



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**DISEÑO DE UN MÉTODO PASIVO DE UBICACIÓN DE UNA
PERSONA EN AMBIENTE DE INTERIORES BASADO EN
APRENDIZAJE AUTOMÁTICO CON DATOS OBTENIDOS A
PARTIR DE ENLACES DE COMUNICACIONES EN LA
BANDA DE 28 GHz.**

TRABAJO DE TITULACIÓN

Previa a la obtención del Título de:

MAGISTER EN TELECOMUNICACIONES

Presentado por:

VERONICA JACQUELINE OJEDA CARRERA

GUAYAQUIL – ECUADOR

AÑO: 2021

AGRADECIMIENTO

Agradezco infinitamente a Dios y a la Virgen María por su infinito amor y protección.

A mis padres Ángel y Jacqueline por todo el amor, cuidados y tiempo dedicado a lo largo de mi vida.

A mi director el PhD. Juan Carlos Avilés por todo lo aprendido y por su apoyo incondicional.

A mi compañero de vida Víctor por su paciencia, apoyo absoluto, consejos y sobre todo por acompañarme en esta hermosa etapa.

A mis compañeros y profesores de maestría por sus enseñanzas y por todo lo compartido a lo largo de este proceso.

DEDICATORIA

Quiero dedicar este trabajo a mi bebito Ángel Abdón quien se ha convertido en mi motivación, mi fortaleza y en la luz de mi vida.

A mi papá y a mi mamá quienes son los mejores padres que Dios pudo darme. Ustedes siempre han creído en mí, me han dado las mejores enseñanzas, pero sobre todo el más grande ejemplo de amor incondicional.

A mi director de proyecto el PhD. Juan Carlos Avilés que me ha dado su apoyo incondicional siempre.

TRIBUNAL DE EVALUACIÓN



Firmado electrónicamente por:
**JUAN CARLOS
AVILES
CASTILLO**

PhD. Juan Carlos Avilés Castillo

DIRECTOR DEL TRABAJO DE TITULACIÓN



Firmado electrónicamente por:
**JORGE ANDRES
BRITO
COLLANTES**

MSc. Jorge Brito Collantes

MIEMBRO PRINCIPAL DEL TRIBUNAL

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este proyecto de graduación nos corresponden exclusivamente, y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

(Reglamento de Graduación de la ESPOL)

A handwritten signature in blue ink, appearing to read 'Verónica', is shown within a rectangular grey box. The signature is fluid and cursive.

Verónica Jacqueline Ojeda Carrera

RESUMEN

En la actualidad, la aparición de la pandemia COVID-19 ha generado desafíos sociales sin precedentes. El esparcimiento del virus ha crecido de manera exponencial de modo que muchas naciones han decretado confinamientos para disminuir la tasa de contagios. El aislamiento asociado a las cuarentenas extendidas tiene efectos inmediatos en las economías mundiales, en el equilibrio socioemocional e inclusive en los cambios climáticos de las últimas décadas.

Bajo esta perspectiva, el objetivo de este trabajo es detectar, en ambientes interiores, la ubicación de una persona que no porta dispositivos electrónicos aprovechando las características de la propagación de señales de radiofrecuencia de altas frecuencias y el aprendizaje automático. El bloqueo parcial o total a la emisión RF producido por la presencia de la persona genera una variación de la potencia de recepción que permite estimar su ubicación geográfica.

El trabajo se divide en dos partes: generación de datos y selección de un modelo de aprendizaje automático supervisado basado en la precisión de la ubicación. En la generación de los datos, se usaron dos escenarios. En el primero, el sistema de comunicaciones se formó con un transmisor (Tx) y 19 receptores (Rx) equipados individualmente con dipolos de media onda (patrón de irradiación tipo omnidireccional), estos últimos instalados en una sola pared. En el segundo escenario, solo el Tx se equipó con un arreglo lineal de antenas (direccional), mientras que los receptores con antenas omnidireccionales se repartieron en tres paredes. En cada caso, el sistema de comunicación inalámbrica permitió obtener los niveles de potencia con y sin el bloqueo de personas ubicadas dentro de una habitación rectangular. Para el efecto, se aplicó una herramienta de simulación de propagación basada en trazado de rayos conjuntamente con MATLAB tanto para automatizar el uso repetitivo de la herramienta como para el registro de esos datos en un formato apropiado. En una segunda parte del trabajo, la información (dataset) generada fue procesada con técnicas de inteligencia artificial usando diferentes diseños de aprendizaje automático supervisado con el objetivo de alcanzar la mejor resolución posible. En esta parte, el algoritmo estima la posición de la persona basado en los niveles de potencia (RSS). Las redes son entrenadas con los datos de las coordenadas x , y de manera independiente usando el error cuadrático medio entre los valores reales y estimados.

Los resultados de los entrenamientos muestran que el método de aprendizaje automático supervisado Ensemble permite alcanzar el nivel más bajo de error cuadrático medio de todos métodos probados.

ÍNDICE GENERAL

AGRADECIMIENTO	i
DEDICATORIA	ii
TRIBUNAL DE EVALUACIÓN	iii
RESUMEN	v
ÍNDICE GENERAL	vi
ABREVIATURAS	xii
ÍNDICE DE FIGURAS	xiii
ÍNDICE DE TABLAS	xiv
CAPÍTULO 1	1
1 111.1	111.2
111.3	131.3.1.
131.3.2.	131.4
141.5 ¡Error! Marcador no definido.	CAPÍTULO 2
5	
2 152.1	152.2
¡Error! Marcador no definido.	2.3 ¡Error! Marcador no definido.
2.4	2.4
¡Error! Marcador no definido.	2.5 ¡Error! Marcador no definido.
2.5.1	2.5.1
¡Error! Marcador no definido.	2.5.2 ¡Error! Marcador no definido.
2.6	2.6
¡Error! Marcador no definido.	CAPÍTULO 3
30	
3 ¡Error! Marcador no definido.	3.1 ¡Error! Marcador no definido.
3.2.	3.2.
¡Error! Marcador no definido.	3.3 ¡Error! Marcador no definido.
3.4	3.4
¡Error! Marcador no definido.	3.5. ¡Error! Marcador no definido.
3.6.	3.6.
¡Error! Marcador no definido.	3.7. ¡Error! Marcador no definido.
CAPÍTULO	
4 36	

4	¡Error! Marcador no definido.4.1	¡Error! Marcador no definido.4.2	4.2
	¡Error! Marcador no definido.4.3	¡Error! Marcador no definido.4.4	4.4
	¡Error! Marcador no definido.	CAPÍTULO 5	48
	CONCLUSIONES		48
	RECOMENDACIONES		49
	APÉNDICES		50
	BIBLIOGRAFÍA		57

ABREVIATURAS

RSSI	Received Signal Strength Indicator
DFL	Device Free Localization
LOS	Line of Sight
NLOS	No Line of Sight
PIR	Passive Infra Red
WiFi	Wireless Fidelity
RFID	Identificación por Radiofrecuencia
ULA	Uniform Linear Array

BR	Bayesian Regularization
LM	Levenberg Marquardt
LR	Linear Regression
EBT	Ensemble Bagged Trees
GHz	Giga Hertz
GPS	Global Positioning System
mW	Mili Watts
Tx	Transmisor
Rx	Receptor

ÍNDICE DE TABLAS

Tabla 3.1 Vectores de arreglo de antena direccional ULA	17
Tabla 3.2 Parámetros del escenario simulado en la primera configuración	22
Tabla 3.3 Parámetros del escenario simulado en la segunda configuración	23
Tabla 3.4 Estructura del dataset	25
Tabla 4.1 Resultados de los valores promedios de las predicciones realizadas bajo el algoritmo Ensemble Bagged Trees.	46
Tabla 4.2 RSME de cada algoritmo probado	46

CAPÍTULO 1

1 ANTECEDENTES.

1.1 Descripción del problema

Los sistemas de ubicación (posicionamiento) permiten determinar geográficamente ya sea terminales de comunicación o la presencia de personas tanto en exteriores como interiores. Esta ubicación se ha vuelto muy importante actualmente debido al impacto de la pandemia Covid-19. En el presente proyecto de titulación se propone determinar la ubicación de

personas en forma pasiva, esto es, sin que lleven ningún dispositivo electrónico tales como terminales de comunicación o brazaletes tipo RFID. Se trata de aprovechar las atenuaciones o bloqueo producidas por las personas a las emisiones de radiofrecuencia desde un transmisor que opera en la banda de 28 GHz en los interiores de edificios. Para el efecto se debe estudiar las principales características eléctricas del cuerpo humano y de los materiales del sitio, así como también las características de propagación en interiores y que dependen de la ubicación de la persona, transmisor y receptores. Adicionalmente se debe investigar diferentes métodos de inteligencia artificial para determinar el que proporciona la mejor resolución. No se trata de un tema trivial debido a que las reflexiones en las paredes en un ambiente de interiores dificultan una mayor diferenciación en los niveles de potencias recibidas.

1.2 Justificación

Debido a que no existen muchos métodos de ubicación pasiva basados en técnicas de aprendizaje automático, se propone una solución inicial al problema de distanciamiento social en espacios cerrados para contribuir en el reforzamiento de las medidas de seguridad y protección frente a la posibilidad de contagio por Covid-19. La propuesta se considera oportuna especialmente en lugares de reunión dentro de edificios donde se dificulta el control usando métodos alternativos basados en videos (por ejemplo, el estudio sobre el comportamiento de la movilidad humana y los efectos del bloqueo de señales de bandas milimétricas [1]). En la presente propuesta, se aprovecha las señales de radiofrecuencia emanadas desde un transmisor para determinar la posición de una persona en el sitio de interés. A diferencia de otros métodos, el proyecto, en ningún caso podría vulnerar la privacidad de las personas o afectar el evento en sí.

La valiosa identificación de la posición de una persona pretende finalmente evitar y alertar de potenciales aglomeraciones en ambientes cerrados que faciliten un contagio inadvertido. Evidentemente este trabajo también tiene otro tipo de aplicaciones tales como el cuidado y monitoreo constante de personas de avanzada edad en lugares especializados. Bajo esta perspectiva,

la investigación realizada en [2] estima con gran precisión la ubicación (activa) de la persona en dos ambientes de interiores diferentes. En [3], se estudia un posicionamiento pasivo para personas mayores dentro de un sitio interior. En este caso se obtiene una buena precisión usando dos lugares de 19 y 26 m² y usando 50 y 60 enlaces RF lo que da un promedio referencial de 2.6 y 2.4 enlaces/m², respectivamente. De manera similar, el trabajo realizado en [4] presenta una técnica basada en algoritmos de aprendizaje automático para estimar en forma pasiva el porcentaje de acierto en la ubicación de una persona usando transmisores instalados cada 0.6 m en un ambiente interior que tiene un área muy pequeña (9 m²). Se aprovecha la variación en el patrón de señal que genera la presencia de una persona.

Por otro lado, existen trabajos de investigación que utilizan detección activa con métodos de aprendizaje automático para aprovechar adecuadamente el entorno de oficina de manera que se localice y se provea rutas seguras de navegación a quienes trabajan en ambientes con una alta aglomeración tales como los observados en empresas de ventas al por menor [5]. En otra aplicación [6], con similar metodología, se integran varios sensores residenciales con énfasis en cámaras 3D y arreglos de micrófonos y sensores PIR.

El trabajo de investigación realizado en [7] menciona el uso de bandas milimétricas para satisfacer el creciente uso de aplicaciones en interiores y los numerosos desafíos vinculados a la movilidad, bloqueos, hardware, entre otros. Las experimentaciones se centran en el uso de Puntos de acceso (Access Point) para extraer las características de propagación del ambiente de interiores. En [8], se hace evidente que la presencia de personas genera un bloqueo o disminución de potencia significativa de la señal recibida en enlaces de 28 GHz. Específicamente en este trabajo, se caracteriza los efectos de dispersión alrededor de enlaces en interiores a 28 GHz cuando dicho enlace está completamente bloqueado por una persona. El impacto de la dispersión se mide usando un reflector metálico y un cuerpo humano. Estudios similares se realizan en las bandas de 2.4 y 11 GHz [9], [10].

Dado que la propuesta incluye simulaciones del ambiente usando la herramienta de trazado (Ray Tracing) Wireless InSite de la empresa Remcom [11], los datos usados son realistas a diferencia de lo que se podría aproximar mediante métodos netamente estadísticos [12]. La literatura confirma que el uso de simuladores basados en técnicas de trazado de rayos tiene una muy buena precisión [12]. Por otro lado, el método puede determinar en forma automática y dinámica la ubicación de la persona incluyendo todos los cambios que se presentan en el lugar (oficina, lugar de comida, etc.).

1.3 Objetivos

1.3.1. Objetivo general

- Diseñar un método de ubicación de una persona en un lugar interior de un edificio, en la banda de 28 GHz, usando técnicas de aprendizaje automático supervisado.

1.3.2. Objetivos específicos

- Diseñar un sitio de simulación en interiores, incluyendo una persona con dimensiones y características eléctricas realistas.
- Simular el canal inalámbrico en la banda milimétrica de 28 GHz usando una herramienta de simulación de propagación.
- Determinar el método de aprendizaje automático supervisado aplicable que genere un error mínimo en la posición buscada.

1.4 Metodología

El presente proyecto se basa en la teoría de la metodología de la investigación científica de tipo cualitativa ya que busca determinar el grado de causalidad entre variables del proyecto, haciendo uso de métodos de análisis de datos dentro de una perspectiva espacial. La investigación de este tipo permite estudiar la influencia de la presencia de una persona en los niveles de señal recibida en un enlace de comunicaciones en la banda de 28 GHz con y sin línea de vista (LOS y NLOS). El proyecto pretende determinar la ubicación de una persona usando las técnicas de aprendizaje automático supervisado y la

combinación de herramientas de software tales como trazado de rayos y MATLAB. El ambiente de trabajo contempla la simulación de un enlace de comunicación entre una antena transmisora y un número de receptores afectados por la presencia de una persona. Por cada posición diferente del cuerpo humano, se determinan los valores de potencia recibidos. La cantidad de simulaciones será un parámetro por determinar dependiendo de la precisión y del tiempo de reentrenamiento necesario. Se analizarán algoritmos de aprendizaje automático supervisado para optimizar el error de detección en la posición de la persona. Se espera que una extensión del presente trabajo permita determinar la distancia entre dos personas ubicadas en el mismo sitio de interiores.

1.5 Alcance

El proyecto, basado en el diseño de un método de aprendizaje automático supervisado, debe establecer la ubicación geográfica de una persona que no porta ningún dispositivo electrónico en ambientes cerrados. Además, se debe adquirir, mediante simulación, una base de datos de potencias que facilite determinar con precisión las coordenadas x, y de la posición buscada.

CAPÍTULO 2

2 ESTADO DEL ARTE.

2.1 Introducción

El surgimiento de la pandemia Covid-19 ha causado pérdidas irreparables en varios aspectos a nivel mundial. Una de las medidas sugeridas por la OMS para contrarrestar su propagación es mantener el distanciamiento social. Frente a ello, los sistemas tanto de detección de personas como de su localización en un ambiente de interiores muestran ser alternativas de ayuda.

Dado que en el presente trabajo se propone desarrollar un método de localización de características particulares, los principales tópicos por considerarse son la detección pasiva de objetos, intensidad de la señal recibida como identificador de la posición y las técnicas eficientes de aprendizaje automático.

2.2 Detección pasiva de objetos mediante sistemas de comunicaciones inalámbricas

Los métodos de detección pasiva de objetos (Device Free Localization - DFL) consideran las variaciones de la propagación de la señal sin la necesidad de que el objeto (persona) porte dispositivo electrónico alguno. Además, establecen el procesamiento de las muestras para determinar una aproximación de la ubicación. Esta técnica presenta un futuro muy prometedor en diversos tipos de aplicaciones, como, por ejemplo, en situaciones de desastre, se puede realizar una tomografía RF para localizar sobrevivientes sin necesidad de encontrarse dentro de la zona de desastre representando una optimización en el tiempo de rescate. Otro ejemplo de aplicación está dirigido a las actividades que pueden controlar parámetros como la iluminación, calefacción y ahorro de energía cuando no se detecten personas en una habitación. Inclusive es usado para monitorear remotamente personas de la tercera edad en hospitales y asilos con la finalidad de no vulnerar su privacidad. La figura 2.1 muestra un ejemplo de aplicación de sistemas DFL en ambiente cerrados.

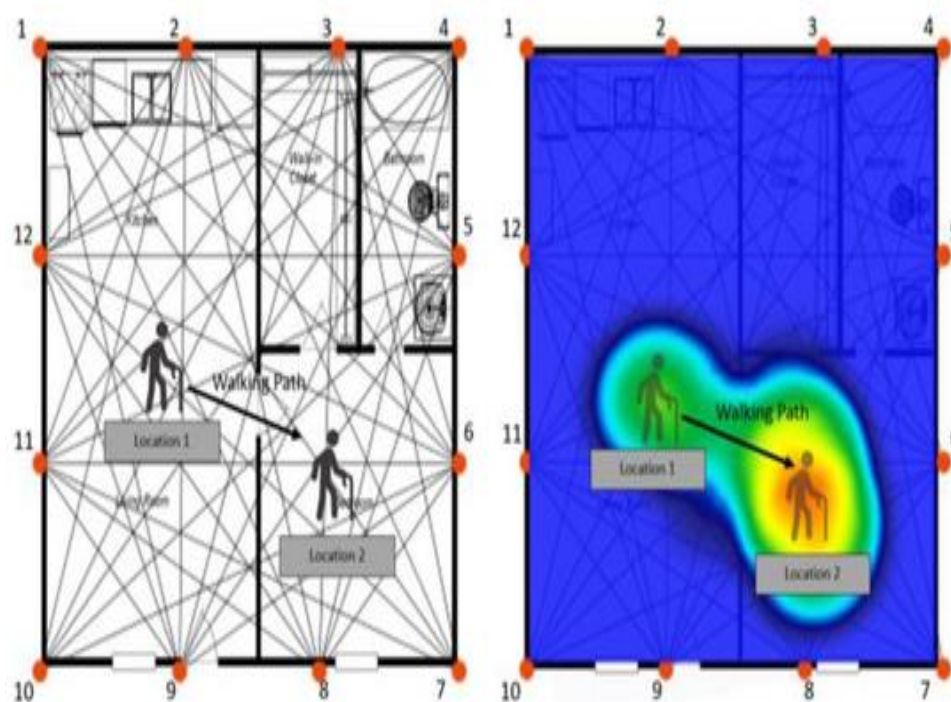


Figura 2.1: Ilustración de ejemplo de sistemas DFL.[30]

La eficacia de DFL depende del tamaño de la base de datos experimentales y una buena correlación entre los mismos, esto es, de los parámetros de propagación de las señales de radiofrecuencia. [15]

2.3 Sistema de monitoreo de personas empleando Detección pasiva de objetos mediante sistemas de comunicaciones inalámbricas

Existen propuestas de investigación que desarrollan algoritmos de localización de personas. Li, Y. et al. establece la relación entre el estado real, las ubicaciones de los sensores y las mediciones [15]. Uno de los sistemas más destacados es el método secuencial de Monte Carlo que mejora significativamente la eficiencia computacional y consecuentemente la precisión de la localización del objetivo. En este estudio se realiza un seguimiento tomográfico de RF del objeto para estimar parámetros claves de propagación con ayuda del algoritmo EM (Expectation - Maximization) en línea.

En el trabajo desarrollado en [4] se desarrolla una nueva técnica basada en algoritmos de aprendizaje automático para estimar remotamente la posición de personas mayores. Las características de absorción del cuerpo humano exhiben un patrón de señal que se usa para el rastreo y localización de las personas en ambientes cerrados.

Saeed, Ahmed et.al. [16], proponen un sistema de detección pasiva que detecta errores aplicando estadística y filtrado secuencial de Monte Carlo en un entorno dinámico para la localización del movimiento de destino.

Otra de las técnicas de aprendizaje automático prueba el rendimiento de DF después de un largo período de tiempo de entrenamiento del algoritmo [17].

Para abordar los desafíos de la detección pasiva, Tsuchida, S. et al., aplica regresión logística basada en el aprendizaje automático al posicionamiento considerando un conjunto de datos (fingerprinting) de potencia recibida. Además, explota el potencial de la base de datos generada a partir de Wi-Fi y señales BLE (Bluetooth Low Energy) para una mejor precisión en la estimación usando la diversidad de los efectos de transmisión. [18]

Por su parte, en [20] se presente MoLoc. Este sistema de detección tiene la capacidad de aprender automáticamente el patrón de variación predictiva de los datos de potencia (fingerprinting) y transferir su medición en línea para adaptarse a los movimientos dinámicos del barco en tiempo real.

Los autores de [29] desarrollaron experimentos en ambientes de interiores que disponen de una mayor cantidad de transmisores y receptores y que sufren cambios con el tiempo (cambio de posiciones de los ocupantes o de los objetos en su interior) para determinar la precisión del método de detección bajo el modelo de regresión logística de aprendizaje automático supervisado.

2.4 Detección pasiva a través de niveles de una base datos de potencias recibidas (RSS)

El RSSI es un indicador de intensidad de la señal recibida determinado por el receptor de radio que puede relacionarse con la distancia del dispositivo que transmite el mensaje y, por lo tanto, se puede utilizar para determinar las coordenadas relativas o absolutas de un objeto. [19] Generalmente, los sistemas DFL utilizan parámetros de propagación tales como la intensidad de las señales recibidas (RSS) como una métrica de la calidad de la señal [1].

Los autores de [19] consideran que el uso de los niveles de RSSI reduce la cantidad de datos a ser normalizados en el entrenamiento y la eficiencia espectral.

Así también, DFL utiliza el indicador RSSI para caracterizar la ubicación del usuario [20]. Dentro de las pruebas, los autores consideran las características estadísticas extraídas de las señales obtenidas en forma inalámbrica.

2.5 Herramienta de simulación en ambiente en interiores en bandas mmW

Las simulaciones del ambiente se desarrollan sobre la herramienta denominada Wireless Insite, basada en Ray Tracing. El programa de la empresa Remcom es usado para modelar escenarios en entornos exteriores como interiores y sistemas de comunicaciones en el rango de 50 MHz a 100 GHz. Es capaz de predecir los diferentes pasos de propagación junto con sus principales características tales como potencia recibida promedio e individual por cada rayo, ángulos de salida y llegada de cada rayo, retrasos, cambios de fase, etc). Permite adicionalmente visualizar los diferentes tipos de enlaces Tx-Rx tipo línea de vista (LOS) y No línea de vista (NLOS), establecidos a través de rebotes y difracciones; estos últimos en fillos verticales, horizontales y bordes de la persona [21]. En el presente trabajo se realizaron y procesaron 248 simulaciones para la primera parte que se aplicaron como datos de entrada para los algoritmos de clasificación y posteriormente 2728 (248x11) simulaciones para la segunda parte que se aplicaron en los algoritmos de regresión logística. La Figura 2.2 muestra un ejemplo del tipo de simulaciones que pueden realizarse con Wireless Insite. La figura muestra también los diferentes pasos de propagación en diferentes colores asociados al nivel de la potencia recibida.

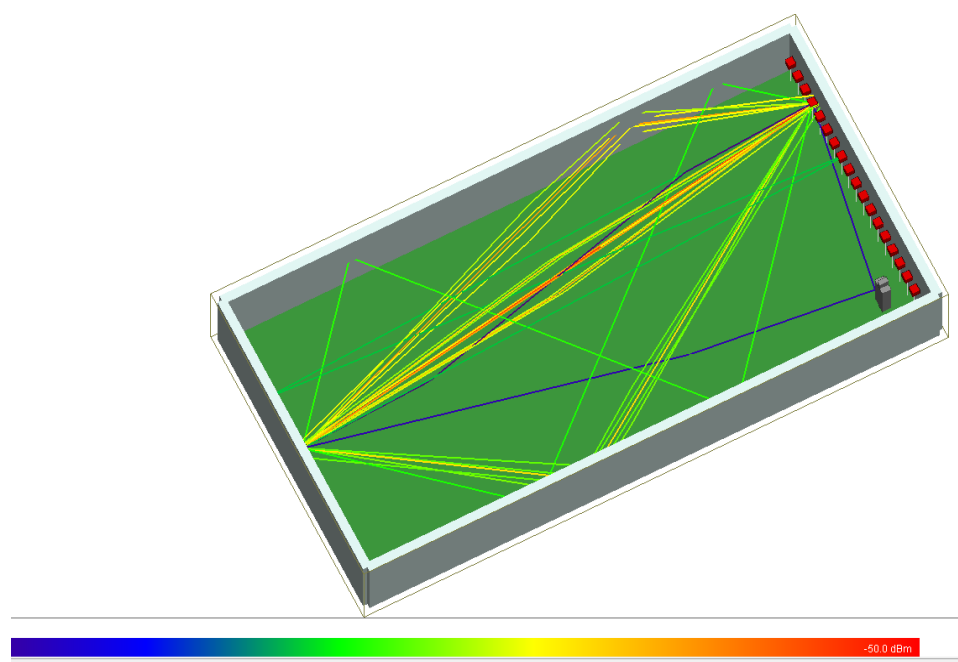


Figura 2.2: Herramienta Wireless Insite

Otra de las características más importante del Wireless Insite es que los resultados de la simulación se presentan en forma de archivos de texto que fácilmente pueden ser procesados con otras herramientas de simulación tales como MATLAB o Python. Para el presente trabajo, el parámetro de propagación a ser considerado es la potencia recibida en los receptores. Algunos de los pasos de propagación son bloqueados o difractados debido a la presencia aleatoria de una persona, afectando la potencia recibida en los Rxs.

2.5.1 MATLAB

MATLAB es un entorno de programación que permite analizar y diseñar sistemas facilitando la visualización y generación de datos. Sus creadores (Mathworks) han desarrollado productos conocidos como toolboxes o paquetes, entre ellos, aquellos orientados al aprendizaje automático, los cuales son usados para realizar aplicaciones con enfoque de inteligencia artificial. [24] Todos sus paquetes son capaces de integrarse entre sí y con otras funciones de MATLAB para crear potentes sistemas para el aprendizaje automático. En el presente proyecto, MATLAB se aplicó para automatizar las

simulaciones. Se debió crear una interfase para arrancar cada simulación cambiando la posición de la persona dentro del ambiente de trabajo. Dado que, por cada simulación, Wireless Insite genera un gran número de tablas tipo texto con los datos de propagación, MATLAB se vuelve a aplicar para extraer los datos necesarios de potencia. De la misma manera, paquetes ya desarrollados por MATLAB (toolboxes) se aplican para diseñar el modelo de aprendizaje automático apropiado.

2.5.2 Neuronal Net Fitting

En el presente trabajo se emplea la herramienta de visualización Neuronal Net Fitting de Matlab. La aplicación permite seleccionar datos, dividirlos en conjuntos de entrenamiento, validación y prueba, definir la arquitectura de red y entrenar la red. En la Figura 2.3 se observa la interfaz gráfica de la aplicación.

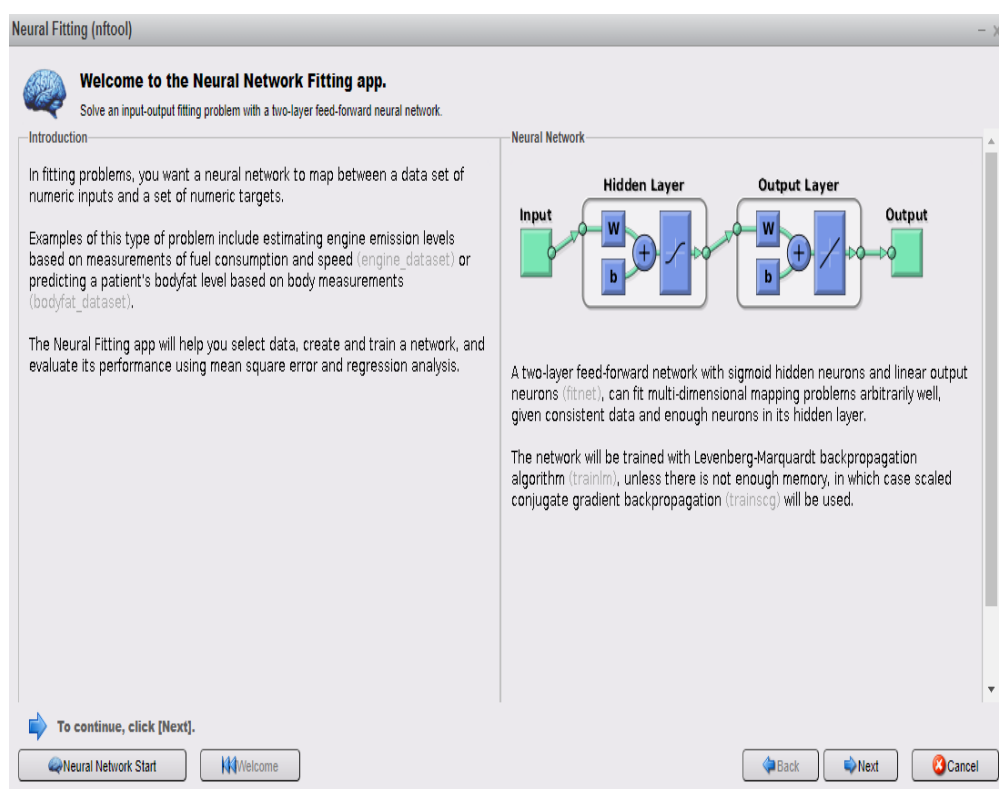


Figura 2.3: Interfaz gráfica de la aplicación Neuronal Net Fitting de Matlab.

2.5.3 Regression Learner

Regression Learner es una aplicación de MATLAB capaz de analizar los datos, clasificar las características (features), presentar una lista de posibles métodos de validación, optimizar el modelo y estimar el rendimiento de predicción del algoritmo a través de un valor numérico. [24] La interfaz gráfica de la aplicación es presentada en la Figura 2.4.

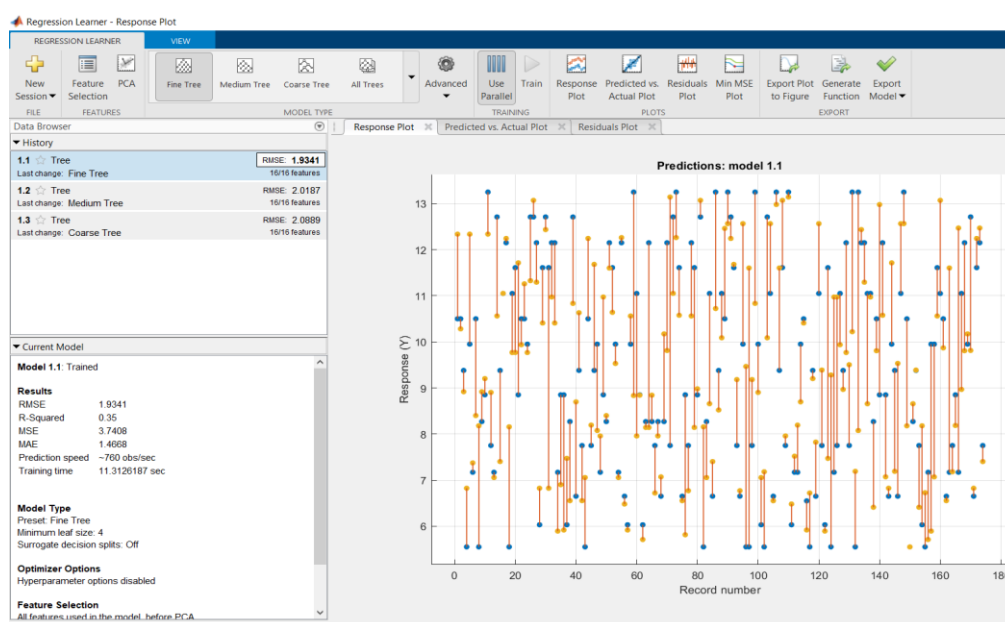


Figura 2.4: Interfaz gráfica de la aplicación Regression Learner de Matlab.

2.6 Métrica de rendimiento

Una de las métricas utilizadas por las herramientas de MATLAB para estimar la precisión del algoritmo y en consecuencia la precisión de localización de las personas es el error cuadrático medio (Root Squared Mean Error - RSME).

Se considera precisión de localización al promedio estadístico de la diferencia entre una ubicación estimada y la ubicación real del objeto. [3]

Además, el RSME resalta el algoritmo con menor error dentro de las aplicaciones descritas en párrafos anteriores. [24]

Para este proyecto, la precisión de localización depende del método y del algoritmo a emplearse.

CAPÍTULO 3

3 DISEÑO DEL MÉTODO DE POSICIONAMIENTO.

3.1 Introducción

En este capítulo se describe las características del ambiente de simulación, los parámetros de propagación, el entorno y el método de recopilación de datos. Luego, se utilizan los resultados de las simulaciones de Ray Tracing de la herramienta Wireless Insite para generar el conjunto de datos de entrenamiento y de prueba. Se diseña el algoritmo de posicionamiento basado en algoritmos de aprendizaje automático supervisado usando Matlab para adaptar los datos recopilados. Para efectos de comparación se analizará la precisión del método de posicionamiento de los algoritmos de aprendizaje supervisado: redes neuronales y regresión lineal, como se describe en la Figura 3.1.

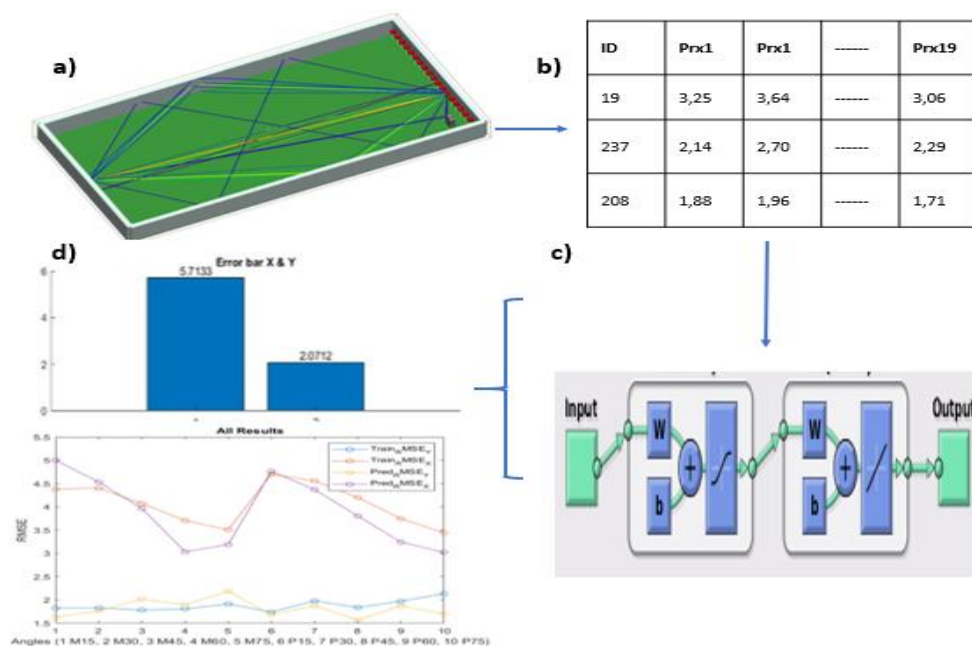


Figura 3.1: Esquema básico de diseño propuesto, a) Escenario de simulación, b) dataset utilizado, c) arquitectura del algoritmo, d) comparación de diseño

3.2 Análisis de las características físicas y eléctricas del ambiente de simulación

El escenario simula un sistema de comunicaciones en una habitación interior de dimensiones $20 \times 10 \times 3 \text{ m}^3$ con piso y techo de hormigón y paredes de ladrillo, típicos en nuestro medio. El transmisor se equipa con una antena omnidireccional o con un arreglo lineal de antenas mientras que los receptores usan antenas omnidireccionales únicamente. El sistema de comunicaciones trabaja bajo dos distintas configuraciones que serán detalladas posteriormente.

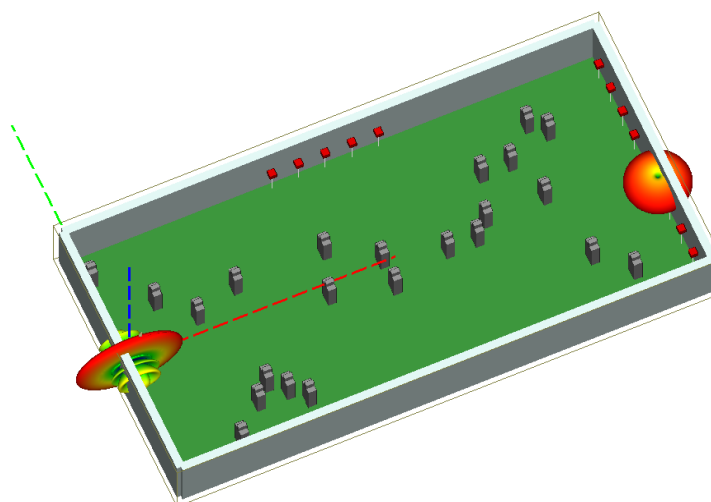


Figura 3.2: Características del ambiente de interiores propuesto.

3.3 Análisis de las características físicas y eléctricas del cuerpo humano.

El cuerpo humano utilizado en las simulaciones tiene dimensiones $1.80 \times 0.50 \times 0.30 \text{ m}$, altura del piso hasta el hombro de 1.50 m , y dimensiones de la cabeza de $0.16 \times 0.25 \text{ m}$. Las características eléctricas de permitividad relativa y conductividad son de 16.6 y 25 [S/m] , respectivamente. En la Figura 3.2 se visualiza el ambiente de simulación descrito. La Figura 3.3 presenta las 248 posiciones posibles de la persona. La distancia entre las posiciones varía entre 0.50 m y 1 m aproximadamente.

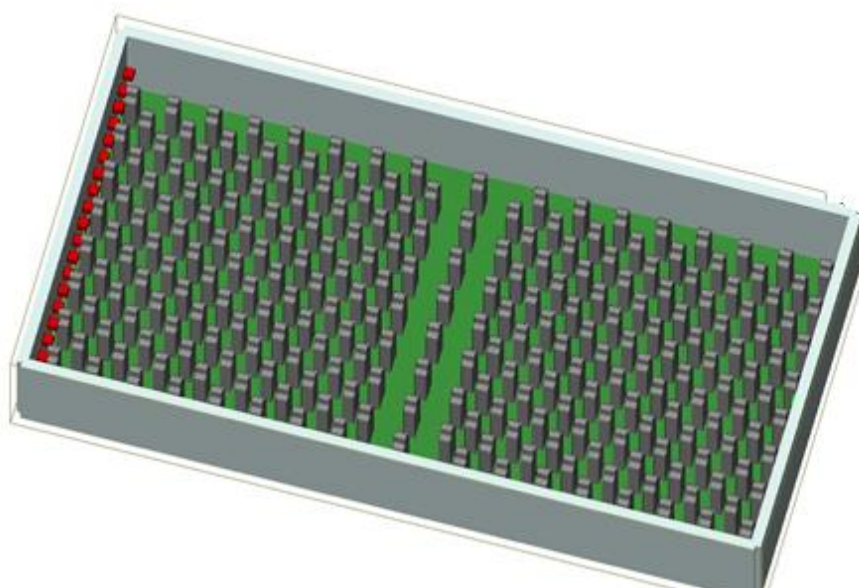


Figura 3.3: Características físicas de los sujetos colocados en el ambiente de interiores.

3.4 Características de antena omnidireccional y direccional

Mediante el uso de dos tipos de antenas, se simula el sistema de comunicaciones en un ambiente de interiores (indoor) que trabajan en la banda de 28 GHz. La configuración (Wireless Insite) de las antenas de ambos equipos se presenta en la Figura 3.4.

In use	Type	Description	Polarization	Maximum gain	VSWR	Waveform	Filename
Yes	Half-wave dipole	Half Wave Dipole	Vertical	Auto	1.000	Sinusoid 28 GHz BW 500 MHz	N/A
No	Half-wave dipole	Created for use by set (004 Rx)	Vertical	Auto	1.000	Sinusoid 28 GHz BW 500 MHz	N/A
Yes	Isotropic	Linear Array 1	Vertical	Auto	1.000	Sinusoid 28 GHz BW 500 MHz	N/A

Figura 3.4: Creación de antenas en software Wireless Insite.

En la Figura 3.5 se puede visualizar las características más relevantes de la antena tipo omnidireccional simulada. Se trata de un dipolo de media onda con una ganancia máxima de 3 dB.

Half-wave dipole antenna properties

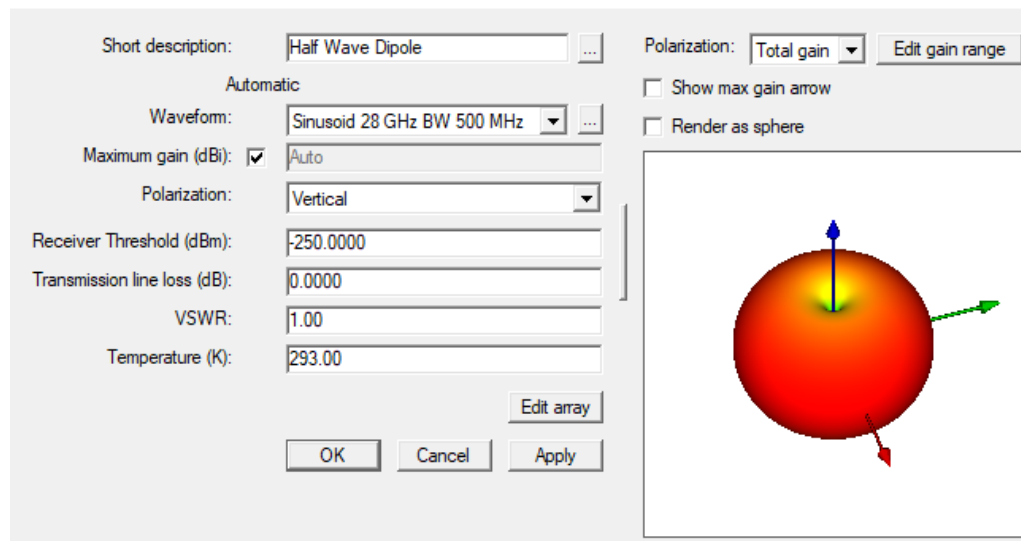


Figura 3.5: Características de la antena direccional simulada.

Por otro lado, se definieron también los parámetros más importantes de la antena direccional implementada mediante un arreglo lineal de 9 antenas isotrópicas separadas físicamente una distancia de media longitud de onda y un ancho de haz de 11.4° ($\theta = 2 * \sin^{-1} \left(\frac{0.891}{N} \right) * 180/\pi$) en el lóbulo principal. Los pesos que se aplicaron a cada una de las antenas para generar el patrón de radiación correspondiente a una ULA-9 (Uniform Linear Array – 9 antenas) se indica en la ecuación 3.1

$$w(n) = \exp \left\{ i \left[n - \left(\frac{N-1}{2} \right) \right] \left(\frac{2\pi d}{\lambda} \right) \sin(\theta_{az}) \right\}; \quad n = 0, 1, 2, \dots, N-1 \quad (3.1)$$

Se asumió un ángulo de elevación de 90° en todos los casos.

La ganancia total del arreglo viene dada por la ecuación 3.2:

$$Gain = 10 \log(n) + Gain_{antena\ base} \quad (3.2)$$

en donde n (=9) representa el número de antenas isotrópicas usadas y $Gain_{antena\ base}$ es la ganancia individual de las antenas.

Wireless Insite permite crear un arreglo de antenas. Para el efecto, la herramienta Wireless Insite emplea la magnitud, la fase y la ubicación relativa de cada elemento (respecto de la ubicación geográfica de la antena) para formar el patrón de radiación buscado. En el caso presente se varía la fase de la señal aplicada a cada antena para cambiar la orientación (ángulo) donde se produce la máxima ganancia de la antena direccional (-75° , -60° , ..., 75°), estando esta antena asignada exclusivamente al transmisor.[13].

Por cada ángulo (-75° , -60° , -45° , -30° , -15° , 0° , 15° , 30° , 45° , 60° , 75°) se configuraron los vectores de pesos como se observa en las Tabla 3.1.

Tabla 3.1 Vectores de arreglo de antena direccional ULA

Grados	Magnitud	Fase (grad)	X	Y	Z
-75 °	1	24,533405	0	4,978571	1,5
	1	-161,5999	0	4,983929	1,5
	1	12,266703	0	4,989286	1,5
	1	-173,8666	0	4,994643	1,5
	1	0	0	5	1,5
	1	173,86665	0	5,005357	1,5
	1	-12,2667	0	5,010714	1,5
	1	161,59995	0	5,016071	1,5
	1	-24,53341	0	5,021429	1,5
-60°	1	96,461709	0	4,978571	1,5
	1	-107,6537	0	4,983929	1,5
	1	48,230855	0	4,989286	1,5
	1	-155,8846	0	4,994643	1,5
	1	0	0	5	1,5
	1	155,88457	0	5,005357	1,5
	1	-48,23085	0	5,010714	1,5
	1	107,65372	0	5,016071	1,5
	1	-96,46171	0	5,021429	1,5
-45 °	1	-149,1169	0	4,978571	1,5
	1	-21,83766	0	4,983929	1,5
	1	105,44156	0	4,989286	1,5
	1	-127,2792	0	4,994643	1,5
	1	0	0	5	1,5
	1	127,27922	0	5,005357	1,5
	1	-105,4416	0	5,010714	1,5
	1	21,837662	0	5,016071	1,5
	1	149,11688	0	5,021429	1,5
-30 °	1	6,49E-14	0	4,978571	1,5
	1	90	0	4,983929	1,5
	1	-180	0	4,989286	1,5
	1	-90	0	4,994643	1,5
	1	0	0	5	1,5
	1	90	0	5,005357	1,5
	1	180	0	5,010714	1,5
	1	-90	0	5,016071	1,5
	1	-6,49E-14	0	5,021429	1,5

-15 °	1	173,65029	0	4,978571	1,5
	1	-139,7623	0	4,983929	1,5
	1	-93,17486	0	4,989286	1,5
	1	-46,58743	0	4,994643	1,5
	1	0	0	5	1,5
	1	46,587428	0	5,005357	1,5
	1	93,174856	0	5,010714	1,5
	1	139,76228	0	5,016071	1,5
	1	-173,6503	0	5,021429	1,5
15 °	1	-173,6503	0	4,978571	1,5
	1	139,76228	0	4,983929	1,5
	1	93,174856	0	4,989286	1,5
	1	46,587428	0	4,994643	1,5
	1	0	0	5	1,5
	1	-46,58743	0	5,005357	1,5
	1	-93,17486	0	5,010714	1,5
	1	-139,7623	0	5,016071	1,5
	1	173,65029	0	5,021429	1,5
30 °	1	-6,49E-14	0	4,978571	1,5
	1	-90	0	4,983929	1,5
	1	180	0	4,989286	1,5
	1	90	0	4,994643	1,5
	1	0	0	5	1,5
	1	-90	0	5,005357	1,5
	1	-180	0	5,010714	1,5
	1	90	0	5,016071	1,5
	1	6,49E-14	0	5,021429	1,5
45 °	1	1,49E+02	0	4,978571	1,5
	1	21,837662	0	4,983929	1,5
	1	-105,4416	0	4,989286	1,5
	1	127,27922	0	4,994643	1,5
	1	0	0	5	1,5
	1	-127,2792	0	5,005357	1,5
	1	105,44156	0	5,010714	1,5
	1	-21,83766	0	5,016071	1,5
	1	-1,49E+02	0	5,021429	1,5

60°	1	-9,65E+01	0	4,978571	1,5
	1	107,65372	0	4,983929	1,5
	1	-48,23085	0	4,989286	1,5
	1	155,88457	0	4,994643	1,5
	1	0	0	5	1,5
	1	-155,8846	0	5,005357	1,5
	1	48,230855	0	5,010714	1,5
	1	-107,6537	0	5,016071	1,5
	1	9,65E+01	0	5,021429	1,5
75 °	1	-2,45E+01	0	4,978571	1,5
	1	161,59995	0	4,983929	1,5
	1	-12,2667	0	4,989286	1,5
	1	173,86665	0	4,994643	1,5
	1	0	0	5	1,5
	1	-173,8666	0	5,005357	1,5
	1	12,266703	0	5,010714	1,5
	1	-161,5999	0	5,016071	1,5
	1	2,45E+01	0	5,021429	1,5

Tanto la Figura 3.6 como la Figura 3.7 muestran las características más relevantes de la antena direccional simulada con un ángulo de 0°.

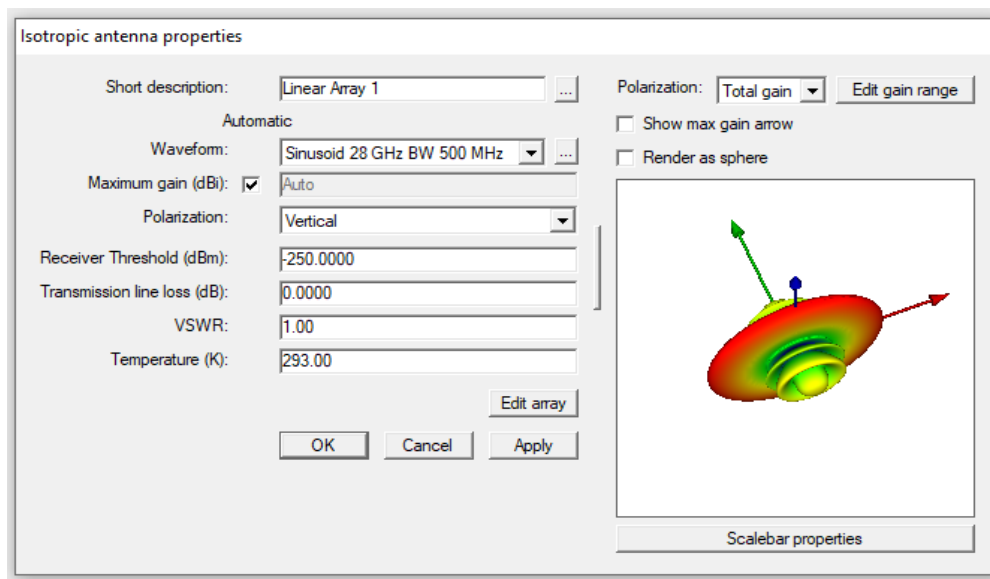


Figura 3.6: Características de la antena direccional simulada.

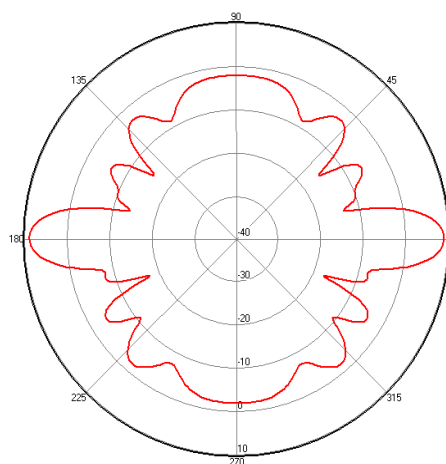


Figura 3.7: Patrón de radiación de antena direccional.

3.5 Diseño y despliegue del sistema de comunicaciones que opera en la banda de 28 GHz.

En los escenarios se sitúa un transmisor y 19 receptores, estos últimos colocados ya sea en una sola pared o distribuidos en 3 paredes. Tanto el Tx como los Rxs están colocados a la misma altura de 1.5 m sobre el suelo. Además, el transmisor fue fijado en la mitad de la pared de 10m de dimensión mientras que la distancia de separación entre los receptores instalados

depende del escenario. Para el efecto se crearon 19 dispositivos con iguales características como se observa en la Figura 3.8.

Wireless InSite - Main: (Untitled Project) [K:\...\SetupBase\010.setup !]

ject Edit View Help

Features Images Study areas Transmitters Receivers Comm. systems Materials Antennas Waveforms Requested output Output

Act...	Visible	Description	Type	No. poi...	Spac...	Antenna	Waveform	Collect...	Boundi...	Filename
lo	No	010 Rx00	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
lo	No	010 Rx01	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
lo	No	010 Rx02	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
'es	Yes	010 Rx03	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
lo	No	010 Rx04	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
'es	Yes	010 Rx05	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
lo	No	010 Rx06	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
'es	Yes	010 Rx07	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
lo	No	010 Rx08	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
'es	Yes	010 Rx09	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
lo	No	010 Rx10	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
'es	Yes	010 Rx11	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
lo	No	010 Rx12	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
'es	Yes	010 Rx13	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
lo	No	010 Rx14	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
'es	Yes	010 Rx15	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
lo	No	010 Rx16	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
'es	Yes	010 Rx17	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
lo	No	010 Rx18	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
'es	Yes	010 Rx19	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
lo	No	010 Rx20	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
'es	Yes	010 Rx21	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
'es	Yes	010 Rx22	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
'es	Yes	010 Rx23	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
'es	Yes	010 Rx24	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
'es	Yes	010 Rx25	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
'es	Yes	010 Rx26	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
'es	Yes	010 Rx27	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
'es	Yes	010 Rx28	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
'es	Yes	010 Rx29	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx
'es	Yes	010 Rx30	points	1	N/A	Half Wave Dipole	Sinusoid 28 GHz BW 500 MHz	Auto	Off	K:\ProyVero_OK_Ene21\SetupBase\010.rx

Figura 3.8: Creación de las antenas receptoras en Wireless Insite.

La simulación del sistema de comunicaciones fue realizada bajo las siguientes configuraciones:

- 1) Configuración uno: dipolo de media onda en el trasmisor y en cada uno de los 19 receptores que se encuentran colocados frente a frente en paredes opuestas. Las antenas tienen un patrón de radiación omnidireccional. La separación de los Rxs es de 0.5 m aproximadamente. En este caso, la distancia mínima entre un Tx y Rx es de 20 m.
- 2) Configuración dos: Un transmisor y 19 receptores colocados en las paredes laterales (6+6) y en la pared opuesta al transmisor (7). Arreglo de antena ULA-9 en Tx con pasos de rotación de 15° de forma horaria y antihoraria y dipolo de media longitud de onda en cada uno de los Rxs.

El resumen de las configuraciones principales del sistema de comunicaciones se detalla en las Tabla 3.2 y 3.3.

Tabla 3.2 Parámetros del escenario simulado en la primera configuración.

Parámetro	Valor
Frecuencia	28 GHz
Ancho de banda	500 MHz
Tipo de onda	Sinusoidal
Antena Tx	Dipolo media onda (Irradiación tipo Omnidireccional)
Antena Rx	Dipolo media onda (Irradiación tipo Omnidireccional)
Ganancia de las antenas	3 dBi
Número de Txs	1
Número de Rxs	19

Tabla 3.3 Parámetros del escenario simulado en la segunda configuración.

Parámetro	Valor
Frecuencia	28 GHz
Ancho de banda	500 MHz
Tipo de onda	Sinusoidal
Arreglo de antena Tx	ULA-9 Direccional
Antena Rx	Dipolo media onda (Irradiación tipo Omnidireccional)
Número de Txs	1
Número de Rxs	19
Ganancia de las antenas	Tx: $10\log_{10}(9) + 0 = -9.54 \text{ dB}$ Rx: 3 dB
Tx azimuth, elevación	Azimuth: -75° a 75° (pasos de 15°) Elevación: 0°

En las Figuras 3.9 y 3.10 se visualizan los diseños finales de los ambientes de simulación basados en las configuraciones uno y dos.

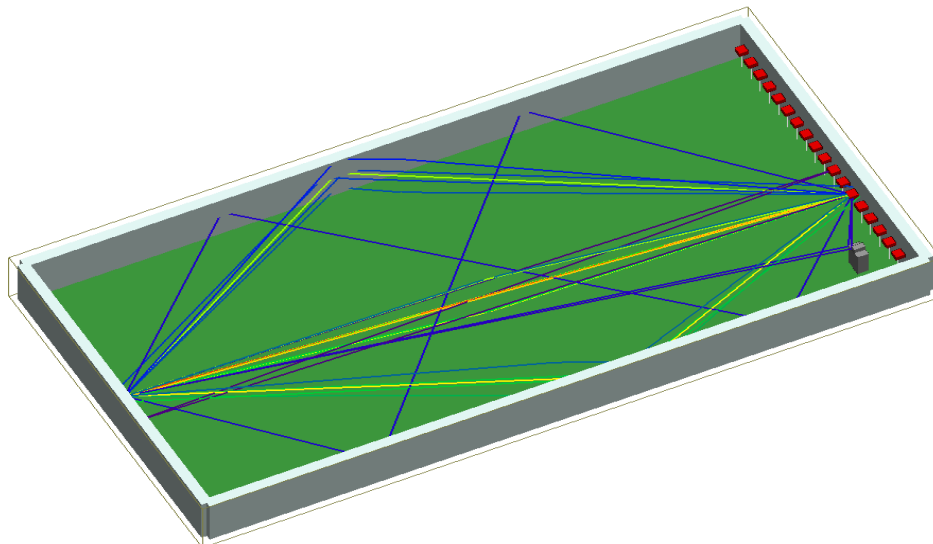


Figura 3.9: Primera configuración del ambiente de interiores (enlace Tx-Rx07)

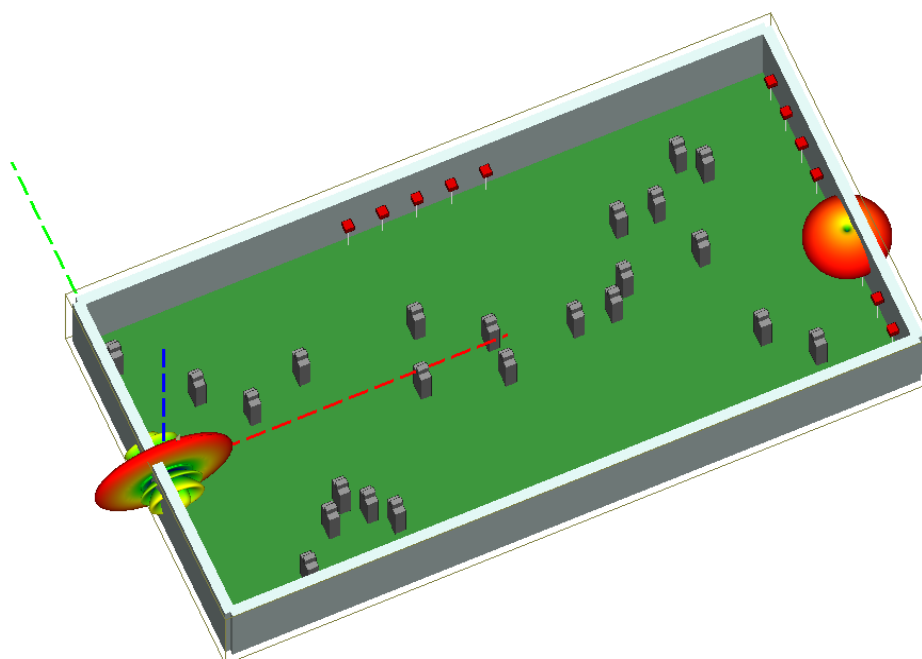


Figura 3.9: Segunda configuración del ambiente de interiores

La Figura 3.9 presenta un ambiente de interiores que emplea un transmisor equipado con arreglo de antena direccional ULA-9 y 19 receptores con dipolo

de media onda dispuestos en 3 paredes del escenario. También se observa el patrón de irradiación de la antena transmisora (0°) y múltiples personas ubicadas cerca de los dispositivos.

3.6 Selección de datos específicos para entrenamiento de algoritmos de aprendizaje automático supervisado: redes neuronales y regresión lineal.

Los datos generados a través del simulador de Ray Tracing fue utilizada para el entrenamiento de los algoritmos. Se consideran los niveles de potencias recibidas en los 19 receptores y las coordenadas reales del sujeto como características (features) y etiquetas (labels) del sistema. La Tabla 3.4 muestra la estructura del conjunto de datos para el primer escenario. Para el segundo escenario se generaron 11 tablas similares (una por cada ángulo de la antena direccional) con el mismo formato. El también llamado dataset forma una matriz con los valores ya mencionados y las coordenadas en x, y de las posiciones reales de los sujetos [22].

Tabla 3.4 Estructura del Dataset.

ID HB	PRx1	PRx2	PRx3	PRx...	PRx19	POS X	POS Y
38	3.706	1.832	9.120	...	5.370	14.4	7.75
23	2.541	3.117	3.917	...	1.114	12.2	8.85
.....
130	2.992	3.093	2.098	...	6.501	28	12.15

Donde:

ID: Número de identificación del sujeto

PRx...: potencias de recepción expresadas en [mw]

Pos X: Posición del sujeto en el eje X

Pos Y: Posición del sujeto en el eje y

Para realizar la selección, se formaron los siguientes conjuntos de datos:

- El conjunto de entrenamiento (training data) es subconjunto de los datos, que se utiliza para entrenar el modelo (para reconocer patrones) con sus características etiquetadas.
- El conjunto de prueba (testing data) utiliza las características sin etiquetar para evaluar la capacidad de generalización o qué tan precisa sea la predicción del modelo entrenado.

Una estrategia comúnmente usada, es dividir el dataset de manera aleatoria o secuencial en una proporción de 70-80% para entrenamiento y entre 30-20% para la validación o pruebas. [26] En el presente caso, de los 248 sujetos situados dentro de la habitación, se tomó en consideración el 70% de los datos (174 individuos aproximadamente) para el proceso de entrenamiento y con el 30% restante se realizó el proceso de prueba y de validación.

3.7 Diseño de método de detección pasiva de individuo usando algoritmos de aprendizaje automático supervisado: redes neuronales y regresión lineal.

Los algoritmos de aprendizaje automático utilizan métodos computacionales para "aprender" información directamente de los datos sin depender de una ecuación predeterminada como modelo.[23] El aprendizaje automático emplea dos tipos de técnicas: aprendizaje supervisado y aprendizaje no supervisado. [24] La primera toma un conjunto de datos de entrada y salida conocidos para que pueda predecir salidas futuras, mientras que la segunda encuentra patrones ocultos o estructuras intrínsecas en los datos de entrada.

Para efectos de estudio, se eligió el aprendizaje supervisado ya que se necesita entrenar un modelo para efectuar la predicción de la ubicación de la persona en un ambiente interior (indoor).

La selección del mejor método se relaciona con el menor error cuadrático medio. Se requirió de algunas pruebas (varios entrenamientos) para encontrar el mejor modelo que se ajuste a los datos. Seleccionar el modelo apropiado requiere de pruebas y mediciones de error. Si un algoritmo no es lo suficientemente preciso, se prueba otro para revisar si cumple con las expectativas. Siguiendo este procedimiento, se probaron tres algoritmos mediante dos aplicaciones de aprendizaje automático supervisado en Matlab. Estos algoritmos son: Bayesian Regularization (BR), Levenberg Marquadt (LM) y Ensemble Bagged Trees (EBT).

El flujo de trabajo sobre el cual operan los algoritmos de aprendizaje automático involucra las siguientes fases: a) selección de los datos de entrenamiento y prueba, b) selección del algoritmo de aprendizaje, y c) entrenamiento y evaluación de los modelos de forma iterativa [27]. A continuación, se muestra el flujo de trabajo del aprendizaje automático supervisado (Ver Figura 3.16).

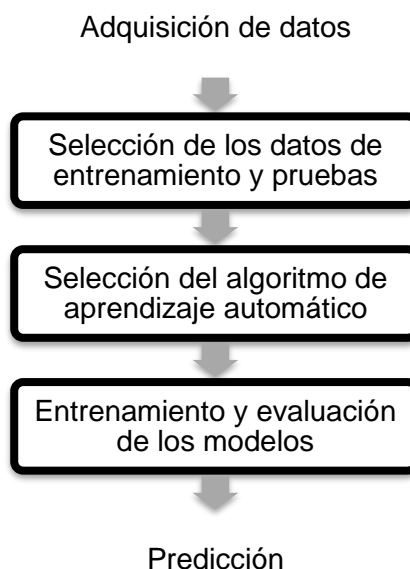


Figura 3.16: Metodología de los algoritmos de aprendizaje automático supervisado.

Todo el flujo de trabajo indicado se pone en marcha con ayuda de dos aplicaciones de redes neuronales y regresión lineal de MATLAB. Inicialmente, se utilizó la aplicación neuronal net fitting bajo el siguiente procedimiento:

1. Usando MATLAB, se exporta como archivo .mat el conjunto de datos de posiciones de personas y potencias recibidas en los receptores obtenidos usando las simulaciones de Ray Tracing (primera configuración). El dataset se crea como una matriz de dimensiones $248 \times (1+19+2)$. Adicional a los 19 niveles de potencia se incluye 1 columna de identificación de cada RX y las 2 coordenadas de la posición real x e y de cada una de las 248 personas (Tabla 3.4)
2. Mediante la aplicación neuronal net fitting de MATLAB se diseñó una red neuronal multicapa de 19 entradas (referente a los 19 valores de potencia de los receptores), 10 neuronas en la capa oculta, y 2 salidas (referentes a la posición X e Y).
3. Se diseña cada una de las capas de estos dos métodos. La configuración de las 3 capas se describe a continuación:
 - Input layer: tiene 19 entradas correspondiente a los 19 valores de potencia de los receptores (features). Los pesos del sistema o entradas están siendo representados por los números de identificación (ID) de los sujetos mientras que los niveles de potencia recibidos (RSS) en los receptores forman parte de los features o características que serán analizadas mediante valores de autocorrelación.
 - Hidden layer: en esta capa se modificaron el número de neuronas para comparar el desempeño de la red neuronal. El número de neuronas definido para esta capa es de diez ya que se observó que a medida que se incrementaba el número de neuronas también aumentaba el número de cálculos que se debe realizar.
 - Output layer: capa de salida que representa la posición en x, y de los sujetos (2 salidas). Las salidas mostrarán los valores estimados de x, y de las personas.

La Figura 3.11 muestra la arquitectura del sistema en la cual se distingue la configuración de cada una de las capas.

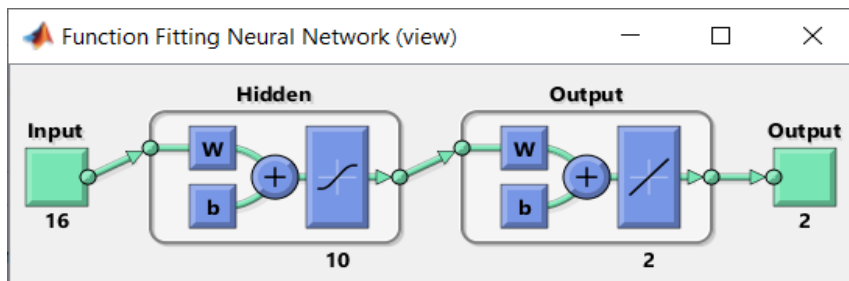


Figura 3.11: Diseño de la red neuronal propuesta.

- Se seleccionan las redes neuronales de aprendizaje automático supervisado: Levenberg Marquardt (LM) y Bayesian Regularization (BR) tomando en consideración los tipos de datos. Dentro de las principales características de LM destaca su rapidez para el entrenamiento de las redes neuronales y la optimización de su rendimiento en función de sus iteraciones a través de la técnica del gradiente. Por otro lado, BR presenta un mayor coste computacional y disminuye los errores de estimación al usar un modelo de precisión generalizado.

Además, se independizan las estimaciones por cada coordenada x , y ya que no se encuentran correlacionadas. Las Figuras 3.12, 3.13, 3.14 y 3.15 presentan las configuraciones realizadas con cada red neuronal para ambas coordenadas.

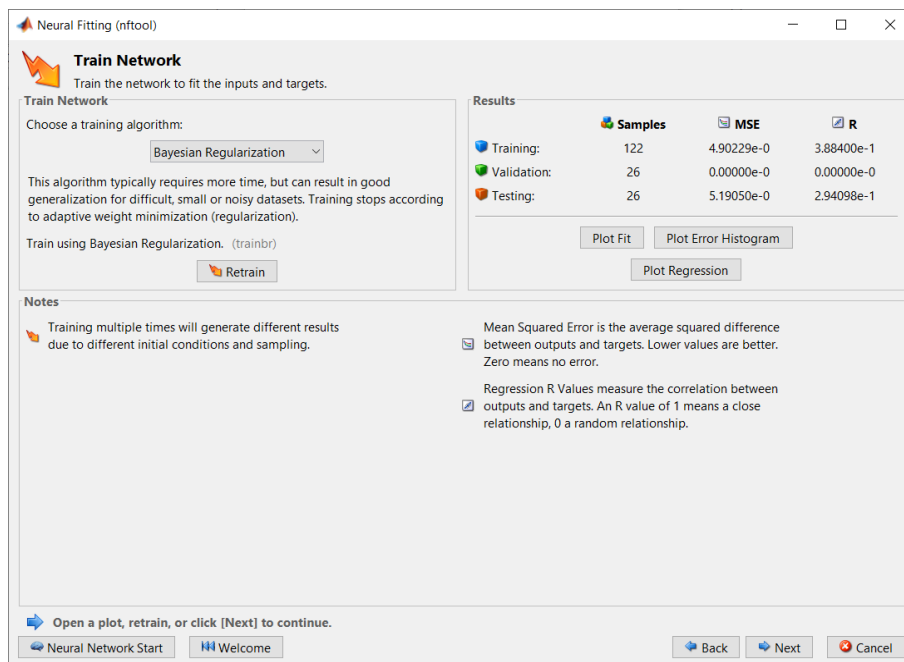


Figura 3.12: Configuración de los parámetros de entrenamiento con Bayesian Regularization para la coordenada y.

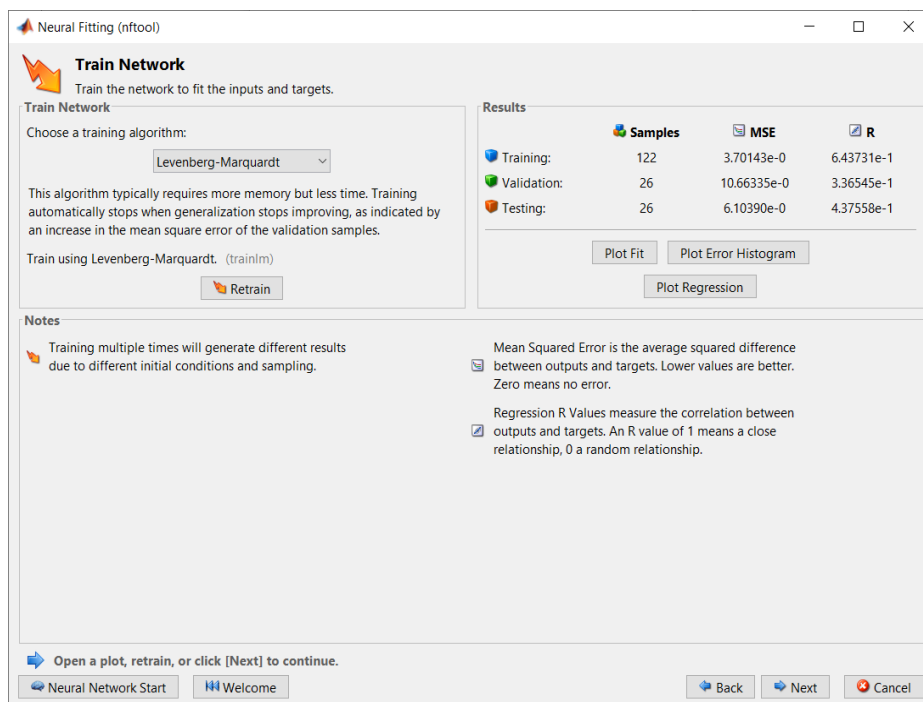


Figura 3.13: Configuración de los parámetros de entrenamiento con Levenberg-Marquardt para la coordenada y

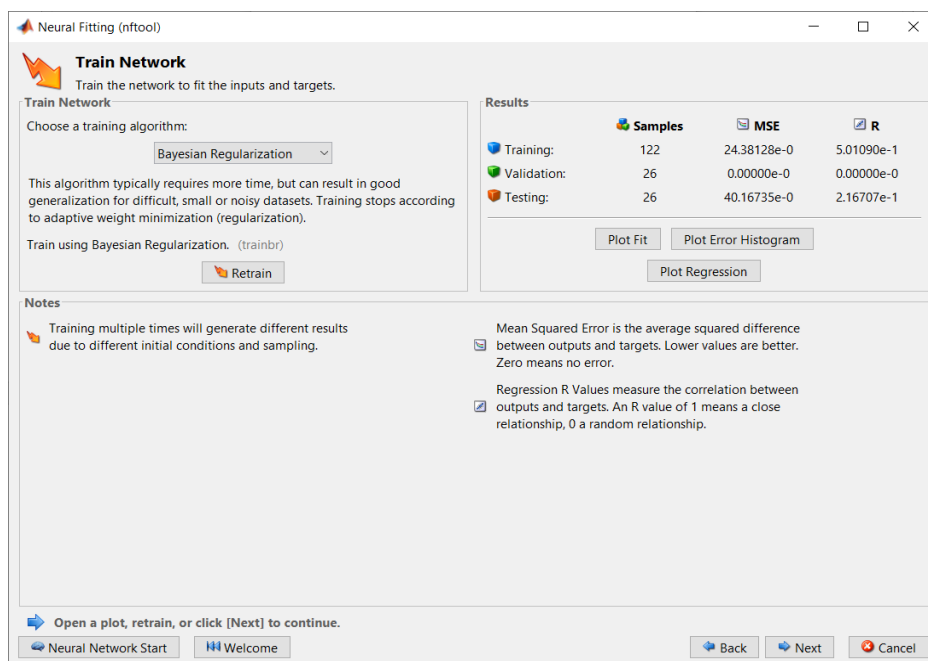


Figura 3.14: Configuración de los parámetros de entrenamiento con Bayesian Regularization para la coordenada x

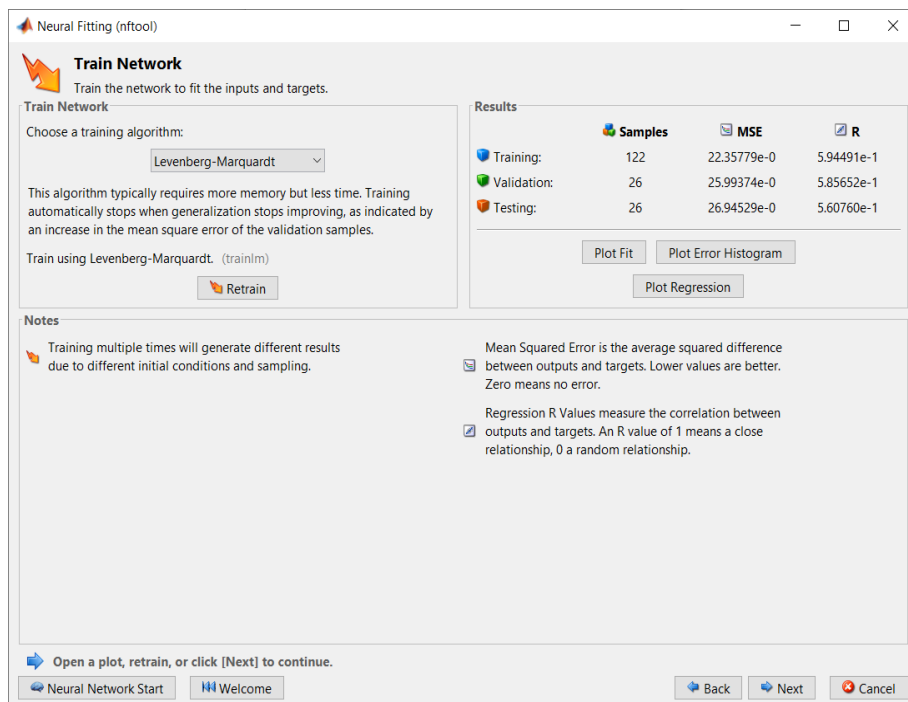


Figura 3.15: Configuración de los parámetros de entrenamiento con Levenberg-Marquardt para la coordenada x

5. Se estima la precisión del método a través de la métrica del error cuadrático medio. Su análisis será presentado en el siguiente capítulo.

Debido a que los resultados obtenidos con el algoritmo de redes neuronales presentaron una precisión relativamente baja, se resolvió investigar otros tipos de algoritmos de aprendizaje supervisado especializados en regresión lineal. Entre ellos se destacan los siguientes (Ver Figura 3.16):

1.1	☆ Linear Regression	Last change: Linear
1.2	☆ Linear Regression	Last change: Interactions Linear
1.3	☆ Linear Regression	Last change: Robust Linear
1.4	☆ Stepwise Linear Regression	Last change: Stepwise Linear
1.5	☆ Tree	Last change: Fine Tree
1.6	☆ Tree	Last change: Medium Tree
1.7	☆ Tree	Last change: Coarse Tree
1.8	☆ SVM	Last change: Linear SVM
1.9	☆ SVM	Last change: Quadratic SVM
1.10	☆ SVM	Last change: Cubic SVM
1.11	☆ SVM	Last change: Fine Gaussian SVM
1.12	☆ SVM	Last change: Medium Gaussian SVM
1.13	☆ SVM	Last change: Coarse Gaussian SVM
1.14	☆ Ensemble	Last change: Boosted Trees
1.15	☆ Ensemble	Last change: Bagged Trees
1.16	☆ Gaussian Process Regression	Last change: Squared Exponential GPR

Figura 3.16: Entrenamiento con todos los algoritmos de regresión lineal disponibles de las herramientas.

Existen muchos modelos que pueden ser elegidos: modelos de regresión lineal (LR), árboles de regresión, modelos de regresión de procesos gaussianos, máquinas de vectores de soporte (SVM) y conjuntos de árboles de regresión (Ensemble Bagged Trees). Debido a que la posición x , y son de naturaleza continua se decidió por estos algoritmos de regresión logística.

Las técnicas de regresión en su mayoría difieren según la cantidad de variables independientes y el tipo de relación entre las variables independientes y dependientes [27]. Todos los entrenamientos realizados estos algoritmos en la herramienta Regression Learner utilizan el siguiente procedimiento para su diseño:

1. Se concatena cada archivo del conjunto de datos adquiridos a partir de las simulaciones de Ray Tracing del segundo escenario. Por cada valor de ángulo de rotación de la antena direccional ULA-9, existe un valor de potencia recibida. Los valores de potencias recibidas en los receptores crean una matriz de dimensiones igual a 248x209.
2. Se divide los datos en 2 subconjuntos de entrenamiento y, validación y pruebas con los siguientes porcentajes:
 - El 70% de los datos se utilizará para entrenamiento (174 valores de potencias recibidas)
 - El 30% de los datos será empleado para validar que la red se está generalizando.
3. Dependiendo de las características (features) definidas, la aplicación asignará automáticamente quienes actuarán como predictores o como salidas. Las características serán comparadas con los valores estimados por el algoritmo que presente el menor error de predicción.
4. Luego de dar inicio al entrenamiento, se selecciona el modelo que dentro de la lista tenga el menor error cuadrático medio. En la Figura 3.17 se distingue Ensemble Bagged Trees como el algoritmo más preciso.

1.1 ☆ Linear Regression	RMSE: 8.489
Last change: Linear	114/114 features
1.2 ☆ Linear Regression	Training <input type="checkbox"/> 114/114 features
Last change: Interactions Linear	
1.3 ☆ Linear Regression	RMSE: 8.49
Last change: Robust Linear	114/114 features
1.4 ☆ Stepwise Linear Regression	Training <input type="checkbox"/> 114/114 features
Last change: Stepwise Linear	
1.5 ☆ Tree	RMSE: 2.3865
Last change: Fine Tree	114/114 features
1.6 ☆ Tree	RMSE: 2.2014
Last change: Medium Tree	114/114 features
1.7 ☆ Tree	RMSE: 2.2702
Last change: Coarse Tree	114/114 features
1.8 ☆ SVM	RMSE: 2.9669
Last change: Linear SVM	114/114 features
1.9 ☆ SVM	RMSE: 5.7409
Last change: Quadratic SVM	114/114 features
1.10 ☆ SVM	RMSE: 20.251
Last change: Cubic SVM	114/114 features
1.11 ☆ SVM	RMSE: 2.4203
Last change: Fine Gaussian SVM	114/114 features
1.12 ☆ SVM	RMSE: 1.9285
Last change: Medium Gaussian SVM	114/114 features
1.13 ☆ SVM	RMSE: 2.1181
Last change: Coarse Gaussian SVM	114/114 features
1.14 ☆ Ensemble	RMSE: 1.7112
Last change: Boosted Trees	114/114 features
1.15 ☆ Ensemble	RMSE: 1.6939
Last change: Bagged Trees	114/114 features
1.16 ☆ Gaussian Process Regression	RMSE: 2.2739
Last change: Squared Exponential GPR	114/114 features
1.17 ☆ Gaussian Process Regression	RMSE: 2.2762
Last change: Matern 5/2 GPR	114/114 features
1.18 ☆ Gaussian Process Regression	RMSE: 2.1848
Last change: Exponential GPR	114/114 features
1.19 ☆ Gaussian Process Regression	RMSE: 2.195
Last change: Rational Quadratic GPR	114/114 features

Figura 3.17: Entrenamiento con todos los algoritmos de regresión logística.

5. Se genera la función del algoritmo que genera MATLAB a partir del modelo que arroje un menor error cuadrático medio. Dentro de MATLAB la construcción del modelo de predicción se considera un proceso iterativo hasta que se acepta el modelo resultante. [26].
6. Luego de obtener la función del mejor algoritmo de regresión (EBT), se generan los resultados de las predicciones realizadas a partir del modelo escogido, los cuales serán analizados en el próximo capítulo.

CAPÍTULO 4

4 ANÁLISIS DE LOS RESULTADOS

4.1 Análisis de las simulaciones

En esta sección se muestran los resultados de las simulaciones de los escenarios en un ambiente de interiores usando el software Wireless Insite y los algoritmos de aprendizaje automático supervisado. En base a los datos de cada configuración se procede a la interpretación de la precisión del método de detección y al análisis de la magnitud del error. Se comparan los valores de las coordenadas x , y reales y estimados.

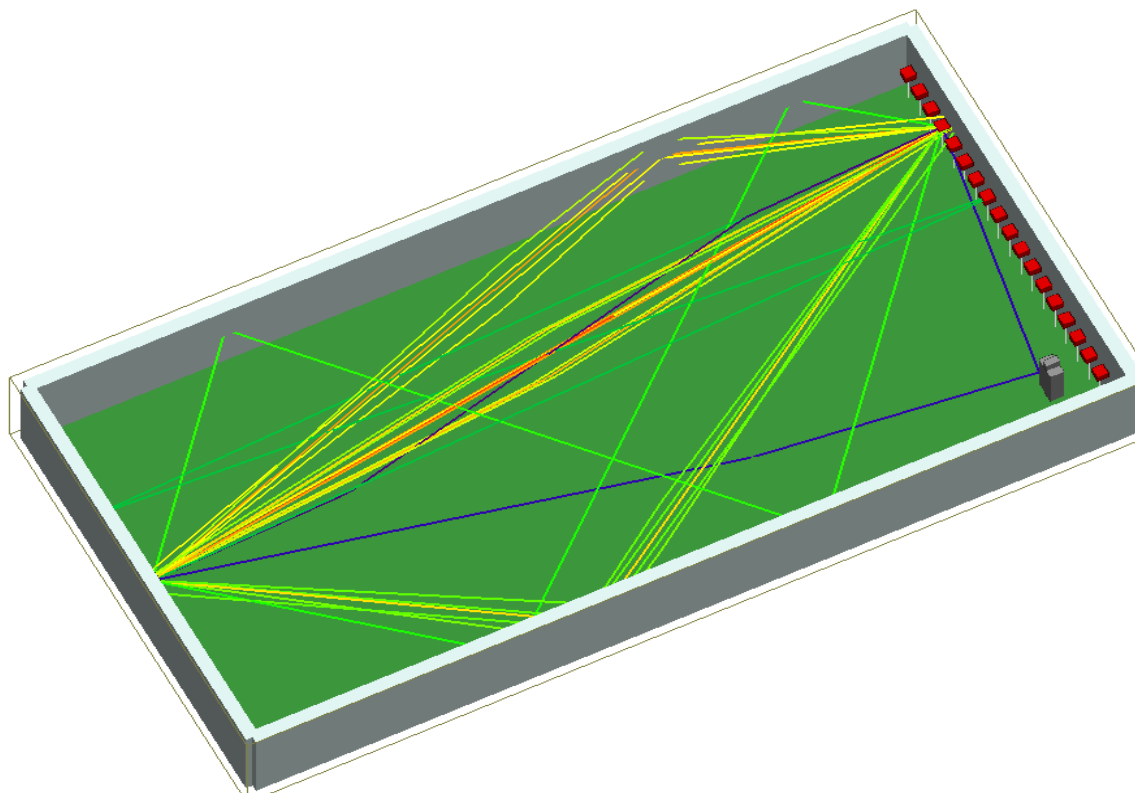
Finalmente, se presenta una comparación de los métodos propuestos con el objetivo de determinar el mejor algoritmo de localización basado en un aprendizaje automático supervisado.

4.2 Análisis del ambiente de propagación indoor usando Wireless Insite

El software de simulación de propagación en un ambiente de interiores se basa en Ray Tracing usando algoritmos propios. La herramienta permite visualizar con diferentes colores los rayos trazados correspondientes a los diferentes pasos de propagación. Los rayos que tienen un color (azul) más oscuro simbolizan niveles bajos de RSS a diferencia de las trayectorias de color rojo que muestran lo contrario. Las Figuras 4.1, 4.2 y 4.3 presentan los diseños de las dos configuraciones propuestas en el capítulo 3.

Específicamente, la Figura 4.1 muestra las simulaciones de Ray Tracing en un ambiente cerrado. La comunicación entre el Tx y cualquiera de los Rx en la banda de 28 GHz se realiza a través de enlaces directos (LOS) y enlaces indirectos mediante reflexiones en paredes, tumbado o piso y difracciones en los bordes verticales u horizontales. La presencia de las personas introduce difracciones y bloqueos en los enlaces. En la configuración (primera) de la figura, tanto el Tx y los 19 Rx están equipados con dipolos de media onda e instalados frente a frente. Para cada una de las posiciones de la persona se simuló el sistema de comunicaciones obteniéndose la potencia recibida en

cada uno de los 19 RXs con lo que generó una tabla de 248 filas (número de personas) x 21 columnas (19 potencias + las coordenadas (x,y)).



**Figura 4.1: Ambiente de propagación para la primera configuración
(enlace Tx-Rx16)**

En la Figura 4.2 se visualiza el gráfico de Ray Tracing para un ambiente de interiores con arreglo de antenas direccionales ULA-9 y 19 receptores¹ con antenas isotrópicas (patrón de irradiación omnidireccional) dispuestos en las paredes laterales (5 Rxs + 5 RXs) y en la pared frontal (9 RXs) al transmisor. La presencia de una persona genera bloqueos y difracciones adicionales de las señales de RF en la banda de 28 GHz. En este caso se generaron 11 tablas de 248x21 correspondientes a las potencias recibidas en cada uno de

¹ En la práctica se aumentó el número de receptores para instalarlos en las paredes laterales y se desactivaron algunos Rxs de la pared frontal al Tx (total 19 Rxs)

ellos 19 Rxs cuando la antena direccional del Tx está orientada en cada uno de los ángulos (-75° , -60° , -45° , -30° , -15° , 0° , 15° , 30° , 45° , 60° , 75°)

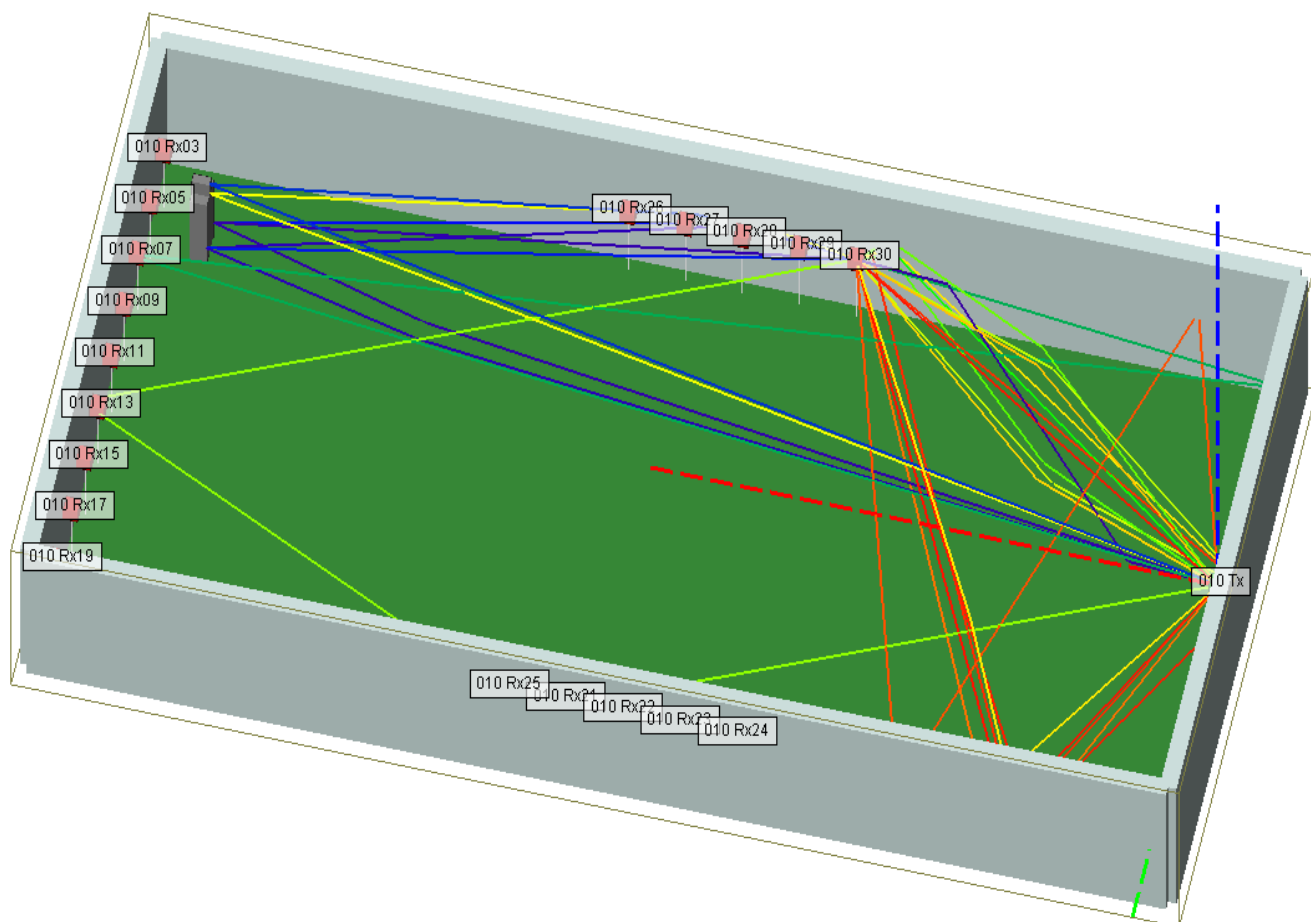


Figura 4.2: Ambiente de propagación para la segunda configuración.

La figura 4.3 muestra el diseño de un ambiente de interiores con presencia de una persona en una configuración que incluye un transmisor equipado con un arreglo de antena ULA-9 orientado en la dirección -30° .

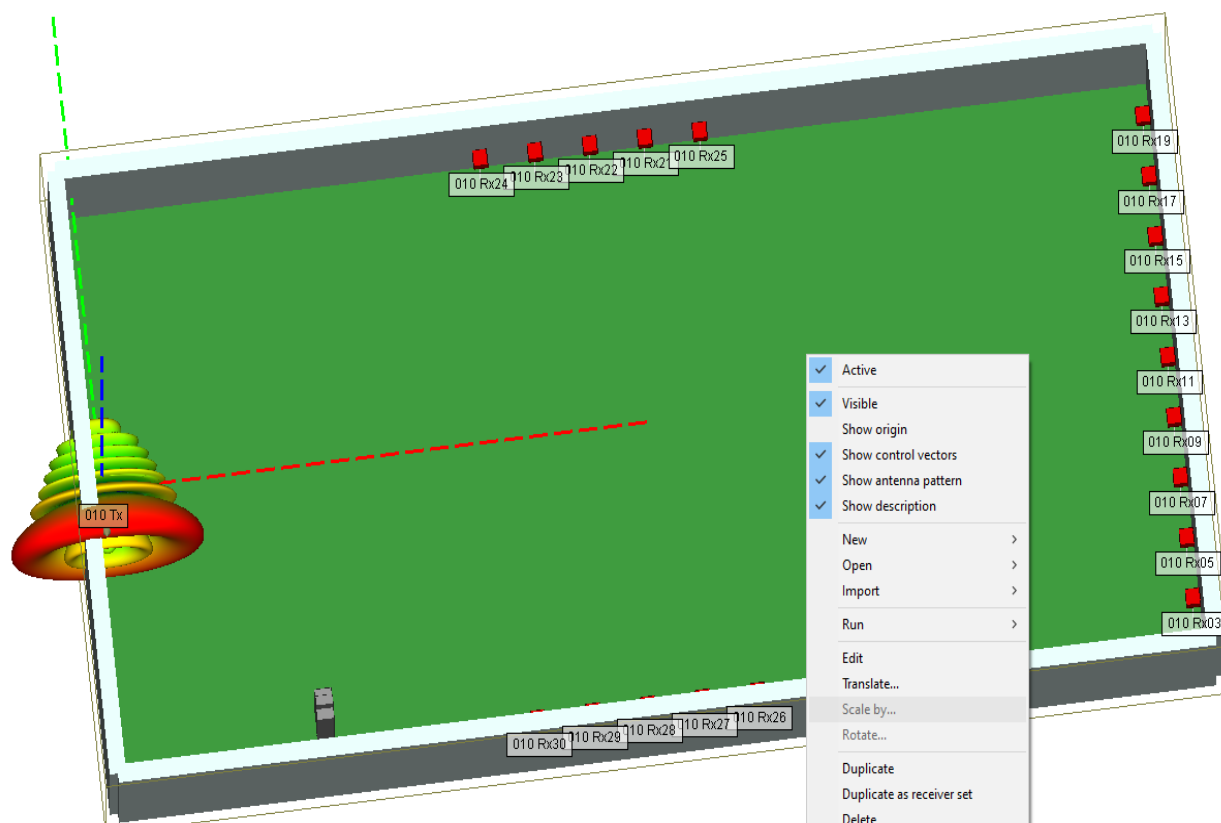


Figura 4.3: Ambiente de propagación para la segunda configuración.

4.3 Análisis de los resultados según la técnica de detección de posición del individuo basada en aprendizaje automático supervisado.

En esta sección se analizan los resultados obtenidos usando los algoritmos de aprendizaje automático supervisado redes neuronales y regresión lineal aplicados sobre los datos de propagación para optimizar el error de posicionamiento de la persona. Se muestra la relación entre la posición real del sujeto y la predicción de la posición obtenida en el entrenamiento. Las primeras predicciones fueron efectuadas con los datos del primer escenario. Para este caso, se investigó la aplicación de las redes neuronales: Bayesian Regularization (BR) y Levenberg-Marquardt (LM)².

² El lector se puede referir a https://www.neuraldesigner.com/blog/5_algorithms_to_train_a_neural_network#Levenberg-Marquardt para la revisión de los detalles de funcionamiento de estos algoritmos respectivamente

Red Bayesian Regularization

La Figura 4.4 presenta los resultados del entrenamiento efectuado con el algoritmo Bayesian Regularization. Se puede ver que el eje X representa el número de posiciones de personas consideradas para el entrenamiento (aproximadamente 174 que corresponde al 70% de los datos) mientras que el eje Y corresponde al eje de referencia de las dimensiones del ambiente de simulación (20m x 10m), respectivamente.

Los datos de los gráficos representan el error promedio del método para ambas coordenadas. El error obtenido durante el entrenamiento fue de 5.203 m ($5.203/20=26\%$) para la coordenada x y de 1.99 m ($1.99/10=19.9\%$) para la coordenada y.

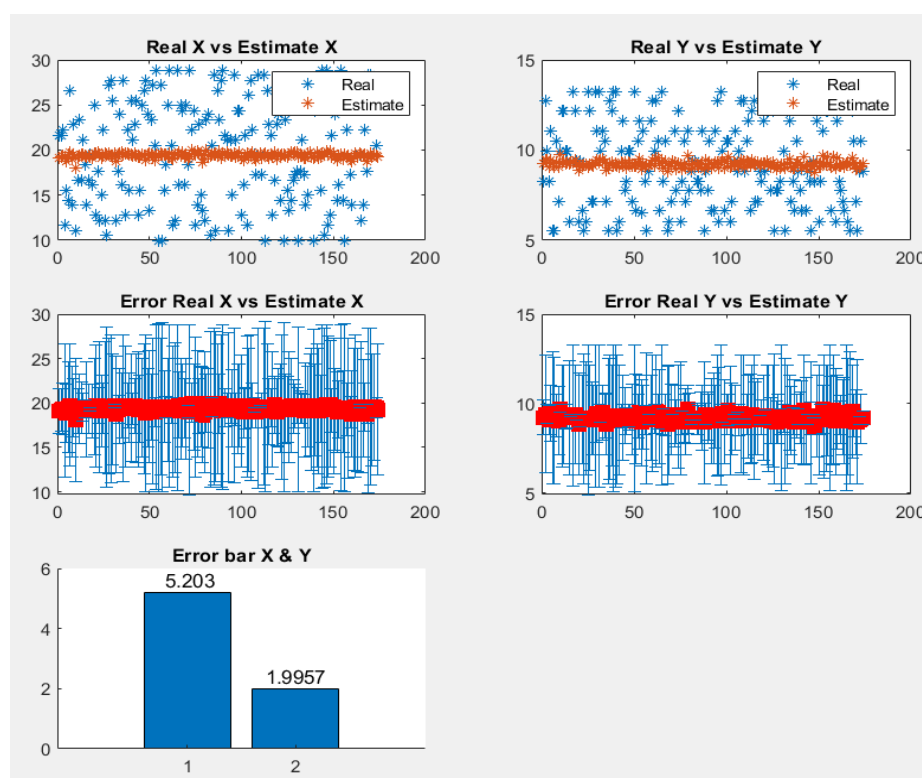


Figura 4.4: RMSE utilizando Training Data_Bayesian Regularization

Por otro lado, en la Figura 4.5 se muestran los resultados de clasificación con los datos de validación. De la misma manera, el eje de las x corresponde a la cantidad de muestras de posiciones de personas (aproximadamente 75 que corresponde al 30% de los datos) y el eje y representa las dimensiones del ambiente de simulación (20m x 10m). La Figura permite visualizar el RSME obtenido. El error estimado para la coordenada en x, y es igual 24,3% (4.87/20) y 18,9% (1.89/10), respectivamente.

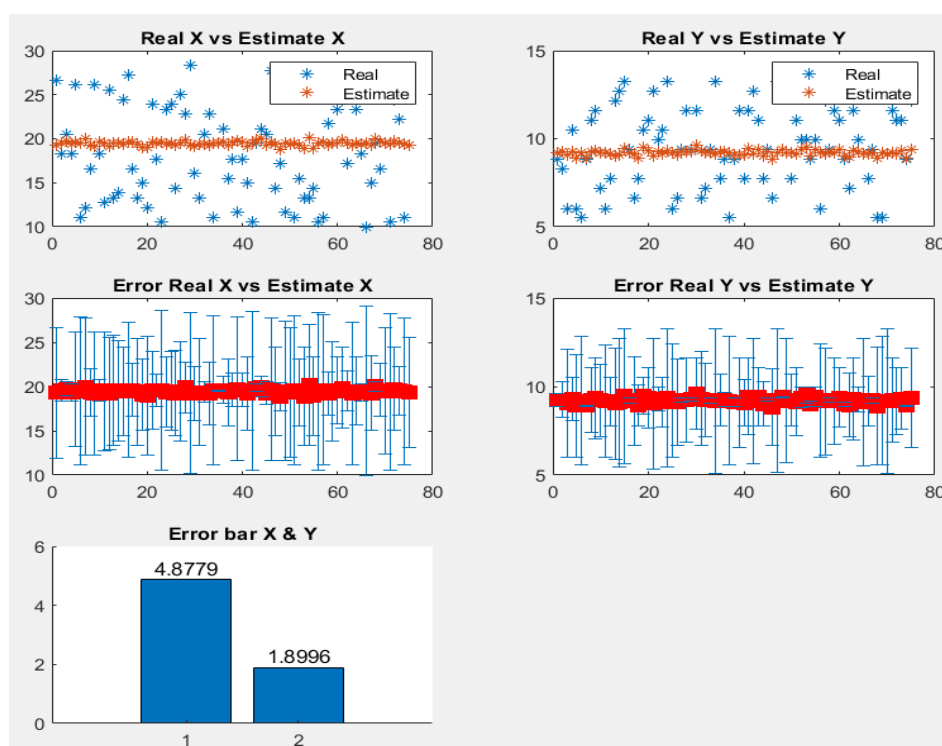


Figura 4.5: RMSE utilizando Testing Data_Bayesian Regularization

Red Levenberg-Marquard:

Bajo una metodología similar, se efectuaron pruebas con el conjunto de datos generados a partir del primer escenario de interiores usando el algoritmo Levenberg-Marquardt (LM). En este caso, el error estimado de los datos de clasificación tomados para el entrenamiento de la red neuronal fue de 24,7% (4.94/20) para la coordenada en x y de 16,6% (1.66/10) para la coordenada en y (Ver Figura 4.6).

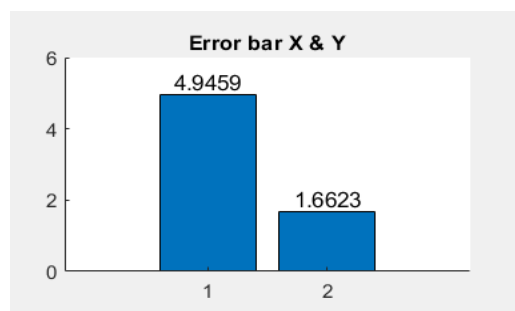


Figura 4.6: RMSE de detección utilizando Training Data_LM.

En la Figura 4.7 se aprecia los resultados de la clasificación con los datos de validación. La coordenada en x presenta un error estimado de 37% (7.54/20) y la coordenada en y un RSME de 38.5% (3.85/10).

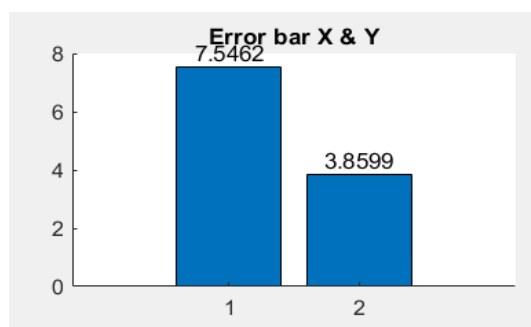


Figura 4.7: RMSE de detección utilizando Testing Data_LM

En algoritmos de aprendizaje automático supervisado, el error estimado para el training data representa el error con el que el dataset entrenó. Lo que se espera, es que el algoritmo generalice los casos con los que no ha entrenado. Por esta razón, se usa como referencia el error estimado con los datos de validación.

Una comparación del rendimiento y el coste computacional de cada una de las redes neuronales configuradas bajo los parámetros expuestos se encuentra que el método de Levenberg-Marquardt tiene un alto coste computacional respecto de Bayesian Regularization. Muchas veces el aumento de este coste computacional representa una mejora en la precisión del algoritmo sin embargo para este trabajo en particular, el aumento de neuronas efectuado en la capa oculta que fue realizado en pruebas anteriores, no produjo mejoras

en la predicción de la posición del sujeto. La mejor precisión obtenida con los métodos anteriores corresponde al 24.3% para x e 18.9% para y.

4.4 Estimación de la posición del objeto obstructor y su precisión.

En vista de que los resultados obtenidos con los dos primeros métodos no fueron los esperados, se estimó necesario investigar otros métodos. Se realizaron nuevas predicciones con los algoritmos de aprendizaje supervisado de regresión logística. Esta vez se reemplazó el dipolo de media onda del transmisor por un arreglo lineal (ULA-9). Se repitió el experimento de ubicar a la persona en cada una de las 248 posiciones cuando la antena direccional apuntaba en cada una de las 11 direcciones (-75°, -60°, -45°, -30°, -15°, 0°, 15°, 30°, 45°, 60°, 75°). Las Figuras 4.8 y 4.9 presentan los resultados obtenidos del RSME para cada coordenada x, y por cada algoritmo bajo el uso de la aplicación regression learner de Matlab.

Como en los casos anteriores, el desempeño de cada uno de los algoritmos de regresión logística anteriores fue determinado bajo la métrica del RSME para ambas coordenadas. Las mejores predicciones se dan a partir del método Ensemble Bagged Trees³ (EBT) en comparación con las redes neuronales utilizadas inicialmente.

La comparación realizada a partir de los valores de RSME (entrenados y estimados) permite conocer si el modelo generaliza los datos, esto es, si puede detectar casos diferentes a los que entrenó. Si se tiene un bajo RMSE de entrenamiento y un bajo RMSE de predicción, el algoritmo no generaliza los casos. Generalizar es saber que tan bueno es el algoritmo para detectar casos para los que no entrenó. Si el método es bueno tanto en el entrenamiento y la predicción, significa que es un algoritmo robusto.

³ El lector se puede referir a <https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/> para la revisión de los detalles de funcionamiento de estos algoritmos

1.1 ☆ Linear Regression	RMSE: 8.489
Last change: Linear	114/114 features
1.2 ☆ Linear Regression	Training  (x)
Last change: Interactions Linear	114/114 features
1.3 ☆ Linear Regression	RMSE: 8.49
Last change: Robust Linear	114/114 features
1.4 ☆ Stepwise Linear Regression	Training  (x)
Last change: Stepwise Linear	114/114 features
1.5 ☆ Tree	RMSE: 2.3865
Last change: Fine Tree	114/114 features
1.6 ☆ Tree	RMSE: 2.2014
Last change: Medium Tree	114/114 features
1.7 ☆ Tree	RMSE: 2.2702
Last change: Coarse Tree	114/114 features
1.8 ☆ SVM	RMSE: 2.9669
Last change: Linear SVM	114/114 features
1.9 ☆ SVM	RMSE: 5.7409
Last change: Quadratic SVM	114/114 features
1.10 ☆ SVM	RMSE: 20.251
Last change: Cubic SVM	114/114 features
1.11 ☆ SVM	RMSE: 2.4203
Last change: Fine Gaussian SVM	114/114 features
1.12 ☆ SVM	RMSE: 1.9285
Last change: Medium Gaussian SVM	114/114 features
1.13 ☆ SVM	RMSE: 2.1181
Last change: Coarse Gaussian SVM	114/114 features
1.14 ☆ Ensemble	RMSE: 1.7112
Last change: Boosted Trees	114/114 features
1.15 ☆ Ensemble	RMSE: 1.6939
Last change: Bagged Trees	114/114 features
1.16 ☆ Gaussian Process Regression	RMSE: 2.2739
Last change: Squared Exponential GPR	114/114 features
1.17 ☆ Gaussian Process Regression	RMSE: 2.2762
Last change: Matern 5/2 GPR	114/114 features
1.18 ☆ Gaussian Process Regression	RMSE: 2.1848
Last change: Exponential GPR	114/114 features
1.19 ☆ Gaussian Process Regression	RMSE: 2.195
Last change: Rational Quadratic GPR	114/114 features

Figura 4.8: RMSE de los algoritmos de aprendizaje automático supervisado: regresión lineal empleados para la coordenada en y

1.1 ☆ Linear Regression Last change: Linear	RMSE: 35.298 114/114 features
1.2 ☆ Linear Regression Last change: Interactions Linear	Canceled 114/114 features
1.3 ☆ Linear Regression Last change: Robust Linear	RMSE: 35.3 114/114 features
1.4 ☆ Stepwise Linear Regression Last change: Stepwise Linear	Canceled 114/114 features
1.5 ☆ Tree Last change: Fine Tree	RMSE: 5.7798 114/114 features
1.6 ☆ Tree Last change: Medium Tree	RMSE: 5.5491 114/114 features
1.7 ☆ Tree Last change: Coarse Tree	RMSE: 5.5896 114/114 features
1.8 ☆ SVM Last change: Linear SVM	RMSE: 6.6574 114/114 features
1.9 ☆ SVM Last change: Quadratic SVM	RMSE: 13.491 114/114 features
1.10 ☆ SVM Last change: Cubic SVM	RMSE: 38.62 114/114 features
1.11 ☆ SVM Last change: Fine Gaussian SVM	RMSE: 5.7556 114/114 features
1.12 ☆ SVM Last change: Medium Gaussian SVM	RMSE: 4.4467 114/114 features
1.13 ☆ SVM Last change: Coarse Gaussian SVM	RMSE: 5.1574 114/114 features
1.14 ☆ Ensemble Last change: Boosted Trees	RMSE: 3.9072 114/114 features
1.15 ☆ Ensemble Last change: Bagged Trees	RMSE: 3.9696 114/114 features
1.16 ☆ Gaussian Process Regression Last change: Squared Exponential GPR	RMSE: 5.8835 114/114 features
1.17 ☆ Gaussian Process Regression Last change: Matern 5/2 GPR	RMSE: 5.8602 114/114 features
1.18 ☆ Gaussian Process Regression Last change: Exponential GPR	RMSE: 5.7703 114/114 features
1.19 ☆ Gaussian Process Regression Last change: Rational Quadratic GPR	RMSE: 5.7199 114/114 features

Figura 4.9: RMSE de los algoritmos de aprendizaje automático supervisado: regresión lineal empleados para la coordenada en x.

Para obtener el error porcentual de predicción por cada coordenada, se promediaron todos los valores de predicción los cuales se resumen en la Tabla 4.1

Tabla 4.1 Resultados de los valores promedios de las predicciones realizadas bajo el algoritmo Ensemble Bagged Trees.

PROM ÁNGULOS	PRED RSME Y	PRED RSME X
TODOS LOS ÁNGULOS	1.69	3.90

La precisión obtenida bajo el método Ensemble Bagged Trees genera un RMSE de 3.90m (19.5%) para x y 1.69m (16.9%) para y en la estimación de las coordenadas. Los resultados demuestran que no existe una mejora sustancial en el error de detección en comparación con las redes neuronales probadas inicialmente.

A partir de estas predicciones, se procede a comparar los errores promedios obtenidos por cada método de aprendizaje automático supervisado probado. (Ver Tabla 4.2)

Tabla 4.2 RSME de cada algoritmo probado.

Coordenada	BR	LM	EBT
X	4.87 (24.35%)	7.54 (37.7%)	3.90 (19.5%)
Y	1.89 (18.9%)	3,85 (38.5%)	1.69 (16.9%)

Finalmente, la Tabla 4.2 permite establecer que la regresión logística realizada por el algoritmo Ensemble Bagged Trees es el mejor método de posicionamiento ya que la precisión del modelo es la mayor en comparación con los demás algoritmos de aprendizaje supervisado probados.

CONCLUSIONES

Para estimar la posición de una persona en forma pasiva, se investigó la variación de los niveles de potencia recibidos en un conjunto de receptores debido a la presencia de una persona. Para el efecto, se efectuaron cambios tanto en la ubicación de los receptores como en los tipos de antena transmisora operando en la banda de 28 GHz. Estos cambios permitieron generar conjuntos de datos a partir de simulaciones de Ray Tracing en escenarios de ambientes cerrados o interiores.

La generación de los datos se la debe automatizar (combinación de MATLAB y Wireless Insite) debido a la gran cantidad de simulaciones necesarias.

Los primeros diseños realizados a partir de las redes neuronales Bayesian Regularization y Levenberg-Marquardt presentan una resolución media. La red neuronal utilizada combina tareas de clasificación entre la capa de entrada (input layer) y la capa escondida (hidden layer) y agrega una tarea de regresión lineal entre la capa escondida y la capa de salida (x o y).

Los resultados mostraron que la precisión obtenida en el segundo diseño usando el algoritmo de regresión logística de aprendizaje supervisado Ensemble Bagged Trees mejora en relación a la precisión alcanzada con las redes neuronales, ya que el algoritmo añade una sola tarea en la capa oculta y una sola capa de salida (x o y).

Se probaron algoritmos de aprendizaje automático supervisado para determinar el método más preciso en la detección de posición. Los resultados obtenidos permiten establecer que la regresión logística del método Ensemble Bagged Trees genera el menor error cuadrático medio en la posición buscada.

RECOMENDACIONES

- Realizar cambios en las posiciones de los receptores o disminuir la cantidad de estos para obtener una base de datos diferente con la cual se puedan realizar nuevos entrenamientos que evalúen el desempeño del algoritmo frente a otros escenarios.
- Ensayar nuevos métodos de aprendizaje automático en una búsqueda de una mejor resolución en la determinación del posicionamiento de una sola persona.
- Incluir la información sobre los ángulos de llegada en las simulaciones de Ray Tracing para tener datos complementarios con los que se entrenen los algoritmos de aprendizaje automático.
- Utilizar la matriz de correlación que permita usar un menor número de receptores que reciben potencias muy parecidas para que el clasificador utilice la cantidad óptima de características (features) en la determinación de la posición.
- Incrementar la cantidad de ejemplos manteniendo la ubicación de la persona y modificando el entorno de ambiente de interiores para obtener una mayor cantidad de muestras de las posiciones de los individuos.
- Emplear técnicas de aprendizaje automático de reducción de características para simplificar el modelo de detección. Pueden ser incluidas las siguientes: análisis de componentes principales (PCA) o reducción secuencial de características.
- Realizar varios entrenamientos con redes neuronales y algoritmos de aprendizaje automático con datos de pruebas tomados de un ambiente de interiores en donde se pueda determinar la distancia entre dos cuerpos ubicados arbitrariamente.

APÉNDICES

APÉNDICE # 1

En el siguiente anexo se presentan los códigos desarrollados en Matlab para la función de la red neuronal de aprendizaje supervisado Bayesian Regularization.

```
%%%%%NEURONAL NETWORK BAYESIAN REGULARIZATION
```

```
function [y1] = my10NeuralNetworkFunction(x1)
```

```
%MYNEURALNETWORKFUNCTION neural network simulation function.
```

```
%
```

```
% Auto-generated by MATLAB, 27-Jan-2021 23:49:52.
```

```
%
```

```
% [y1] = myNeuralNetworkFunction(x1) takes these arguments:
```

```
% x = Qx19 matrix, input #1
```

```
% and returns:
```

```
% y = Qx2 matrix, output #1
```

```
% where Q is the number of samples.
```

```
% ===== SIMULATION =====
```

```
% Dimensions
```

```
Q = size(x1,1); % samples
```

```
% Input 1
```

```
x1 = x1';
```

```
xp1 = mapminmax_apply(x1,x1_step1);
```

```
% Layer 1
```

```
a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*xp1);

% Layer 2
a2 = repmat(b2,1,Q) + LW2_1*a1;

% Output 1
y1 = mapminmax_reverse(a2,y1_step1);
y1 = y1';
end

% ===== MODULE FUNCTIONS =====
% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
y = bsxfun(@minus,x,settings.xoffset);
y = bsxfun(@times,y,settings.gain);
y = bsxfun(@plus,y,settings.ymin);
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y,settings)
x = bsxfun(@minus,y,settings.ymin);
x = bsxfun(@rdivide,x,settings.gain);
x = bsxfun(@plus,x,settings.xoffset);
end
```

APÉNDICE # 2

En el siguiente apéndice se va a colocar el código fuente de la función del mejor algoritmo de predicción obtenida como resultado de los entrenamientos de los datos.

```
clear;clc;

Datos=load('tabla001mw_rand_R248h_28GHz.mat');Datos=Datos.TabPowPerAoDm
w_rand;

%248 sujetos x 22 (1st ID subject, 2nd -20th mw por Rx (19), position x and y)

Train=0.7; Test=1-Train;%<-----% Training Data and Testing Data

%Training data

Dout=Datos(1:cast(size(Datos,1)*Train,'uint8'),size(Datos,2)-
1:size(Datos,2));%Out(position x and y)

Din=Datos(1:cast(size(Datos,1)*Train,'uint8'),2:size(Datos,2)-2);%ln(Pow mw for
each Rx)

%Validation data

Tout=Datos(cast(size(Datos,1)*Train,'uint8'):size(Datos,1),size(Datos,2)-
1:size(Datos,2));%Out(position x and y)

Tin=Datos(cast(size(Datos,1)*Train,'uint8'):size(Datos,1),2:size(Datos,2)-2);%ln(Pow
mw for each Rx)

%Feature data

for i=1:19

    for j=i+1:19

        DFin=[DFin abs(Din(:,i)-Din(:,j))];

        TFin=[TFin abs(Tin(:,i)-Tin(:,j))];

    end

end

end
```

```
nnstart

%% Validation with training feature data

Salida1=[];

%Salida1=myFNeuralNetworkFunction(DFin);%<---- Funcion Original

%Salida1=myF100NeuralNetworkFunction(DFin);%<---- Funcion Original

Salida1=myF5NeuralNetworkFunction(DFin);%<---- Funcion Original

Xout=Dout;

Xin=DFin;

%% Validation with testing feature data

Salida1=[];

%Salida1=myFNeuralNetworkFunction(TFin);%<---- Funcion Original

%Salida1=myF100NeuralNetworkFunction(TFin);%<---- Funcion Original

Salida1=myF5NeuralNetworkFunction(TFin);%<---- Funcion Original

Xout=Tout;

Xin=TFin;

%% Validation with training data

Salida1=[];

Salida1=myNeuralNetworkFunction(Din);%<---- Funcion Original

Xout=Dout;

Xin=Din;

%% Validation with testing data

Salida1=[];

Salida1=myNeuralNetworkFunction(Tin);%<---- Funcion Original

Xout=Tout;

Xin=Tin;
```

APÉNDICE # 3

En el siguiente apartado se detalla el código del algoritmo de aprendizaje automático supervisado seleccionado

PROGRAMA PARA DETERMINAR LA UBICACIÓN DEL SUJETO BASADO EN APRENDIZAJE DE MÁQUINA SUPERVISADO: ENSEMBLE BAGGED TREES

Importing all raw data

```
AllData = cell2mat(importdata('Datos/files.txt'));
ValTrain_RMSEY=[];
ValTrain_RMSEX=[];
ValPred_RMSEY=[];
ValPred_RMSEX=[];
for i = 1:size(AllData,1)
%i=1
    data_name =AllData(i,:);
%end
fprintf('Initializing raw data\n')
raw_data_path = fullfile('./Datos', data_name);%file with row data
var_names = {'R1'; 'R2'; 'R3'; 'R4'; 'R5'; ...
    'R6'; 'R7'; 'R8'; 'R9'; 'R10'; 'R11'; 'R12'; ...
    'R13'; 'R14'; 'R15'; 'R16'; 'R17'; 'R18'; 'R19'};
data=load(raw_data_path);data=cell2mat(struct2cell(data));
[~, s] = sort(data(:, 1));
%data(s, :)
Var_names_All=[Var_names_All;strcat(num2str(i),var_names)];
dataAll=[dataAll data(s,2:20)];%data struct to mat file
end
```

```
Initializing raw data
Initializing raw data
Initializing raw data
Initializing raw data
Initializing raw data
Initializing raw data
Initializing raw data
Initializing raw data
Initializing raw data
Initializing raw data
Initializing raw data
```

```
dataAll=[dataAll data(s,21:22)];%data struct to mat file
Var_names_All=[Var_names_All;{'X'; 'Y'}];
```


Preprocessing raw data (Import data to table)

```

training_size = 0.7;%Training Data
Tdata = array2table(dataAll, 'VariableNames',Var_names_All);%Data to table with labels
training=Tdata(1:cast(size(Tdata,1)*training_size,'uint8'),2:size(Tdata,2));%without ID
evaluation=Tdata(cast(size(Tdata,1)*training_size,'uint8'):size(Tdata,1),2:size(Tdata,2));%without ID
fprintf('Training data: %d x %d\n',size(training))

```

Training data: 174 x 191

```

fprintf('Evaluation data: %d x %d\n',size(evaluation))

```

Evaluation data: 75 x 191

```

%Training data
Tout=training(:,size(training,2)-1:size(training,2));%Out(position x and y)
Tin=training(:,1:size(training,2)-2);%In(Pow mw for each Rx)
fprintf('Training data: %d x %d\n',size(Tin))

```

Training data: 174 x 114

```

%Evaluation data
Eout=evaluation(:,size(evaluation,2)-1:size(evaluation,2));%Out(position x and y)
Ein=evaluation(:,1:size(evaluation,2)-2);%In(Pow mw for each Rx)
fprintf('Evaluation data: %d x %d\n',size(Ein))

```

Evaluation data: 174 x 114

Linear regression

```
%[trainedModelY, validationRMSEY] = All20TrainRegressionModelY(training);%Modelo Y con un threshold al 20%-----
[trainedModelY, validationRMSEY] = AllTrainRegressionModelY(training);%Modelo Y con un threshold al 50%-----
```

```
validationRMSEY %RMSE with validation data, 10 folds
pEoutY=trainedModelY.predictFcn(Ein);
EoutY=table2array(Eout(:,2));%using function to predict
RMSEY = sqrt(mean((EoutY - pEoutY).^2)) %RMSE with evaluation data
```

```
%[trainedModelX, validationRMSEX] = All20TrainRegressionModelX(training);%Modelo X con un threshold al 20%-----
[trainedModelX, validationRMSEX] = AllTrainRegressionModelX(training);%Modelo X con un threshold al 50%-----
```

```
validationRMSEX %RMSE with validation data, 10 folds
pEoutX=trainedModelX.predictFcn(Ein);
EoutX=table2array(Eout(:,1));%using function to predict
RMSEX = sqrt(mean((EoutX - pEoutX).^2)) %RMSE with evaluation data
ValTrain_RMSEY=[ValTrain_RMSEY;validationRMSEY];
ValTrain_RMSEX=[ValTrain_RMSEX;validationRMSEX];
ValPred_RMSEY=[ValPred_RMSEY;RMSEY];
ValPred_RMSEX=[ValPred_RMSEX;RMSEX];
%end
```

BIBLIOGRAFÍA

- [1] Kaya, A. Ö., & Viswanathan, H. (2017, December). Coverage and capacity impact of mobility and human body blocking at millimeter waves. In GLOBECOM 2017-2017 IEEE Global Communications Conference (pp. 1-7). IEEE.
- [2] Kolakowski, J., Djaja-Josko, V., Kolakowski, M., & Broczek, K. (2020). UWB/BLE Tracking System for Elderly People Monitoring. *Sensors*, 20(6), 1574.
- [3] Jin, Z., Bu, Y., Liu, J., Wang, X., & An, N. (2015, August). Development of Indoor Localization System for Elderly Care Based on Device-Free Passive Method. In 2015 Sixth International Conference on Intelligent Systems Design and Engineering Applications (ISDEA) (pp. 328-331). IEEE.
- [4] Vance, P. J., Prasad, G., Harkin, J., & Curran, K. (2011). A wireless approach to device-free localisation (DFL) for indoor environments.
- [5] Tirumalareddy, R. R. (2020). BLE Beacon Based Indoor Positioning System in an-Office Building using Machine Learning
- [6] B. Mrazovac, M.Z. Bjelica, I. Papp and N. Teslic, "Smart Audio/Video Playback Control Based on Presence Detection and User Localization in Home Environment", ECBS-EERC '11, Conf. Proceeding, pp. 44-53, Bratislava, Slovakia, 2011
- [7] Singh, R., & Sicker, D. (2020). SHINE (Strategies for High-frequency INdoor Environments) with Efficient THz-AP Placement. arXiv preprint arXiv:2003.07015
- [8] Dalveren, Y., Alabish, A. H., & Kara, A. (2019). A simplified model for characterizing the effects of scattering objects and human body blocking indoor links at 28 GHz. *IEEE Access*, 7, 69687-69691.
- [9] A. Kara and H. L. Bertoni, "Effect of people moving near short-range indoor propagation links at 2.45 GHz," *J. Commun. Netw.*, vol. 8, no. 3, pp. 286-289, Sep. 2006.

- [10] A. Kara, "Human body shadowing variability in short-range indoor radio links at 3-11 GHz band," *Int. J. Electron.*, vol. 96, no. 2, pp. 205-211, Feb. 2009.
- [11] REMCOM, Wireless Insite. <https://www.remcom.com/wirelessinsite-empagation-software>
- [12] Zhan, J., Zhang, J., Tian, L., Zhang, X., Tang, P., Dou, J., & Wei, H. (2018). Comparative channel study of ray tracing and measurement for an indoor scenario at 28 GHz
- [13] REMCOM, Wireless Insite Reference Manual, 2013.
- [14] Zhan, J., Zhang, J., Tian, L., Zhang, X., Tang, P., Dou, J., & Wei, H. (2018). Comparative channel study of ray tracing and measurement for an indoor scenario at 28 GHz
- [15] Li, Y., Chen, X., Coates, M., & Yang, B. (2011, May). Sequential Monte Carlo radio-frequency tomographic tracking. In 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 3976-3979). IEEE.
- [16] A. Saeed, A.E. Kosba, M. Youssef Ichnaea: a low-overhead robust WLAN device-free passive localization system *IEEE J. Sel. Top. Signal Process.*, 8 (1) (2014), pp. 5-15,
- [17] C. Xu, B. Firner, Y. Zhang, R. Howard, J. Li, and X. Lin, "Improving RF-based device-free passive localization in cluttered indoor environments through probabilistic classification methods," in *Proc. Information Processing in Sensor Networks (IPSN-2012)*, April 2012, pp. 209–220.
- [18] Tsuchida, S., Takahashi, T., Ibi, S., & Sampei, S. (2019, November). Machine Learning-Aided Indoor Positioning Based on Unified Fingerprints of Wi-Fi and BLE. In 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) (pp. 1468-1472). IEEE.
- [19] Hojatian, H., Nadal, J., Frigon, J. F., & Leduc-Primeau, F. (2020). Unsupervised Deep Learning for Massive MIMO Hybrid Beamforming. arXiv preprint arXiv:2007.00038.

- [20] Chen, M., Liu, K., Ma, J., Zeng, X., Dong, Z., Tong, G., & Liu, C. (2020). MoLoc: Unsupervised fingerprint roaming for device-free indoor localization in a mobile ship environment. *IEEE Internet of Things Journal*, 7(12), 11851-11862.
- [21] REMCOM, Wireless Insite. <https://www.remcom.com/wirelessinsite-antennas>
- [22] Kumar, S., & Lee, S. R. (2014, June). Localization with RSSI values for wireless sensor networks: An artificial neural network approach. In *International Electronic Conference on Sensors and Applications* (Vol. 1). Multidisciplinary Digital Publishing Institute.
- [23] Kadavi, P., Lee, C.-W., & Lee, S. (2018). Application of Ensemble-Based Machine Learning Models to Landslide Susceptibility Mapping. *Remote Sensing*, 10(8), 1252. doi:10.3390/rs10081252
- [24] https://link.springer.com/chapter/10.1007/978-1-4842-5124-9_2
- [25] Chen, X., Edelstein, A., Li, Y., Coates, M., Rabbat, M., & Men, A. (2011, April). Sequential Monte Carlo for simultaneous passive device-free tracking and sensor localization using received signal strength measurements. In *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks* (pp. 342-353). IEEE.
- [26] Levy, M., Raviv, D., & Baker, J. (2019, January). Data center predictions using matlab machine learning toolbox. In *2019 IEEE 9th annual computing and communication workshop and conference (CCWC)* (pp. 0458-0464). IEEE.
- [27] Raju, M. P., & Laxmi, A. J. (2020). IoT based online load forecasting using machine learning algorithms. *Procedia Computer Science*, 171, 551-560.

- [28] Kho, Y. H., Chong, N. S., Ellis, G. A., & Kizilirmak, R. C. (2015, April). Exploiting RF signal attenuation for passive indoor location tracking of an object. In 2015 International Conference on Computer, Communications, and Control Technology (I4CT) (pp. 152-156). IEEE.

- [29] Lei, Q., Lv, H., Zhang, H., Sun, H., & Tang, L. (2016, November). Logistic regression-based device-free localization in changeable environments. In 2016 IEEE 13th International Conference on Signal Processing (ICSP) (pp. 1062-1066). IEEE.

- [30] Abdull Sukor, A. S., Kamarudin, L. M., Zakaria, A., Abdul Rahim, N., Sudin, S., & Nishizaki, H. (2020). RSSI-Based for Device-Free Localization Using Deep Learning Technique. *Smart Cities*, 3(2), 444-455.