentaje de error NLOS	Porcentaje de error LOS
0.001	0 %	9.3 %
0.002	100 %	0 %
0.003	100 %	0 %
0.004	100 %	0 %
0.005	100 %	0 %

Tabla 6.13: Resultados ajuste de NN para 32 neuronas.

Como análisis de los resultados expuestos, salvo en ciertos modelos, se presencia un claro ejemplo del impacto del sesgo hacia la clase mayoritaria, producido por el desbalanceo explicado anteriormente. En términos numéricos y relativo a los resultados de la red neuronal, el efecto del sesgo puede conllevar una alta tasa de aciertos para la clase LOS pero baja para la clase NLOS , y eso es precisamente lo devuelto por los diseños.

Enfocándose en las funciones de pérdidas, sucedería una minimización principalmente por los pocos errores en la clase mayoritaria, ignorando los errores en la clase minoritaria al no representar un porcentaje significativo respecto del total de muestras. De la misma manera, este efecto se puede dar también en las gráficas de precisión con niveles realmente altos, lo que significaría un valor inusual.

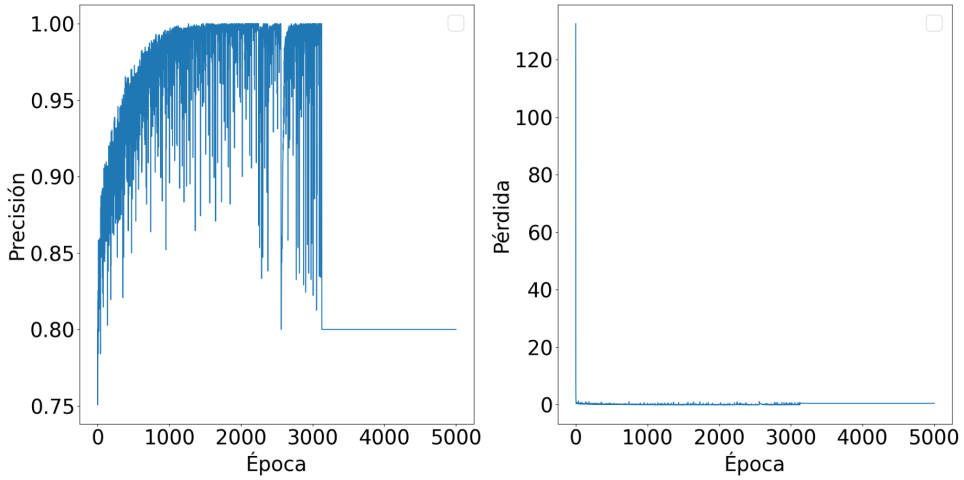

Figura 6.15: Gráficas de precisión y pérdidas del modelo con 8 neuronas y lr= 0.002.

Además, en la Figura 6.15 se identifica un problema potencialmente derivado del desajuste proporcional entre ambas clases: la finalización del aprendizaje. Este hecho está representado en la total estabilización de la gráfica de precisión a partir de un cierto número de épocas, en las que el algoritmo ya no aprende de ninguna de los datos de ninguna de las dos clases. No es el único ejemplo, a continuación se muestra otro en la Figura 6.16.

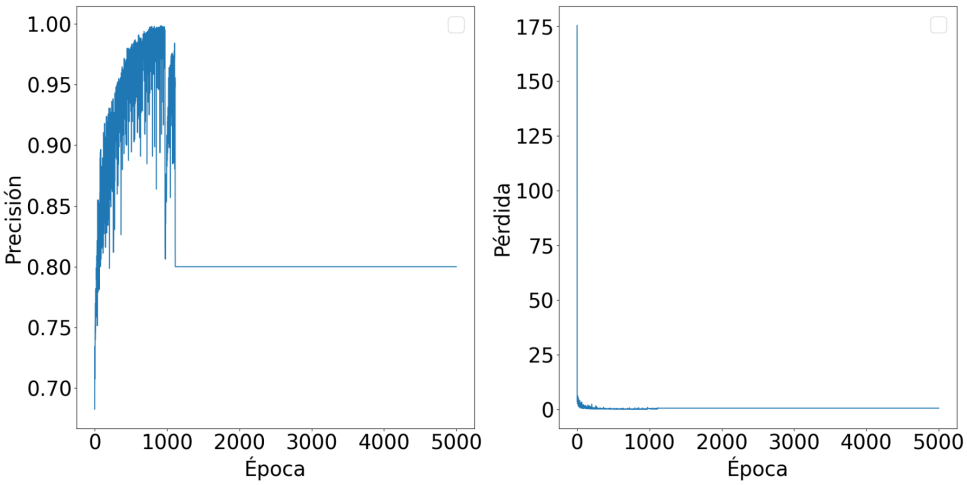

Figura 6.16: Gráficas de precisión y pérdidas del modelo con 8 neuronas y lr= 0.004.

Sumado al inconveniente del desbalanceo de datos se identifica otra cuestión a tener en cuenta y relacionada con la eficiencia del algoritmo: el número de épocas. En el diseño de las NN se han establecido 5000 épocas a modo de orientación para poder observar el comportamiento del método con el aumento de los bucles realizados en su entrenamiento. Contrariamente a lo que posiblemente fuera lo más lógico, con el paso de las épocas, el hecho de haber realizado más bucles en busca de patrones y relaciones no está relacionado directamente con una mejora del algoritmo, sino que hay ocasiones, como las mostradas en las Figuras 6.15 y 6.16, en las que se reduce la precisión en una importante cantidad, empeorando el desempeño de la red. Precisamente, las únicas dos gráficas de precisión que

no rebajan sus expectativas a partir de un número de épocas, las representadas en las Figuras 6.17 y 6.18, son las que mejores resultados aportan.

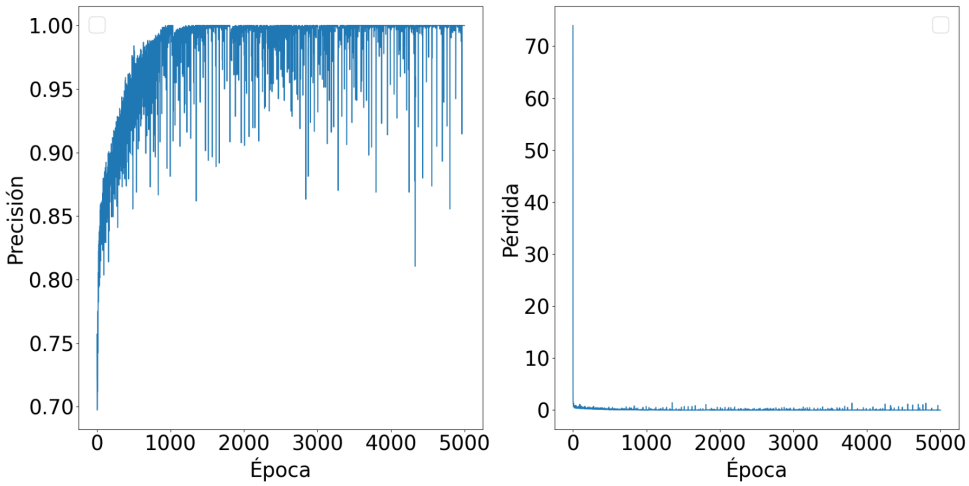

Figura 6.17: Gráficas de precisión y pérdidas del modelo con 15 neuronas y lr= 0.001.

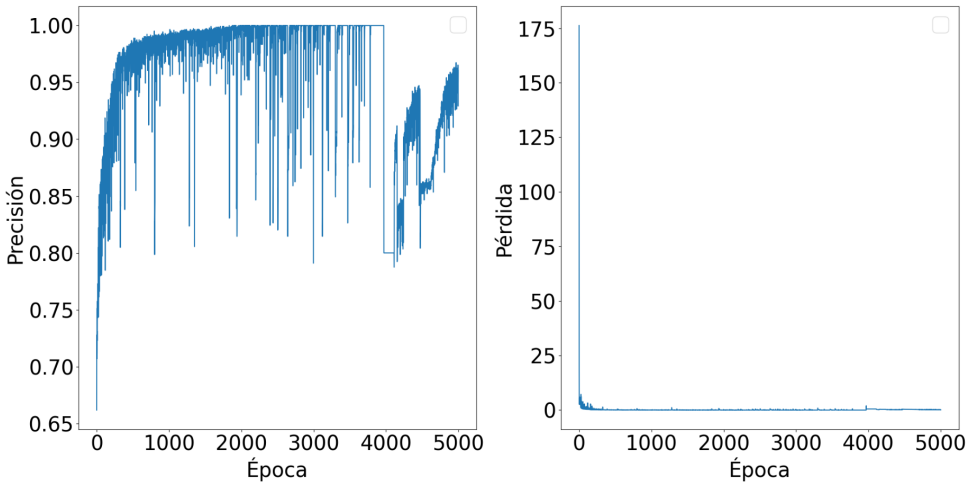

Figura 6.18: Gráficas de precisión y pérdidas del modelo con 25 neuronas y lr= 0.003.

Es por ello por lo que se ha utilizado una herramienta denominada *EarlyStopping*. El mecanismo utilizado por la función está basado en la comparación de una métrica escogida a lo largo de un número determinado de épocas. En caso de no detectar mejoras en la métrica, la ejecución de los bucles se detendrá.

En este caso, la métrica escogida para tener en cuenta ha sido la precisión, aunque se podrían establecer otras como las pérdidas o personalizarlas para detener la ejecución cuando lleguen a ciertos niveles, etc. El número de épocas en las que se tienen que mantener o no mejorar los valores de precisión se ha establecido como 150, escogido tras numerosas pruebas en busca de una mejora en los resultados.

- **Entrenamiento con medidas primeras y función *EarlyStopping***

  Los modelos que se han destacado en color verdáceo son los que mejores resultados devuelven tras su entrenamiento y proceso de evaluación con muestras no vistas anteriormente y en entornos distintos al entrenado.

  Para un análisis más cercano a cada modelo, se mostrarán las gráficas de pérdidas y de precisión, como anteriormente, y se estudiará cuán influyente es cada parámetro del PDP en la resolución final, similarmente al diagrama de pesos de SVM. Estos últimos gráficos se han obtenido gracias a la librería importada SHAP (*Shapley Additive Explanations*) [24].

  En referencia a los resultados obtenidos por los modelos entrenados con 8 neuronas en su capa oculta, se muestran la Tabla 6.14 y la Figura 6.19.

Learning Rate	Porcentaje de error NLOS	Porcentaje de error LOS
0.001	0.69 %	14.74 %
0.002	0 %	32.01 %
0.003	0.69 %	9.7 %
0.004	1.38 %	1.26 %
0.005	0 %	3.23 %

Tabla 6.14: Resultados ajuste de NN para 8 neuronas con *EarlyStopping*.

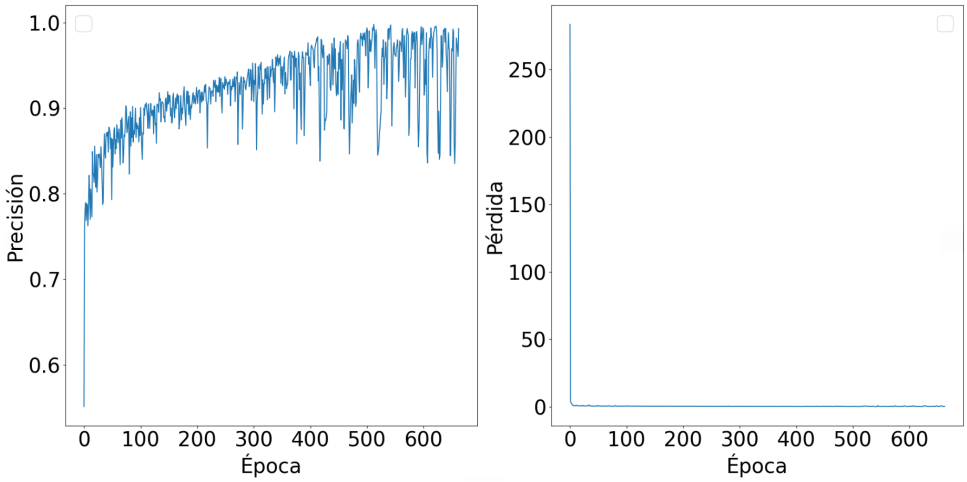

Figura 6.19: Gráficas de precisión y pérdidas del modelo con 8 neuronas, lr= 0.004 y *EarlyStopping*.

En cuanto a la gráfica de precisión de la Figura 6.19, se observa un claro crecimiento en las primeras épocas. Esto indica que el modelo está aprendiendo adecuadamente de los datos y mejorando su capacidad para hacer predicciones correctas. Sin embargo, a medida que el entrenamiento avanza, la precisión empieza a mostrar una serie de fluctuaciones, lo que sugiere que el modelo podría estar experimentando sobreajuste, donde posiblemente esté

aprendiendo demasiado los detalles específicos del conjunto de entrenamiento, lo que reducirá su capacidad de generalizar bien a datos no vistos.

En la gráfica de las pérdidas de la Figura 6.19, se identifica una rápida disminución de la pérdida en las primeras etapas del entrenamiento, lo inverso al aumento de la precisión. Esta reducción de las pérdidas es habitual cuando el modelo comienza a ajustar sus pesos para minimizar la función de coste. Sin embargo, la forma de la gráfica indica que alcanza una estabilidad demasiado rápido y sin fluctuaciones, permaneciendo constante en la mayoría del entrenamiento, lo que no transmite una seguridad en el modelo.

Además, gracias al funcionamiento de EarlyStopping se ha detenido el entrenamiento antes de que se produzca un sobreajuste severo.

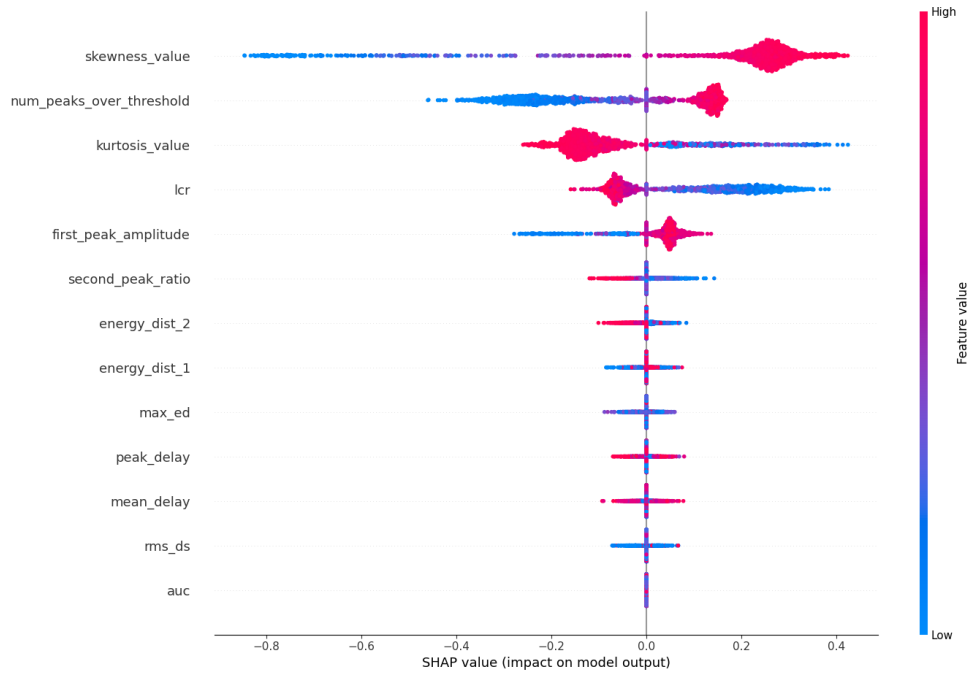

Figura 6.20: Gráfica de influencia en los parámetros en las predicciones del modelo on 8 neuronas, lr= 0.004 y *EarlyStopping*.

Estudiando el impacto de cada *feature*, mostrado en la Figura 6.20, se puede observar puntos de diferentes colores a lo largo de un eje horizontal, representando valores positivos o negativos. Cada punto representa un valor de esa característica en cada una de las muestras y sigue el esquema de colores indicado en la derecha. Además, cada punto está posicionado a lo largo del eje horizontal dependiendo del valor SHAP, indicando un impacto a favor de la clase LOS si tiene asignado valores positivos o impacto negativo para LOS, o lo que es lo mismo, impacto positivo a favor de NLOS .

Al observar la gráfica, se puede notar que la característica *skewness* tiene un impacto positivo considerable en las predicciones del modelo cuando sus valores son altos. Sin embargo, cuando los valores empiezan a disminuir, se dispersan por el eje negativo indicando un impulso hacia la clase NLOS . La característica representativa del número de picos por encima del umbral sigue un patrón similar.

Por otro lado, las características *lcr* y *curtosis* destacan por seguir el mismo patrón que las dos comentadas previamente, pero al contrario, favoreciendo valores altos a la clase NLOS y los valores más bajos a LOS.

En cuanto a la amplitud del pico más alto, se identifica un impacto mixto. Los valores bajos de esta característica tienen un impacto negativo, mientras que los valores altos tienen un impacto positivo. Sin embargo, la influencia general parece ser moderada en comparación con las características anteriormente mencionadas. Las características restantes no demuestran tener una gran influencia en los resultados.

Los modelos entrenados con 10 neuronas en su capa oculta aportan unos resultados representados en la Tabla 6.15 y en la Figura 6.21.

Learning Rate	Porcentaje de error NLOS	Porcentaje de error LOS
0.001	0 %	4.49 %
0.002	0 %	47.79 %
0.003	0 %	8.20 %
0.004	0 %	19.24 %
0.005	0.69 %	10.64 %

Tabla 6.15: Resultados ajuste de NN para 10 neuronas con *EarlyStopping*.

Revisando los porcentajes de errores, se pondrá como objeto de estudio al modelo con un 4.49 % de errores en LOS y 0 % en NLOS . Los demás han sido descartados por sus mayores porcentajes de errores, aunque como siempre, aclarar que con un diferente campo de datos sí que podrían suponer buenos modelos.

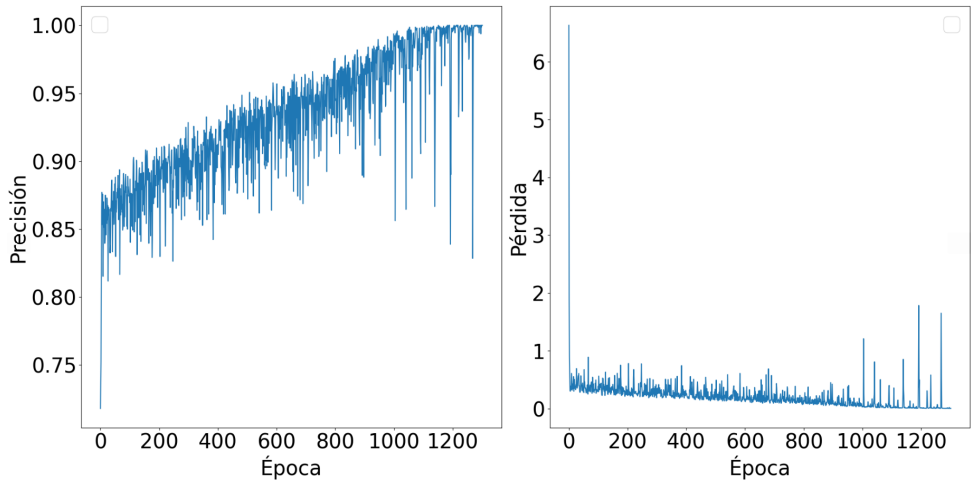

Figura 6.21: Gráficas de precisión y pérdidas del modelo con 10 neuronas, lr= 0.001 y *EarlyStopping*.

Durante las épocas se ve una tendencia ascendente en la precisión, como en anteriores gráficas, alcanzando casi el 100 % hacia la época 1200, lo que indica que el modelo mejora

su capacidad para clasificar correctamente los datos conforme avanza el entrenamiento. Por destacar un posible problema en el entrenamiento de este modelo, serían los desvanecimientos en la precisión en ciertas épocas, en las que se reduce la métrica en un 20 %, por lo que se ha de tener especial cuidado a la hora de utilizar la función de *EarlyStopping*.

La Figura 6.21, por otro lado, muestra una disminución significativa de las pérdidas, con picos esporádicos, lo que sugiere que el modelo está aprendiendo a reducir los errores, aunque con algunas fluctuaciones posiblemente debido a la variabilidad inherente en los datos de entrenamiento o a la configuración del proceso de optimización.

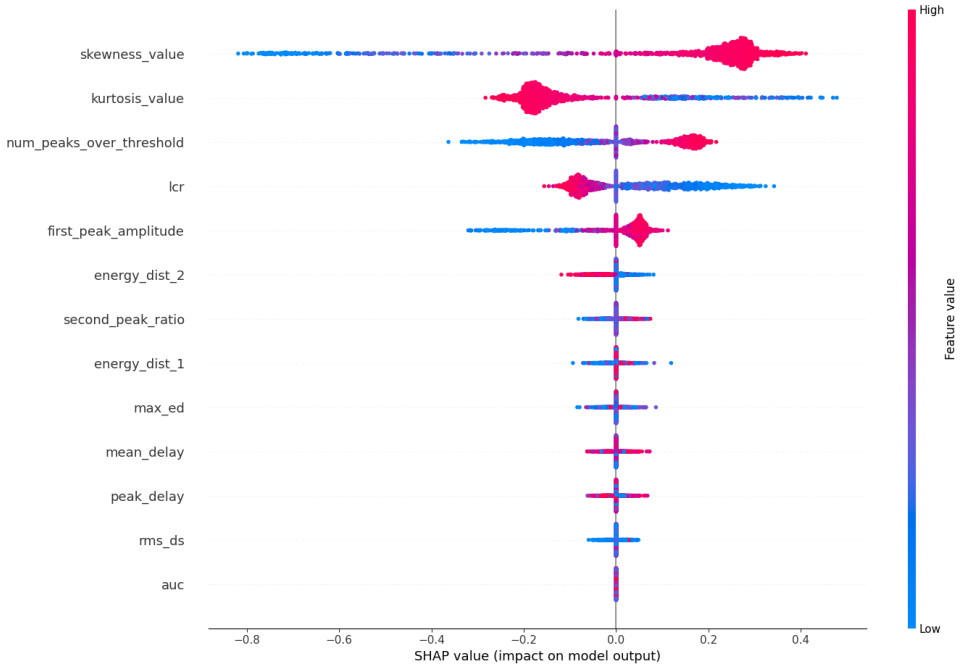

Figura 6.22: Gráfica de influencia en los parámetros en las predicciones del modelo on 10 neuronas, lr= 0.001 y *EarlyStopping*.

En la gráfica mostrada en la Figura 6.22 se representa que las características *"skewness_value"*, *"kurtosis_value"*, *"num_peaks_over_threshold"*, *"lcr"* y *"first_peak_amplitude"* tienen un impacto considerable en el modelo, evidenciado por la mayor dispersión y concentración de puntos en estas variables.

Realizando un estudio similar al de anteriores gráficas SHAP, se posicionan las *features* *"skewness_value"*, *"kurtosis_value"* y *"lcr"* como los más favorables para la clase 1, cada uno con valores altos o bajos en sus métricas.

Los valores *"skewness_value"* podrían destacarse como los más influyentes si se tuviera que escoger una, ya que valores altos representan totalmente una inclinación hacia LOS y valores bajos, un gran posicionamiento hacia NLOS , estando los valores medios en el centro.

A continuación se muestran en la Tabla 6.16 y en la Figura 6.23 los resultados de los modelos con 15 neuronas en la capa intermedia.

Learning Rate	Porcentaje de error NLOS	Porcentaje de error LOS
0.001	1.38 %	15.69 %
0.002	0.69 %	5.99 %
0.003	0 %	3.39 %
0.004	0 %	13.72 %
0.005	4.16 %	5.20 %

Tabla 6.16: Resultados ajuste de NN para 15 neuronas con *EarlyStopping*.

De los dos modelos seleccionados de la Tabla 6.16, se tomará para un análisis más profundo el modelo entrenado con un *learning rate* de 0.005.

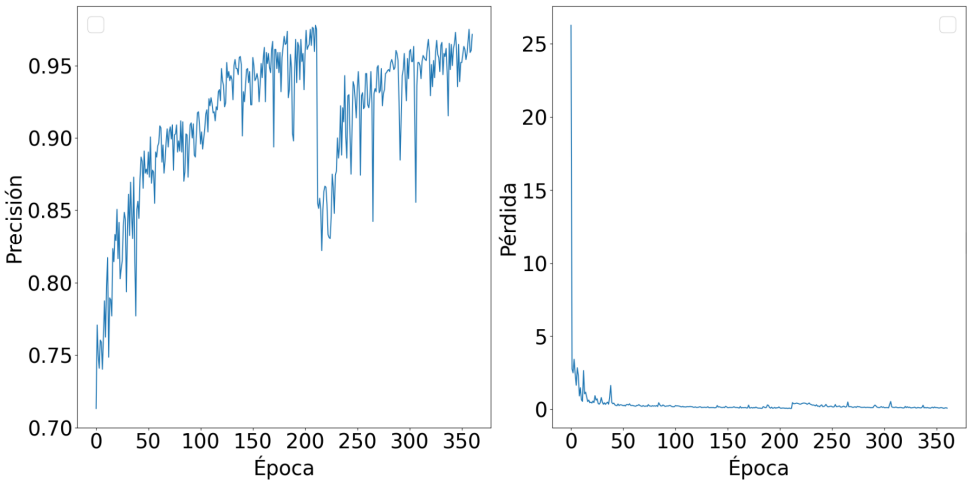

Figura 6.23: Gráficas de precisión y pérdidas del modelo con 15 neuronas, lr= 0.005 y *EarlyStopping*.

En la Figura 6.23 se muestra una tendencia de aprendizaje similar de nuevo a las anteriores, pero con la peculiaridad de un desvanecimiento en un significativo número de épocas, durando desde la época 200 hasta la 300, en la que se recuperan los niveles anteriores. Pese a ello, la precisión acaba siendo alrededor del 95 %.

Además, en la Figura 6.23 se tiene una rápida disminución de la pérdida de nuevo, lo que sugiere que el modelo aprende rápidamente a reducir los errores, y la estabilización posterior indica una fase de convergencia donde los ajustes adicionales en los pesos del modelo tienen un impacto menor en la reducción de la pérdida, lo que puede conllevar a una memorización de los patrones en vez del aprendizaje si no se para a tiempo el entrenamiento.

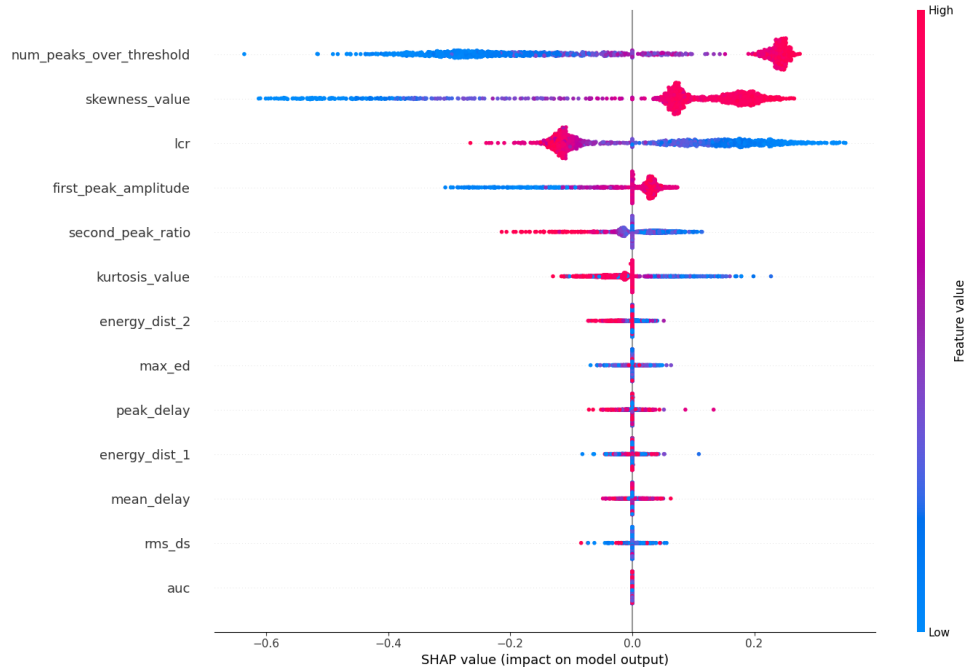

Figura 6.24: Gráfica de influencia en los parámetros en las predicciones del modelo on 15 neuronas, lr= 0.005 y *EarlyStopping*.

En la lista de *features* de este modelo, representada en la Figura 6.24, con mayor peso en la predicción repiten los parámetros *"num_peaks_over_threshold"*, con valores altos a favor de LOS y con valores medios y bajos para NLOS ; también *"skewness_value"*, siguiendo la misma norma y dispersión a lo largo del eje. Respecto a *"lcr"* también se denota un buen impacto, pero en este caso los valores altos inclinarán la balanza hacia NLOS .

Como novedad se puede destacar la presencia del ratio entre el primer y segundo pico entre las características más influyentes. La *"first_peak_amplitude"* también tiene un impacto notable, con una distribución de los puntos que sugiere que tanto los valores altos como bajos pueden influir en los resultados, aunque estos últimos de manera más pronunciada.

Las características *"energy_dist_2", "energy_dist_1", "max_ed", "mean_delay", "peak_delay" y rms_ds"* tienen un impacto menor y más consistente en las predicciones del modelo, con una menor variabilidad en los valores SHAP.

Finalmente, *"auc"* tiene uno de los menores pesos en las predicciones, con una mínima dispersión sus valores en el eje x.

Los modelos entrenados con 20 neuronas en su capa oculta arrojan los resultados representados en la Tabla 6.17 y la Figura 6.25.

Learning Rate	Porcentaje de error NLOS	Porcentaje de error LOS
0.001	10.41 %	1.65 %
0.002	0.69 %	17.74 %
0.003	0 %	9.54 %
0.004	1.38 %	2.52 %
0.005	0.69 %	8.51 %

Tabla 6.17: Resultados ajuste de NN para 20 neuronas con *EarlyStopping*.

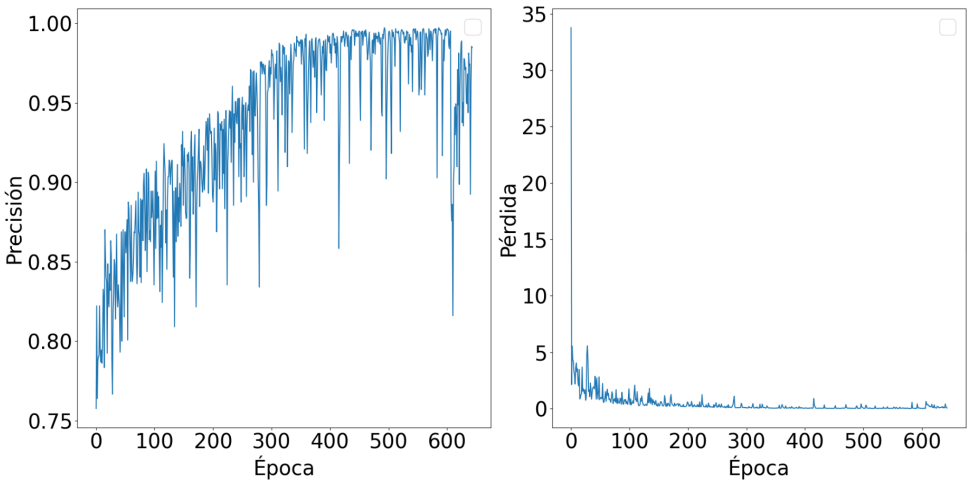

Figura 6.25: Gráficas de precisión y pérdidas del modelo con 20 neuronas, lr= 0.004 y *EarlyStopping*.

En la Figura 6.25 se representa que la precisión empieza en aproximadamente 0.75, mostrando una tendencia creciente con las épocas hasta alcanzar valores cercanos a 1.0 al final del entrenamiento. Sin embargo, también se observa una alta variabilidad, lo cual sugiere que el modelo experimenta fluctuaciones significativas en su capacidad para clasificar correctamente. A pesar de esos desvanecimientos, el modelo a partir de la época 300, deja de tener esa tendencia ascendente, y salvo los mínimos locales de los que se comentan, la precisión no mejora.

Respecto a las pérdidas se muestra una disminución rápida y sostenida desde un valor inicial hasta acercarse a cero, lo cual es un indicativo positivo de que el modelo está aprendiendo y ajustándose adecuadamente a los datos de entrenamiento.

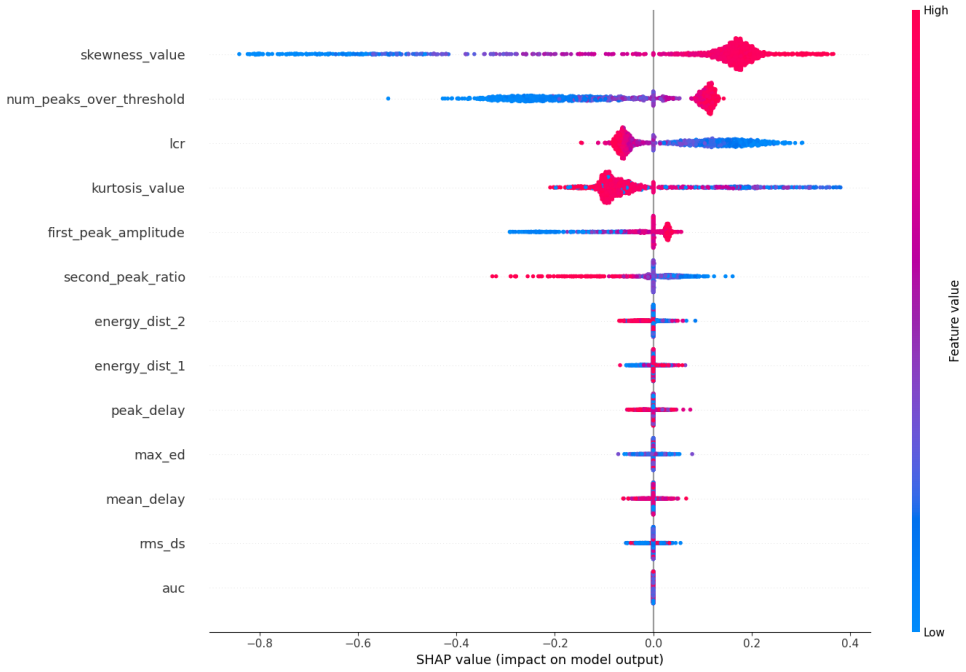

Figura 6.26: Gráfica de influencia en los parámetros en las predicciones del modelo on 20 neuronas, lr= 0.004 y *EarlyStopping*.

Los cambios en la Figura 6.26 respecto a gráficas SHAP anteriores no son notables, ya que se continúa con la dinámica de un gran impacto por parte de las *features* como *"skewness_value"* , *"num_peaks_over_threshold"*, etc.

Para destacar quedaría la dispersión de valores de distinto rango, tanto altos como bajos, a lo largo de todo el eje x, impactando de manera positiva y negativa en la predicción. Este hecho ocurre con la característica *"kurtosis_value"*.

Learning Rate	Porcentaje de error NLOS	Porcentaje de error LOS
0.001	0 %	44.40 %
0.002	0 %	44.08 %
0.003	1.38 %	5.20 %
0.004	0 %	49.44 %
0.005	0 %	20.26 %

Tabla 6.18: Resultados ajuste de NN para 25 neuronas con *EarlyStopping*.

De entre el conjunto de soluciones de la Tabla 6.18, se escoge para estudio el único posible de los 5 modelos debido al gran número de errores arrojados por los otros. Será el modelo con el *learning rate* = 0.003.

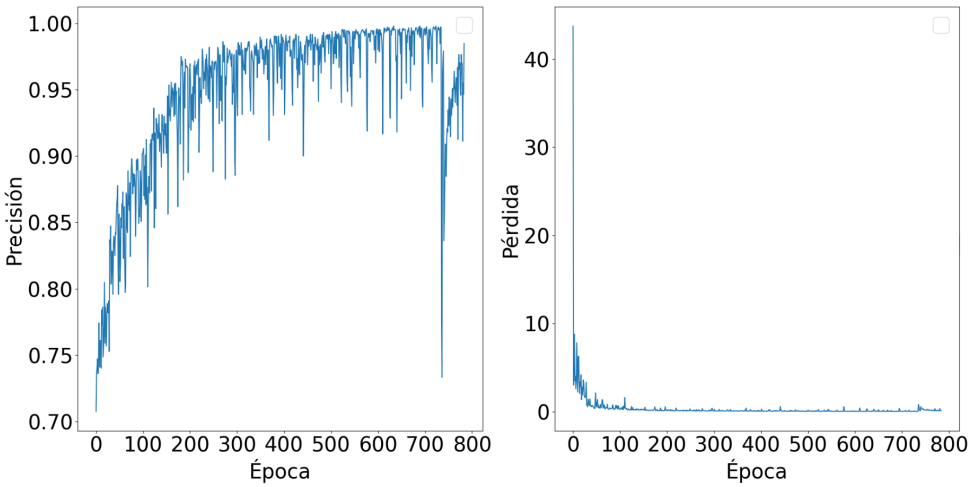

Figura 6.27: Gráficas de precisión y pérdidas del modelo con 25 neuronas, lr= 0.003 y *EarlyStopping*.

En la Figura 6.27 se muestra como el modelo sigue el mismo patrón que el escogido para 20 neuronas: un crecimiento en las primeras épocas para su posterior estabilización a partir de las 300. En este caso los desvanecimientos no son tan profundos, salvo el producido en las épocas cercanas a 750, aunque rápidamente se recupera. Aquí, de nuevo, gracias al *EarlyStopping* se evita que el modelo empeore su precisión con un sobreentrenamiento.

Con las pérdidas no hay ningún evento destacable más allá de un rápido descenso para su posterior estabilización. Sí que hay que tener especialmente cuidado al recibir unas pérdidas extremadamente bajas, lo que podría significar un posible *overfitting*, aunque en este caso la gráfica de precisión despeja duda alguna sobre este posible contratiempo.

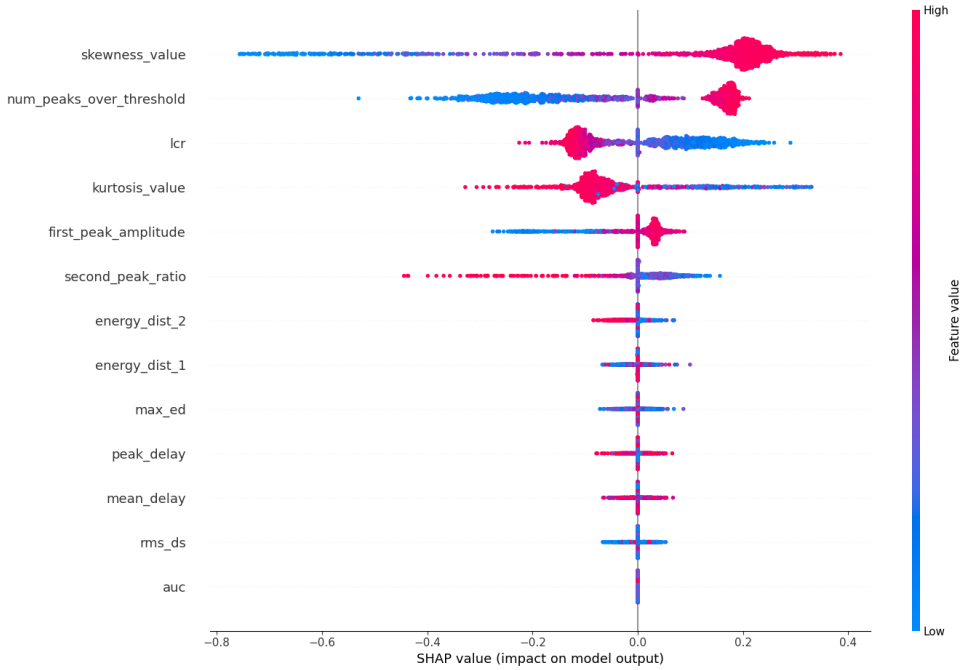

Figura 6.28: Gráfica de influencia en los parámetros en las predicciones del modelo on 25 neuronas, lr= 0.003 y *EarlyStopping*.

En la Figura 6.28 se identifica una gran dispersión de los datos en diferentes características, como la asimetría, el ratio entre el primer y segundo pico y el número de contribuciones por encima del umbral, aunque cada uno con sus distintos pesos hacia las dos clases.

Características como el área bajo la curva o la distribución de energía siguen sin tener una trascendencia significativa en el resultado.

Learning Rate	Porcentaje de error NLOS	Porcentaje de error LOS
0.001	0 %	9.06 %
0.002	0 %	6.86 %
0.003	0 %	7.49 %
0.004	2.08 %	3.47 %
0.005	0 %	16.71 %

Tabla 6.19: Resultados ajuste de NN para 32 neuronas con *EarlyStopping*.

Para este último análisis de los resultados mostrados en la Tabla 6.19, se tienen varias opciones que serían interesantes para un estudio más centrado, aunque se ha decantado por analizar el único modelo que además de tener un porcentaje bajo de errores, los comete en ambas clases. Aclarar que predecir erróneamente en ambas clases no es mejor opción que los otros modelos, ya que sus porcentajes de fallo son similares.

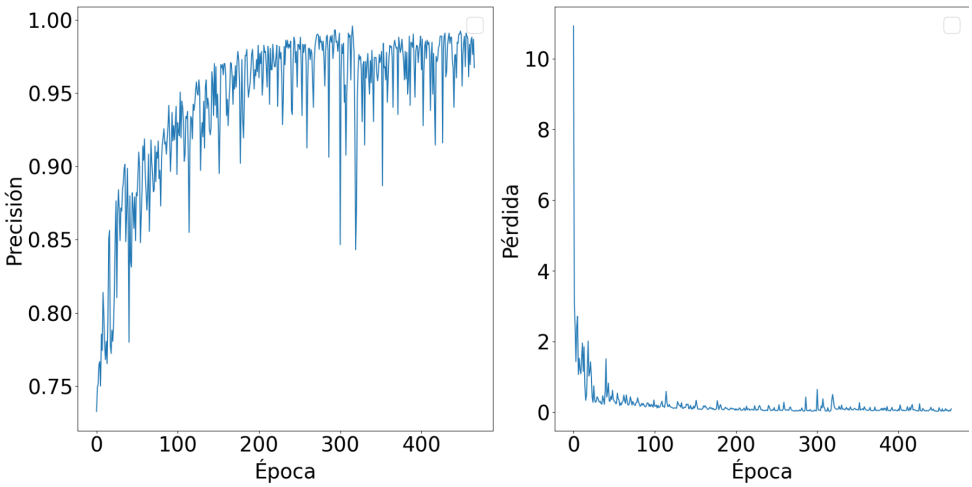

Figura 6.29: Gráficas de precisión y pérdidas del modelo con 32 neuronas, lr= 0.004 y *EarlyStopping*.

En la Figura 6.29 se alcanza un máximo alrededor de la época 300, cercano también al desvanecimiento más profundo. A pesar de ello, la gráfica denota una pendiente al alza en precisión, deteniéndose antes de la época 500. Las pérdidas destacan por sus valores casi despreciables, viéndose una escala en el eje y que apenas supera el 10. Rápidamente además se estabiliza sin ningún sobresalto.

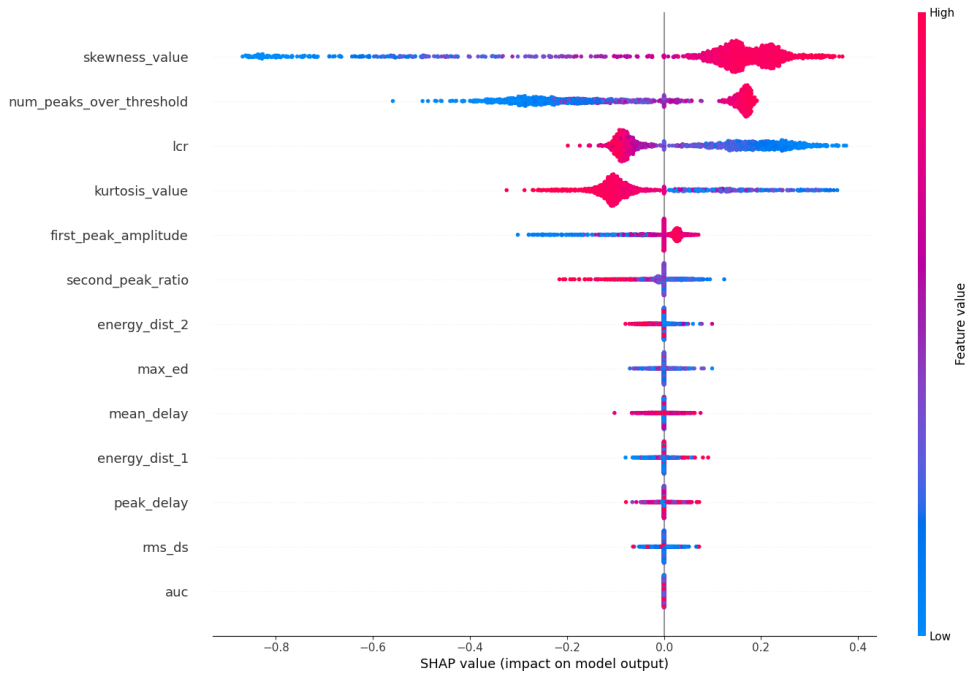

Figura 6.30: Gráfica de influencia en los parámetros en las predicciones del modelo on 32 neuronas, lr= 0.004 y *EarlyStopping*.

En la Figura 6.30 se refleja lo mismo que anteriormente, por lo que no cabe un análisis

diferencial en el que obtener conclusiones específicas. Siguen teniendo una gran consideración las 4 primeras características, dejando como residual el impacto de las más bajas en la gráfica.

Para mitigar los efectos negativos del desbalanceo, se pueden aplicar diversas técnicas. La primera estrategia, y más rudimentaria, es la recolección de datos adicionales de la clase subrepresentada. Otra técnica es el remuestreo, que incluye tanto el sobremuestreo de la clase minoritaria mediante métodos como SMOTE (*Synthetic Minority Over-sampling Technique*), como el submuestreo de la clase mayoritaria seleccionando una muestra representativa. También se podría ajustar la función de pérdida del modelo para asignar un mayor peso a los errores en la clase minoritaria, incentivando al modelo a mejorar su rendimiento en esta clase.

Por tanto, aunque el desbalanceo en el conjunto de datos de entrenamiento puede afectar negativamente el rendimiento del modelo, la aplicación de técnicas adecuadas puede mitigar estos efectos y contribuir al desarrollo de una NN más precisa y equitativa.

Capítulo 7

Conclusiones, trabajo futuro y contribuciones del proyecto

En este capítulo se analizan las conclusiones, el trabajo futuro y las contribuciones académicas, empresariales y globales, refiriéndose a la Agenda 2030 establecida por la ONU.

7.1. Conclusiones

Los resultados arrojados por los algoritmos utilizados en el TFG dan una visión transversal del funcionamiento de métodos de IA para la definición y caracterización del canal radio, punto tan importante en las telecomunicaciones.

En unos momentos en los que la eficiencia energética es un tema angular en las vías de actuación de las empresas e instituciones, el objetivo principal consiste en reducir su consumo sin renunciar a ofrecer sus servicios con una calidad extrema y cada vez a mayor volumen de población. La optimización de las redes y los sistemas de telecomunicaciones no solo contribuye a la sostenibilidad medioambiental, sino que también ofrecería beneficios económicos significativos para las empresas del sector. Las estrategias del mundo TIC (Tecnologías de la Información y la Comunicación) se basan en tres puntos principales: innovación tecnológica, optimización de procesos e infraestructuras y la gestión y *marketing* organizacional, en el que se transmite una preocupación por el cambio climático a una sociedad cada vez más concienciada. En consecuencia de la tendencia verde, la oportunidad de mejorar la eficiencia y reducir el gasto ha llevado a implementar los algoritmos mostrados en esta memoria junto con sus resultados.

El análisis del proceso de clusterización gira entorno a implementar el modelo SV para poder estandarizar el modelo y los efectos de los elementos en un contexto *indoor*. Previo al modelo SV se ha clusterizado a través de K-means, el cual ha necesitado el uso de métricas para establecer el número k óptimo. Estas métricas no han resuelto por completo la incógnita del k, sino que ha sido necesaria una revisión para acomodar los resultados de la clusterización a los objetivos, generalmente desembocando en el aumento del k para reducir el rango del ángulo acimut en cada clúster.

Además, revisando la información que más influye en la clusterización, se ha demostrado que el ángulo acimut junto con el retardo son los que más pesan en las soluciones, viéndose claramente en

las gráficas mostradas en el Capítulo 5. Como añadido, este ángulo mencionado aporta una visión clara y nítida de las direcciones horizontales en las que han sido recibidas las MPCs, proporcionando la información de qué elementos han producido qué reflexiones y cómo han afectado.

Finalmente, en la clusterización se han obtenido los parámetros γ, Γ, λ y Λ del modelo SV, los cuales han necesitado de un ajuste para evitar los valores negativos que causen unas líneas de tendencia ascendentes en potencia, referidos al parámetro Γ. Para ello, en vez de calcularlo en base a las amplitudes reales de las primeras contribuciones de cada clúster, se constituyó la amplitud de esas muestras como las amplitudes teóricas según el parámetro γ en los retardos mínimos de cada clúster. Pese a ello, algunas medidas han seguido mostrando líneas de tendencia positivas, por lo que se han retirado con el fin de no contaminar el modelo estadístico.

En la clasificación, la segunda parte principal del estudio, el análisis se podría enfocar en los parámetros obtenidos de cada PDP más influyentes en las predicciones de la IA, además de los tipos de modelos que mejores resultados aportan. Para SVM, la amplitud del primer pico, la relación entre el primer y segundo pico y la asimetría son los 3 valores que mayor peso ostentan. Respecto a los modelos según su *kernel*, el lineal con un $C = 0.0001$ aportaría la mejor solución, secundado por el sigmoide. El modelo polinómico y con *kernel RBF* no dan soluciones óptimas en su mayoría.

Las NN por otro lado demuestran con sus resultados un gran sesgo derivado por la falta de muestras NLOS en la suma total de datos. Tras la implementación de la herramienta *EarlyStopping* se ha conseguido mejorar el desempeño de los modelos, los cuales han sido probados con múltiples hiperparámetros para aportar una visión general del funcionamiento y variabilidad de este algoritmo. Respecto a los parámetros más influyentes según las gráficas SHAP, estos son la asimetría, curtosis, el número de picos sobre el umbral y el *level crossing rate*, presentes como los más importantes en todos los modelos entrenados.

7.2. Trabajo futuro

Como trabajo futuro, al estar tratando en ámbitos tan extensos como es la IA, quedan infinidad de puntos a desarrollar o a mejorar. De forma esquematizada se resumirían a modo de recomendación los siguientes pasos:

- Introducción a la simulación Monte Carlo [25] tras la parametrización del entorno a través del modelo SV.
- Reducción de los parámetros extraídos de las MPCs que no tengan un impacto significante en el resultado de las predicciones, llegando a optimizar en cuanto a tiempo y recursos energéticos la implementación y entrenamiento de los algoritmos.
- Disminución del conjunto de entrenamiento acorde con las gráficas de aprendizaje expuestas en el proyecto, en las cuales se muestra cómo a partir de ciertas muestras se reducen las mejoras en el aprendizaje. Esto conllevaría al igual que en anterior punto un ahorro importante de recursos energéticos, reduciendo las muestras para procesar y por consiguiente el tiempo de entrenamiento.
- Implementación de la herramienta *dropout*, la cual consiste en una desconexión de ciertas neuronas e impidiendo así la dependencia mayoritaria de la NN con ellas.

- Inclusión de mayor número de muestras NLOS para combatir el sesgo producido por el desbalanceo en el conjunto de datos.
- Estudio más profundo de las contradicciones extraídas del *kernel* polinómico en SVM, en el que la curvas de validación, a partir de un rango del número C se reduce considerablemente, pero en el testeo con muestras no vistas, ese mismo rango de C representa la mejor opción.

7.3. Contribución del TFG

7.3.1. Contribuciones prácticas

- **Optimización de Redes Inalámbricas**: Mediante la caracterización precisa del canal radio, las operadoras pueden optimizar la ubicación de antenas y repetidores, mejorando la cobertura y calidad de la señal en entornos *indoor*. Además, la identificación y clasificación de las MPCs permite mitigar las interferencias, resultando en una transmisión de datos más estable y de mayor calidad.
- **Desarrollo de Modelos Estándar**: La creación de modelos estándar, como el modelo SV, facilita la simulación y planificación de redes inalámbricas en diversos entornos, ahorrando tiempo y recursos en la fase de diseño e implementación. Los modelos desarrollados pueden aplicarse en una variedad de entornos *indoor*, desde oficinas y centros comerciales hasta instalaciones industriales, adaptándose a las particularidades de cada uno.
- **Implementación de Tecnologías Avanzadas**: Los avances en la caracterización del canal radio son fundamentales para el despliegue eficiente de tecnologías emergentes, asegurando que estas puedan operar a su máxima capacidad en entornos complejos. Además, la optimización del canal radio es crucial para el funcionamiento eficiente de dispositivos domóticos, mejorando la conectividad y la fiabilidad de las comunicaciones entre dispositivos.
- **Eficiencia Energética y Sostenibilidad**: La optimización en la transmisión de señales y la reducción de interferencias contribuyen a un uso más eficiente de la energía, disminuyendo el consumo energético de las infraestructuras de telecomunicaciones. Al mejorar la eficiencia de las redes de telecomunicación, se reduce la huella de carbono asociada a la operación de estas infraestructuras, contribuyendo a objetivos de sostenibilidad y reducción de emisiones.
- **Mejora en la Seguridad y Fiabilidad**: La implementación de técnicas de IA permite detectar y clasificar anomalías en la transmisión de señales, mejorando la seguridad y fiabilidad de las comunicaciones. La capacidad de clasificar y adaptar la red a diferentes condiciones de visibilidad (LOS/NLOS) asegura una mayor resiliencia ante cambios en el entorno, garantizando un servicio adecuado.

7.3.2. Contribuciones académicas

A nivel académico, la realización del TFG me ha permitido aprender y mejorar ciertos puntos tanto personales como académicos. Algunos de los temas tratados en el TFG han sido estudiados a lo largo del grado de Ingeniería de Tecnologías y Servicios de Telecomunicaciones y otros

han necesitado de un aprendizaje autónomo a través de diversas fuentes, como las referenciadas posteriormente.

Si se retoman las dos vías de estudio del TFG, se pueden dividir las metas formativas en clasificación y clusterización.

7.3.2.1. Aprendizaje por la clusterización

- Obtener un conocimiento más profundo sobre los datos derivados de las mediciones, tratando con esos datos y adaptándolos para su posterior uso.
- Alcanzar un entendimiento del modo de funcionamiento del algoritmo de IA K-means, conociendo sus procesos y cómo parametrizarlo según criterios como las métricas utilizadas.
- Comprender y poner en práctica el uso de las distribuciones de probabilidad aplicado a un problema real, conociendo sus ventajas e inconvenientes y pudiendo escoger la que mejor se adapta a las exigencias del proyecto.
- Descubrir el modo de estandarización representado por el modelo Saleh-Valenzuela, haciendo incapié en sus múltiples usos y ventajas para posteriores investigaciones.

7.3.2.2. Aprendizaje por la clasificación

- Llevar a cabo el preprocesamiento de los datos recogidos por las medidas para el correcto uso en los diferentes algoritmos de IA.
- Adentrarse en un mundo, como es la IA, para llevar a cabo diferentes algoritmos, poniéndolos a prueba ante los datos.
- Lograr un entendimiento y mejorar en la capacidad de análisis sobre las gráficas proporcionadas por los métodos de IA, tanto SVM como las NN, extrayendo las conclusiones y llevándolas a la práctica con las mejoras necesarias.
- Conocimiento de las múltiples opciones de mejoras e hiperparametrización de la IA para alcanza los objetivos propuestos.

Aparte de los puntos de conocimiento adquiridos en el TFG por parte de estas dos vías previas, en el plano personal y formativo se han mejorado u obtenido:

- Desarrollo de habilidades de programación en el lenguaje de programación *Python*, utilizando bibliotecas tan extendidas como *keras*, *scikitlearn*, *pandas*, etc.
- Evolución en la capacidad de trabajo y organización para el correcto desarrollo de un proyecto de meses de duración, planificando los tiempos de cada tarea.
- Potenciar las aptitudes de diagnosticación de problemas o puntos débiles a través de los resultados de un algoritmo y expansión en el plano de soluciones conocidas para proponer.

7.3.3. Consecución de los ODS (Objetivos de Desarrollo Sostenible de las Naciones Unidas)

- **ODS 4: Educación de Calidad**: Este proyecto promueve el aprendizaje y la aplicación de conocimientos avanzados en telecomunicaciones y técnicas de IA, favoreciendo una educación de calidad para los estudiantes involucrados. El TFG incluye el estudio y aplicación de modelos matemáticos y algoritmos de ML, proporcionando habilidades prácticas y teóricas en tecnologías emergentes.
- **ODS 9: Industria, Innovación e Infraestructura**: La investigación y desarrollo en el campo de las telecomunicaciones, especialmente en tecnologías como 5G y 6G, fomentan la innovación y la construcción de infraestructuras resilientes.
- **ODS 11: Ciudades y Comunidades Sostenibles**: Las mejoras en las telecomunicaciones contribuyen a la creación de ciudades más inteligentes y sostenibles, con mejores servicios de conectividad y comunicación, a la vez que se acercan estas herramientas a lugares menos favorecidos.
- **ODS 12: Producción y Consumo Responsables**: La optimización de recursos en telecomunicaciones puede ayudar a un uso más eficiente y responsable de la energía y otros recursos.
- **ODS 13: Acción por el Clima**: Al mejorar la eficiencia energética de las redes de telecomunicación, se trabaja en favor de la reducción de emisiones de gases de efecto invernadero asociadas a la operación de estas infraestructuras.

Referencias

[1] Centum. *Las 5 tendencias que revolucionarán la industria de las telecomunicaciones en 2023*. Accedido: 08/04/2024. 2023. URL: https://centum.com/las-5-tendencias-que-revolucionaran-la-industria-de-las-telecomunicaciones-en-2023/.

[2] PricewaterhouseCoopers (PwC). *Presión en las telecos por grandes inversiones y crecimiento limitado*. Accedido: 25/03/2024. 2023. URL: %5Curl%7Bhttps://www.pwc.es/es/sala-prensa/notas-prensa/2023/presion-telecos-grandes-inversiones-crecimiento-limitado.html%7D.

[3] YouTube. *La Primera Gran Señal Inalámbrica Fue IMPOSIBLE*. Accedido: 07/06/2024. 2024. URL: https://www.youtube.com/watch?v=Ll1TbPyYX38.

[4] Scikit-learn developers. *sklearn.metrics.plot_confusion_matrix*. Accedido: 26-04-2024. 2020. URL: https://qu4nt.github.io/sklearn-doc-es/modules/generated/sklearn.metrics.plot_confusion_matrix.html.

[5] Lorenzo Rubio Arjona. *Tema 4: Caracterización del canal radio y su impacto sobre el sistema*. Presentación de diapositivas. Accedido: 25-05-2024. 2024. URL: https://www.upv.es/pls/oalu/sic_asi.ficha_Asig?P_ASI=12433&P_IDIOMA=c&P_VISTA=normal&P_CACA=2023.

[6] María de Diego Antón. *Tema 2: Modulaciones digitales avanzadas*. Presentación de diapositivas. Accedido: 01-06-2024. 2024. URL: http://www.upv.es/pls/oalu/sic_asi.ficha_Asig?P_ASI=12429&P_IDIOMA=c&P_VISTA=normal&P_CACA=2023.

[7] Arjan Meijerink y Andreas F. Molisch. "On the Physical Interpretation of the Saleh–Valenzuela Model and the Definition of Its Power Delay Profiles". En: *IEEE Transactions on Antennas and Propagation* 62.9 (sep. de 2014), págs. 4780-4793. DOI: 10.1109/TAP.2014.2335812. URL: https://doi.org/10.1109/TAP.2014.2335812.

[8] Hao Zhanjun, Li Beibei y Dang Xiaochao. "A Signal Recovery Method Based on Bayesian Compressive Sensing". En: *Mathematical Problems in Engineering* 2019 (feb. de 2019). Accedido: 12/04/2024, págs. 1-13. DOI: 10.1155/2019/7235239.

[9] José Luis Abad Abad. "Caracterización del Canal Radio en Entornos Indoor Mediante Medidas Experimentales y Modelos Estadísticos". Accedido: 12-06-2024. Tesis doct. Valencia, Spain: Universitat Politècnica de València, 2023. URL: https://riunet.upv.es/handle/10251/195571?show=full.

[10] Anthony Barrios. *Tutorial del algoritmo K-Means en Python*. Accedido: 30-01-2024. 2023. URL: https://medium.com/latinxinai/tutorial-del-algoritmo-k-means-en-python-d8055751e2f3.

[11] Universidad de Oviedo. *El algoritmo k-means aplicado a clasificación y procesamiento de imágenes*. Accedido: 21-04-2024. 2024. URL: https://www.unioviedo.es/compnum/laboratorios_py/kmeans/kmeans.html.

[12] Daniel Rodríguez. *Identificar el número de clústeres con Calinski-Harabasz en k-means e implementación en Python*. Accedido: 07-04-2024. 2023. URL: %5Curl%7Bhttps://www.analyticslane.com/2023/06/16/identificar-el-numero-de-clusteres-con-calinski-harabasz-en-k-means-e-implementacion-en-python/%7D.

[13] Jyoti Yadav. *Selecting the Optimal Number of Clusters for KMeans with the Elbow Score*. Accedido: 19-03-2024. 2024. URL: https://www.linkedin.com/pulse/selecting-optimal-number-clusters-kmeans-score-jyoti-yadav.

[14] MathWorks. *Calinski-Harabasz criterion clustering evaluation object - MATLAB*. Accedido: 12-05-2024. 2024. URL: https://es.mathworks.com/help/stats/clustering.evaluation.calinskiharabaszevaluation.html.

[15] MathWorks. *Support Vector Machine (SVM)*. Accedido: 28-03-2024. 2024. URL: https://es.mathworks.com/discovery/support-vector-machine.html.

[16] Andrew Ng. *Supervised Machine Learning: Regression and Classification*. Accedido: 13-03-2024. 2022. URL: https://www.coursera.org/learn/machine-learning.

[17] MathWorks. *¿Qué es una red neuronal?* Accedido: 20-03-2024. 2024. URL: https://es.mathworks.com/discovery/neural-network.html.

[18] IBM. *¿Qué es una red neuronal?* Accedido: 30-04-2024. 2024. URL: https://www.ibm.com/es-es/topics/neural-networks.

[19] Diego Calvo. *Clasificación de Redes Neuronales Artificiales*. Accedido: 25-05-2024. 2024. URL: https://www.diegocalvo.es/clasificacion-de-redes-neuronales-artificiales/.

[20] Lorenzo Rubio et al. "K-factor analysis based on channel measurements from 24 to 40 GHz in a laboratory scenario". En: *Measurement-based propagation models*. Track: AP-S: Propagation and Scattering. Universitat Politècnica de València, Spain; Universidad de Cantabria, Spain. Room 10, jul. de 2024.

[21] Universidad ORT Uruguay. *Los 10 lenguajes de programación más usados actualmente*. Accedido: 14-05-2024. 2024. URL: https://fi.ort.edu.uy/blog/los-10-lenguajes-de-programacion-mas-usados-actualmente.

[22] Scikit-learn developers. *Graficando curvas de validación*. Accedido: 21-04-2024. 2020. URL: https://qu4nt.github.io/sklearn-doc-es/auto_examples/model_selection/plot_validation_curve.html.

[23] Scikit-learn developers. *Graficando curvas de aprendizaje*. Accedido: 19-04-2024. 2020. URL: https://qu4nt.github.io/sklearn-doc-es/auto_examples/model_selection/plot_learning_curve.html.

[24] Vinícius Trevisan. *Using SHAP Values to Explain How Your Machine Learning Model Works*. Accedido: 28-05-2024. 2022. URL: https://towardsdatascience.com/using-shap-values-to-explain-how-your-machine-learning-model-works-732b3f40e137.

[25] Amazon Web Services. *¿Qué es la simulación de Monte Carlo?* https://aws.amazon.com/es/what-is/monte-carlo-simulation/. Accedido: 10-02-2024. 2024.