

**UNIVERSIDAD AUTÓNOMA DE BAJA
CALIFORNIA**
INSTITUTO DE INGENIERÍA
MAESTRÍA Y DOCTORADO EN CIENCIAS E INGENIERÍA



**CONTEO DE PERSONAS USANDO
CHANNEL STATE INFORMATION
GENERADO POR WIFI Y ALGORITMOS DE
MACHINE LEARNING Y DEEP LEARNING**

**TESIS PARA OBTENER EL GRADO DE:
MAESTRO EN CIENCIAS**

Presenta:

MIGUEL ANGEL TORRES CERDA

Director de Tesis:

DR. FÉLIX FERNANDO GONZÁLEZ NAVARRO

Codirector de Tesis:

DR. JESÚS CARO GUTIÉRREZ

MEXICALI, B. C.

DICIEMBRE 2024

Dedicatorias

Esta tesis es dedicada a mis padres Daniel y Guadalupe, quienes han sido mi pilar fundamental a lo largo de toda mi vida. Su apoyo incondicional, sus consejos y su amor me han dado la fuerza y la motivación para lograr mis objetivos. A Mayra Cristel, mi compañera de vida y sueños, le agradezco profundamente su paciencia, comprensión y cariño. Su presencia ha sido mi refugio y apoyo emocional a lo largo de mis estudios.

Agradecimientos

Me gustaría darle las gracias a los profesores que me ayudaron y guiaron durante el tiempo que recorrí mi estudio de posgrado, en especial a mi director de Tesis Félix Fernando Gonzalez Navarro por brindarme la oportunidad para realizar mis estudios de maestría.

Agradecer al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) y la Universidad Autónoma de Baja California (UABC) por el apoyo y el espacio que me brindaron para realizar mis estudios.

También agradecer a mis amigos quienes voluntariamente me apoyaron para realizar las tareas de recolección de datos.

Abstract

El conteo de personas es una técnica que nos ayuda en situaciones como mantener la salud dentro de edificios o regular sistemas A/C. En esta Tesis se presentan dos experimentos de conteo de personas que utilizan el acercamiento de Channel State Information generado a partir de una red WiFi para generar un conjunto de datos, el cual es explorado mediante algoritmos de Machine Learning y Deep Learning, pasando previamente por etapas de procesamiento de datos. De los dos sistemas presentados, el segundo es el que presenta un mayor rendimiento. En el análisis de los algoritmos de Deep Learning se presenta la posibilidad de crear una aplicación de tiempo real que se encargue del conteo de personas utilizando dispositivos ESP32 y el algoritmo de clasificación 1DCNN. Los resultados en la tasa de reconocimiento del sistema nos presentan un sistema confiable, que puede medir al 100% de precisión cuando una habitación se encuentra vacía, hay 1 persona dentro, o hay más de 2 personas.

Contenido

1	Introducción	1
1.1	Planteamiento del problema	2
1.2	Objetivos y metas	4
1.2.1	Objetivo General	4
1.2.2	Objetivos específicos	4
1.2.3	Metas	4
2	Estado del arte	5
2.1	Machine learning	5
2.2	Deep Learning	8
3	Marco Teórico	12
3.1	Multiplexación por División de Frecuencias Ortogonales	12
3.2	Channel State Information	12
3.3	Imputación de datos	13
3.4	Filtros	14
3.4.1	Filtro de mediana	14
3.4.2	Filtro de Hampel	15
3.5	Selección de subportadoras	15
3.5.1	minimum Redundancy Maximum Relevance (mRMR)	15
3.5.2	Selección por Forward y Backward Selection	16
3.6	Visualización de datos	19
3.6.1	T-distributed Stochastic Neighbor Embedding	19
3.7	Algoritmos de Machine Learning	19
3.7.1	Clasificador Naive Bayes	19
3.7.2	K Nearest Neighbor (KNN)	19
3.7.3	Clasificadores Discriminantes	19
3.7.4	Support Vector Machines (SVM)	20
3.8	Algoritmos de Deep Learning	20
3.8.1	One Dimensional Convolutional Neural Network - (1DCNN)	21
3.9	Jaula de Faraday	21

4	Materiales y métodos	22
4.1	Primer experimento	22
4.1.1	Metodología del sistema	22
4.1.2	Recursos	23
4.1.3	Construcción del primer experimento - configuración A (E1A)	26
4.1.4	Construcción del primer experimento - configuración 2 (E1B)	31
4.2	Jaula de Faraday	35
4.2.1	Cálculos de la malla	36
4.2.2	Construcción de la Jaula	36
4.2.3	Reforzamiento de la jaula	37
4.3	Segundo experimento	39
4.3.1	Metodología del experimento	39
4.3.2	Recursos del experimento	40
4.3.3	Configuración de la Red Neuronal Convolutiva de 1 Dimensión	44
4.3.4	Construcción del segundo experimento - configuración C (E2C)	44
4.3.5	Construcción del segundo experimento - configuración D (E2D)	45
5	Resultados	47
5.1	Primer experimento	47
5.1.1	Configuración A	47
5.1.2	Configuración B	49
5.2	Segundo experimento	56
5.2.1	Configuración C	56
5.2.2	Configuración D	58
6	Conclusiones y trabajos futuros	59
6.1	Conclusiones	59
6.2	Trabajos futuros	60
7	Apéndice A	62
8	Apéndice B	67
	Referencias	79

Figuras

3.1	Representación de CSI	13
3.2	Ejemplo de uso del filtro de mediana	15
3.3	Ejemplo de uso del filtro de Hampel	16
4.1	Primer experimento	23
4.2	Diagrama de ubicación física de las antenas	24
4.3	Imagen de la habitación	24
4.4	Intel NUC transmisor con una antena	25
4.5	Intel NUC receptor con tres antenas	25
4.6	Subportadoras principales para el primer conjunto de datos	27
4.7	Primera versión del conjunto de datos de E1A	28
4.8	Señal para 0 personas antes y después de aplicar los filtros de Hampel y mediana en el primer conjunto de datos	29
4.9	Segunda versión conjunto de datos de E1A	30
4.10	Tercera versión conjunto de datos de E1A	30
4.11	Primera versión del conjunto de datos de E1B	31
4.12	Segunda versión conjunto de datos de E1B	32
4.13	Tercera versión del conjunto de datos de E1B	33
4.14	Cuarta versión del conjunto de datos de E1B	33
4.15	Versión final conjunto de datos de E1B	34
4.16	Conjunto de datos para 2 clases de E1B	35
4.17	Construcción de la estructura de la jaula	37
4.18	Estructura de la jaula	37
4.19	Ejemplo de un parche en la malla	38
4.20	Ejemplo de un segundo parche en la jaula	38
4.21	Medición de continuidad en la jaula	38
4.22	Diagrama a bloques del segundo experimento	40
4.23	ESP32 que actúa como estación	41
4.24	ESP32 que actúa como punto de acceso	41
4.25	Primera versión del conjunto de datos de E2C	42
4.26	Segunda versión del conjunto de datos de E2C	42
4.27	Tercera versión del conjunto de datos de E2C	43
4.28	Cuarta versión del conjunto de datos de E2C	43

4.29	Arquitectura de la Red Neuronal Convolutiva	44
4.30	Señales principales para 0 personas antes y después de aplicar los filtros de Hampel y mediana en el segundo conjunto de datos	45
4.31	Versión final del conjunto de datos de E2D	46
5.1	Comportamiento del sistema 1 con la técnica T-sne	48
5.2	Comportamiento del conjunto de datos corregido	49
5.3	Comportamiento del conjunto de datos con sólo 2 clases	50

Tablas

2.1	Resumen de los trabajos que usan Machine Learning	8
2.2	Resumen de trabajos que usan Deep Learning	11
4.1	Relación con el total de personas y las clases para las actividades . .	25
5.1	Métricas de algoritmos de Machine Learning con el conjunto de 2 clases	50
5.2	Resultados de SVM usando el conjunto de 2 clases	51
5.3	Resultados de SVM usando el conjunto de 9 clases y la estrategia <i>one vs. all</i>	51
5.4	Resultado de SVM usando el conjunto de 9 clases y la estrategia <i>one vs. one</i>	52
5.5	Total de iteraciones por clase para aplicar SVM con estrategia one vs. one	54
5.6	Matriz de confusión para 1DCNN usando 2 clases	55
5.7	Matriz de confusión para la 1DCNN contando 9 clases	55
5.8	Matriz de confusión para la 1DCNN contando 7 clases	56
5.9	Matriz de confusión para la 1DCNN con el modelo optimizado . . .	58
6.1	Variables de entorno para la configuración de los sistemas	61

1. Introducción

Debido a la situación de salud global ocurrida en el año 2020 fuimos obligados por el virus COVID-19 a fortalecer nuestras medidas de seguridad sanitarias. Se implementó un distanciamiento social entre la población como medida para evitar esparcir el virus de manera directa [1, 2]. Una manera de asegurar que se cumpla el distanciamiento social es limitar la capacidad de personas que pueden acceder al mismo tiempo dentro de una habitación cerrada. Como herramienta para saber cuantas personas hay dentro del espacio cerrado se utiliza el conteo de personas, con el cual podemos saber si se cumple el distanciamiento, conociendo el número de personas dentro del espacio cerrado y de esta manera ayudar a preservar una buena salud en la población al minimizar la posibilidad de contagios.

El conteo de personas es una técnica que se implementa hoy en día mediante aplicaciones que utilizan cámaras digitales para tomar fotografías y contar a las personas dentro de la imagen, o con aplicaciones que dependen en que las personas lleven consigo algún dispositivo radioreceptor para contar las señales recibidas por los dispositivos. Una mejora para estas técnicas de conteo son aplicaciones que utilizan WiFi para generar señales a partir de la Información del Estado de Canal (Channel State Information o CSI), y realizar el conteo de personas mediante algoritmos de clasificación en base a estas señales generadas.

En esta Tesis se presenta un trabajo de conteo de personas usando Channel State Information generado por una red WiFi, apoyado de algoritmos de Machine Learning y Deep Learning que permiten realizar la tarea de clasificación para las señales generadas a partir del CSI que se obtiene de la red WiFi y de esta manera conocer el total de personas que se encuentra dentro de una habitación al momento de generar cada señal de CSI.

El documento se encuentra dividido en un total de 6 capítulos. En el capítulo 1 se detallan los problemas existentes con los métodos actuales de conteo de personas, así como los beneficios presentes al realizar el conteo con redes WiFi sobre los métodos de conteo tradicionales.

Todo el material encontrado en la literatura y relacionado a este trabajo donde

utilizan el Channel State Information para generar conjuntos de datos y realizar un conteo de personas se encuentran en el capítulo 2. Entre estos trabajos relacionados se encuentran autores que implementan algoritmos tanto de Machine Learning como Deep Learning para la etapa de clasificación.

En el capítulo 3 se explica las técnicas y conceptos empleados en este sistema de conteo de personas que utiliza CSI, entre ellos se encuentran las definiciones de Multiplexación por División de Frecuencias Ortogonales (OFDM) y Channel State Information (CSI) los cuales son conceptos clave para generar señales de las cuales es posible realizar el conteo de personas. Además, se muestra algoritmos de limpieza de datos, reducción de dimensionalidad, visualización de datos y clasificación que se utilizaron en los experimentos de este trabajo.

Los experimentos realizados, así como los materiales y metodología empleada para llevar a cabo cada uno de ellos se muestra en el capítulo 4. En total se realizaron dos experimentos de conteo de personas utilizando distintos materiales y entornos entre uno y otro. En medio de estos dos experimentos, se llevó a cabo la construcción de una Jaula de Faraday para experimentar con un ambiente controlado.

Después de la experimentación se presentan los resultados en el capítulo 5. En este se presenta los resultados de experimentación resaltados en porcentaje de tasa de reconocimiento y matrices de confusión, así como las áreas de oportunidad generadas en cada experimento. Cada experimento se encuentra dividido en dos configuraciones, esto debido a que en la primera configuración de cada experimento se presentan áreas de oportunidad que permiten incrementar la eficiencia, lo cual se ve reflejado con la segunda configuración.

Por último, se encuentran las conclusiones y trabajos futuros en el capítulo 6. De acuerdo a los experimentos realizados se puede concluir que sí es posible realizar una aplicación de tiempo real que se encargue de contar personas a través de redes WiFi. Sin embargo, para llevar a cabo una optimización de dicha aplicación es necesario realizar como trabajo futuro algunas pruebas de rendimiento y configurar de distintas maneras los diversos algoritmos de clasificación.

1.1 Planteamiento del problema

Las técnicas de conteo de personas tuvieron un incremento en relevancia debido a la pandemia ocurrida por el virus COVID-19. A día de hoy, la comunidad se encuentra más consciente de la importancia de mantener una distancia segura entre personas para cuidar su salud y la de los demás al evitar propagaciones de distintos tipos de virus. Debido a esto es muy importante mantener bajo control el total de personas que se encuentran dentro de espacios cerrados, para evitar contagiar con algún virus a otras personas [1, 2].

1.1 Planteamiento del problema

Existen distintos métodos de conteo de persona, entre ellos podemos destacar aquellos que necesitan cámaras para obtener imágenes, o los que necesitan que las personas que se encuentran dentro del conteo lleven consigo algún dispositivo radioreceptor para ser tomados dentro del conteo. Sin embargo, estos métodos cuentan con algunos problemas.

Los métodos que se basan en cámaras pueden presentar puntos ciegos dependiendo del punto de vista que tenga la cámara desde su punto de instalación; las cámaras son susceptibles a la contaminación ambiental como rayos de luz; es posible que al momento de obtener una imagen se encuentren objetos bloqueando la vista de los sujetos que deben ser contados; las cámaras toman imagen de las personas, lo cual invade de la privacidad de las mismas al ser identificadas al capturar la imagen.

Los métodos que no se encuentran basados en cámaras necesitan que las personas dentro del conteo carguen consigo un dispositivo radioreceptor, como teléfonos móviles o etiquetas RFID. Con este punto es imposible confiar en que todas las personas lleven siempre consigo un dispositivo para aplicar técnicas de conteo en situaciones reales.

Una opción para los métodos de conteo es utilizar el Channel State Information (CSI) que generan las conexiones WiFi. De manera resumida, una conexión WiFi estima el CSI para poder conocer el entorno y obtener una ruta óptima para la comunicación. El conteo de personas basado en CSI es relativamente nuevo, comparado con los métodos tradicionales, por lo que el alcance de este es aún desconocido.

El conteo de personas con CSI presenta ventajas sobre los actuales métodos de conteo. Es importante remarcar que el CSI, al generarse a partir de una comunicación WiFi, no requiere de un lente para realizar el conteo, por esta razón, no presenta las desventajas que poseen las cámaras como contaminación visual producida por rayos de luz, neblina, polvo u objetos que causen interferencia; tampoco necesitan que los sujetos que se encuentran dentro del conteo lleven consigo cualquier dispositivo para ser incluidos. El CSI que se obtiene mediante el WiFi es un conjunto de información sobre la condición del canal de comunicación, por esta razón, este tipo de conteo no recolecta información como imágenes de las personas que están dentro de la señal, lo que permite mantener la privacidad de los individuos.

El conteo de personas tiene un amplio campo de aplicaciones. El primero de ellos es actuar como herramienta en un sistema de control de multitudes, donde al minimizar el total de personas en espacios cerrados se pueden reducir los riesgos en la salud; algunos negocios como hoteles pueden ser beneficiados en temas de seguridad al saber si los huéspedes respetan el número máximo de personas que les provee la instalación; en tiendas de supermercado es importante conocer las

áreas que los clientes suelen visitar con mayor frecuencia para realizar mejoras continuas; en edificios inteligentes es posible regular la energía utilizada en algunos dispositivos como el aire acondicionado al regular la temperatura dependiendo del total de personas en la habitación. Otra área de aplicación son las ciudades inteligentes donde, de manera similar a como se plantea para los supermercados, tener conocimiento sobre las áreas visitadas por los ciudadanos ayuda a la toma de decisiones para invertir los recursos de la ciudad en desarrollar planes de desarrollo que permitan mejorar la calidad de vida de las personas.

1.2 Objetivos y metas

1.2.1 Objetivo General

Evaluar experimentalmente algoritmos de Machine Learning y Deep Learning, como Support Vector Machines y 1 Dimension Convolutional Neural Network, respectivamente, para tareas de clasificación en conteo de personas mediante el Channel State Information (CSI) de señales WiFi.

1.2.2 Objetivos específicos

- Identificar dentro de los algoritmos de clasificación Naive Bayes, K Nearest Neighbor (KNN), Linear Discriminant Classifier (LDC), Quadratic Discriminant Classifier (QDC), Support Vector Machines (SVM), 1 Dimension Convolutional Neural Network (1DCNN) aquellos con mayor tasa de reconocimiento para realizar una aplicación de conteo de personas en ambientes controlados.
- Caracterizar las ventajas y desventajas de utilizar dispositivos ESP32 y la Intel NUC Kit D54250WYKH Wifi Link 5300 en el conteo de personas.
- Desarrollar un conjunto de herramientas de software para el conteo de personas en lenguaje MATLAB y Python.

1.2.3 Metas

- Un conjunto de herramientas de software para el conteo de personas.
- Tesis de Maestría.
- Publicación de artículo en congreso.

2. Estado del arte

La técnica de conteo de personas mediante WiFi puede ser explorada mediante las dos ramas principales de la Inteligencia Artificial, con Machine Learning y Deep Learning. Algunos métodos de Machine Learning aplicados para clasificar el conteo son Support Vector Machine (SVM), K Nearest Neighbor (KNN), Random Decision Forest (RDF), etc. Por el lado de los algoritmos de Deep Learning hay autores que han explorado las Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), Long Short Term Memory Networks (LSTM), etc. En este capítulo se presenta la información relacionada con los trabajos actuales que exploran el conteo de persona con CSI mediante algoritmos de Machine Learning y Deep Learning.

2.1 Machine learning

En esta sección se recapitulan aquellos trabajos relacionados con el conteo de personas que utilizan técnicas de Machine Learning para lograr el objetivo.

Di Domenico [3] presenta un sistema basado en CSI, utilizando una conexión entre transmisor y receptor con el protocolo ICMP, y el método de índices Davies Bouldin para contar a las personas que se encuentran dentro de una habitación pequeña y otra grande. Este método presenta una exactitud de 74% estimando dentro de la habitación pequeña las etiquetas de habitación vacía, 1 persona, 2 personas, 3 o 4 personas, y 5 a 7 personas. En la habitación grande la tasa se reduce hasta un 52%.

Yang [4] presenta un sistema usando 1 transmisor y 2 receptores en una puerta para detectar la dirección y contar a las personas que pasan a través de la misma. Este sistema utiliza un Análisis de Componentes Independientes para encontrar el número de personas que pasan a través con una precisión del 95% detectando la dirección y un 92% contando a las personas.

Para cubrir la necesidad de contar a personas en espacio cerrados Saandeep Depatla [5] propone una metodología de conteo donde es posible contar hasta un máximo de 20 personas con las antenas detrás de paredes. En este detallan que al obstruir la Línea de Visión (Line Of Sight) de la comunicación se generan cambios pronunciados en la amplitud (Signal dips) y estos cambios pronunciados son los que poseen información más robusta que les permite realizar el conteo de personas a través de las paredes.

El ahorro en el consumo de energía que implica conocer el número de habitantes para las construcciones inteligentes fue la razón principal que motivó a Han Zou [6] para diseñar su sistema WiFree, el cual permite detectar un total de 11 personas a través de dispositivo IoT. El sistema utiliza dos routers de 5GHz para obtener el CSI y usan el algoritmo SVM para clasificar y contar a los 11 individuos con una exactitud del 92.8%.

Mohan Liyanage [7] menciona lo importante que se ha vuelto el conteo de personas en los años recientes y su aplicación en situaciones como festivales, control peatonal, control de masas en lugares como subterráneos y tiendas. El sistema presentado permite exactitud del 99% contando hasta 16 personas con tres transmisores y tres receptores. Utiliza varios algoritmos de clasificación, como son Complex Tree, SVM, y KNN, presentando los mejores resultados para el sistema con el KNN.

Jens Weppner [8] menciona la importancia del conteo de personas en varias aplicaciones como evacuaciones de emergencia coordinadas, u optimizar el uso de energía y espacio en oficinas. Esto lo llevó a diseñar un sistema que sea capaz de medir el nivel de ocupación de una habitación larga con un total de hasta 8 personas con el algoritmo Random Decision Forest. En el trabajo se comparan los resultados de este algoritmo con SVM, obteniendo resultados similares, con la diferencia que RDF resulta ser 14 veces más rápido que SVM.

El bajo costo, la preservación de la privacidad de los individuos y la capacidad actual del WiFi de estar presente en casi cualquier sitio motivó a Omotayo Oshiga [9] a realizar un sistema de conteo basado en CSI, donde fue posible contar hasta 10 personas. Omotayo menciona la efectividad de los métodos usados de preprocesamiento (filtro Chebyshev) y clasificación (SVM, Random Forest y GB), con los que puede obtener más del 95% de exactitud.

La administración energética fue la causa para que Chen [10] propusiera una técnica de WiFi basada en RSSI (Indicador de la Fuerza de la Señal Recibida) para estimar el total de personas utilizando la Función de Densidad Probabilística, contando desde 0 hasta 4 personas con tasas de reconocimiento desde 100% para 0 personas, hasta 94% para 4 personas.

Ramon F. Brena [11] previendo la necesidad de permanecer seguro en espacios cerrados debido a la situación global provocada por el virus COVID-19, realizó un

sistema capaz de contar hasta 8 personas simultáneamente utilizando dos transmisores y tres receptores. En el trabajo usan distintos algoritmos de clasificación, como Random Forest, Weighed KNN, Linear Discriminant y SVM. Dentro de sus experimentos encontraron que, para ese trabajo, el mejor patrón de reconocimiento se obtuvo utilizando un mayor número de pares de antenas entre transmisor y receptor.

La capacidad del CSI que nos permite realizar aplicaciones de conteo de personas, reconocimiento de actividades humanas y localización de las mismas fue la razón para que Hyuckjin Choi [12] diseñara el sistema Wi-Cal. Dicho sistema permite contar y obtener la localización de hasta 10 personas dentro de una habitación. Esto se realiza con la ayuda de los filtros Hampel y Savitzky-Golay para limpiar las señales generadas a partir del CSI; como algoritmos de clasificación se utilizan Linear Regresor, Random Forest Regresor, Random Forest Classifier y Logistic Classifier.

En [13] se habla sobre el posible ahorro energético y protección del ambiente a causa de ajustar sistemas HVAC (aire acondicionado) de acuerdo a la cantidad de personas que se encuentran dentro de una habitación. Para lograr esto, crearon un sistema basado en CSI capaz de contar hasta 4 personas. Evalúan distintos algoritmos de Machine Learning como SVM, RF, KNN o DT, obteniendo una mayor precisión con las SVM de 87.2%. Para el sistema utilizan router COTS para obtener el CSI y crear señales a partir de este, seguido de etapas comunes en Data Mining como son Selección de información, remoción de ruido, extracción de características, y aplicación de los métodos de Machine Learning para clasificar.

El conteo de personas dentro de una habitación (Indoor Crowd Counting, o ICC) posee una amplia gama de aplicaciones, como el manejo de salud, monitoreo de seguridad o ahorro de energía. Debido a esto, en [14] crearon su sistema para contar hasta un total de 8 personas con distintos algoritmos como LR, SVM y KNN, obteniendo una tasa de reconocimiento del 99% con LR. Para lograr esto, utilizan tanto la información de amplitud como la información de fase que proporciona el CSI. Utilizan CNN para extraer características y posteriormente alimentar los algoritmos de Machine Learning. De acuerdo a su trabajo, utilizando escenarios dinámicos (gente en movimiento) se pueden obtener mejores resultados que con escenarios estáticos.

Debido a la creciente necesidad de monitoreo de multitudes en espacios públicos para propósitos de salud que provocó la pandemia COVID-19, en [15] crean un sistema para contar desde 0 hasta 17 personas con un 84% de reconocimiento. En el trabajo mencionan que es posible usar señales LTE para cubrir áreas donde el WiFi no esté presente. A partir de las señales de CSI extraídas, sólo aplican métodos de suavizado, y las técnicas que utilizan para la predicción son CSI secant y Singular Value Decomposition en el CSI.

En la tabla 2.1 se visualiza una recopilación acerca de los trabajos relacionados que utilizan Machine Learning. En ella se presenta el autor, hasta cuantas personas se logró contar como objetivo, los algoritmos utilizados y el porcentaje de exactitud más alto que se obtuvo con el primer algoritmo que se muestra en la columna de algoritmo.

2.2 Deep Learning

Sanaz Kianoush [16] menciona las ventajas que presenta el conteo de personas a partir de WiFi sobre otros métodos tradicionales como cámaras, razón por la cual creó un sistema que permite contar hasta 5 personas por medio del CSI generado a partir de la señal WiFi. En dicho sistema utilizan una configuración de 1 transmisor y 9 receptores, donde el método que arrojó mejores resultados es Recurrent Neural Network.

Ibrahim [17] presenta un sistema usando Recurrent Neural Networks para estimar el total de personas entre 0 a 10 a través de una pared de ladrillos. Esta

Autor	Objetivo	Algoritmo	Exactitud
Di Domenico [3]	Contar 7 personas	DBI	74%
Yang [4]	Contar personas caminando a través de una puerta	ICA	92%
Saandeep Depatla [5]	Contar 20 personas	- - -	- - -
Han Zou [6]	Contar 11 personas	SVM	92.8%
Mohan Liyanage [7]	Contar 16 personas	KNN	99%
Jens Weppner [8]	Contar 8 personas	RFD	- - -
Omotayo Oshiga [9]	Contar 10 personas	SVM	95%
Chen [10]	Contar 4 personas	PDF	94%
Ramon F. Brena [11]	Contar 8 personas	RF, KNN, LD, SVM	- - - - - -
Hyuckjin Choi [12]	Contar 10 personas	LR, RFR, RFC, LC	- - - - - -
Liu, Zhixin [13]	Contar 4 personas	SVM, RF, KNN, DT	87.2%
Hou, Huawei [14]	Contar 8 personas	LR, SVM, KNN	99%
De Sanctis, Mauro [15]	Contar 17 personas	CSI Secant, SVD	84%

Tabla 2.1: Resumen de los trabajos que usan Machine Learning

técnica presenta resultados del 63% contando de 0 a 7 personas, mientras que para 0 a 10 se obtiene un 55%.

La posibilidad de desarrollar tecnología de vigilancia no intrusiva fue lo que llevó a Wei Shi [18] a diseñar un sistema que permita contar hasta 7 personas dentro de un elevador. Para ello, se utilizaron los algoritmos de preprocesamiento de datos Savitzky Golay, Mean Centering, Hampel, y Down Sampling para limpiar las señales, después se aplicaron transformaciones wavelet para utilizar como entrada a una Convolutional Neural Network para llevar a cabo el conteo.

La efectividad para la administración y seguridad en eventos públicos de larga escala llevó a Singh [19] a proponer un sistema utilizando modelos Long Short Term Memory para predecir la densidad de una población de hasta 5,000 personas. Singh usa cinco distintos modelos LSTM, obteniendo una tasa de reconocimiento de 59.17% prediciendo la densidad de 2 ciudades distintas con el modelo ConvLSTM.

Rui Zhou [20] menciona tres puntos relevantes acerca del conteo de personas, el primero es la importancia de esto en distintas aplicaciones, después menciona que los métodos tradicionales invaden la privacidad de las personas al tomar imágenes de los sujetos, y por último, el alto costo que tienen los métodos actuales. Todo esto lo llevó a diseñar un sistema que permite contar hasta un máximo de 34 personas utilizando tres pares de antenas. En su trabajo es posible observar que al aumentar el número de antenas, los errores comienzan a bajar, y que a un mayor número de personas la precisión del conteo empeora.

El bajo costo y la preservación de la privacidad que posee el conteo de personas mediante WiFi así como las aplicaciones que tiene en sistemas de optimización de energía en construcciones inteligentes y monitorear las filas de espera para informar al personal de restaurantes y tiendas es la razón por la que Chong Tang [21] realizó su sistema PWR, el cual puede detectar hasta 4 personas al mismo tiempo en una habitación. Para ello utiliza Deep Neural Network para clasificar 4 personas dentro de la habitación.

Wei Zhuang [22] vio la necesidad que provocó la pandemia del año 2020 en el tema de conteo de personas en áreas públicas. Debido a esto presenta un sistema robusto capaz de contar hasta 10 personas dentro de una habitación utilizando Convolutional Neural Networks.

Debido al virus COVID-19, la importancia de mantener nuestras distancias se incrementó drásticamente, por esta razón Yichuan Zhang [23] presenta un sistema basado en CSI que permite monitorear qué sillas se encuentran ocupadas. Para ello usa cinco sillas y dos participantes, donde las personas cambian de ubicación entre qué sillas se encuentran ocupadas para realizar las distintas combinaciones. En este utiliza los algoritmos de clasificación Convolutional Neural Network, Naive Bayes, Support Vector Machines y KNN, obteniendo el mejor resultado con CNN con un 90% de exactitud.

El potencial que tienen las aplicaciones de WiFi sensing en distintos escenarios como hospitales o percibir señales vitales debido a la ubicuidad del WiFi y el bajo costo de los sistemas fue la motivación en [24] para crear TWCC, un sistema que permite contar hasta 5 personas con un 90% de exactitud, y un 98.54% en detectar presencia de personas. Para ello, utilizan una CNN alimentada con espectrogramas creados a partir de la transformada Hilbert.

La importancia de la detección inalámbrica de la respiración en distintas aplicaciones, y la escasa información en el estado del arte acerca del reconocimiento de humanos a partir de la frecuencia respiratoria llevó a [25] a crear un sistema capaz de rastrear la respiración de múltiples personas por medio del CSI, permitiendo saber la cantidad de personas y cuál respiración corresponde a cada una de las personas. El sistema presenta una tasa de reconocimiento del 86% contando hasta 4 personas. El sistema presenta tres fases, la primera de recolección de CSI, seguido del seguimiento de la respiración a través de STFT, por último el conteo de personas, donde usan la información de la fase 2 en el dominio del tiempo y un novedoso filtro cuasi bilateral que les permite realizar dicho conteo.

En la tabla 2.2 se muestra un resumen de aquellos trabajos relacionados que utilizan Deep Learning como algoritmo de clasificación principal. En la tabla se muestra el autor del trabajo, el total de personas que se logró contar, el algoritmo utilizado y el mejor porcentaje de exactitud con dicho algoritmo.

De los trabajos relacionados de este capítulo se pueden destacar algunos algoritmos de clasificación. Por parte de Machine Learning, se encontró que en los trabajos donde se utiliza Support Vector Machines (SVM) se presentan altas tasas de reconocimiento. Por el lado de Deep Learning la mayoría de trabajos encontrados utilizan Convolutional Neural Network. En el siguiente capítulo se describen los algoritmos de preprocesamiento y clasificación seleccionados para este experimento.

Autor	Objetivo	Algoritmo	Exactitud
Sanaz Kianoush [16]	Contar 5 personas	RNN	- - -
Ibrahim [17]	Contar 10 personas	RNN	55%
Ibrahim [17]	Contar 7 personas	RNN	63%
Wei Shi [18]	Contar 7 personas	CNN	- - -
Singh [19]	Predecir la densidad con 5,000 personas	LSTM	59.17%
Rui Zhou [20]	Contar hasta 34 personas	- - -	- - -
Chong Tang [21]	Contar 4 personas	DNN	- - -
Wei Zhuang [22]	Contar 10 personas	CNN	- - -
Yichuan Zhang [23]	Contar 2 personas en 5 sillas distintas	CNN	90%
Guo, Zhengxin [24]	Contar 5 personas	CNN	90%
Wang, Fengyu [25]	Contar 4 personas	STFT	86%

Tabla 2.2: Resumen de trabajos que usan Deep Learning

3. Marco Teórico

En este capítulo se detallan los conceptos clave, técnicas de procesamiento de señales, métodos de visualización de datos y algoritmos de clasificación de Machine Learning y Deep Learning utilizados en este trabajo de tesis.

3.1 Multiplexación por División de Frecuencias Ortogonales

Las redes WiFi que implementan el estándar IEEE 802.11n permiten utilizar la técnica de Multiplexación por División de Frecuencia Ortogonal (OFDM por sus siglas en inglés). Esta técnica ayuda a combatir el desvanecimiento de datos por trayectos múltiples, transmitiendo los datos en paralelo a través de la modulación de subportadoras ortogonales [26].

3.2 Channel State Information

Channel State Information (CSI) es una caracterización sobre cómo se propaga una señal WiFi desde transmisor hasta receptor. La amplitud y fase que presenta el CSI puede presentar variaciones o atenuaciones durante el proceso de transmisión, causadas por distintos efectos como reflexión o dispersión de la señal inalámbrica a través de las distintas rutas de propagación disponibles. Esas variaciones pueden ser medidas y analizadas para distintas actividades de reconocimiento humano, entre ellas, el conteo de personas.

La comunicación WiFi basada en el estándar IEEE 802.11n se divide en múltiples subportadoras debido a la técnica OFDM. Una subportadora de CSI se puede modelar como $y = Hx + n$, donde y es la señal recibida, x es la señal transmitida. n es el ruido que se puede presentar y H es la matriz de CSI, que está compuesta

Algoritmo 1: Algoritmo de imputación de datos

```

1  $[renglones, columnas] \leftarrow find(isnan(datos));$ 
2  $datos_2 \leftarrow datos;$ 
3  $datos[renglones, :] \leftarrow [];$ 
4  $k \leftarrow 13;$ 
5  $res \leftarrow zeros(size(renglones, 1));$ 
6  $i \leftarrow 1;$ 
7 while  $i < size(columnas)$  do
8    $datos_3 \leftarrow datos_2;$ 
9    $datos_3(:, columnas(i)) \leftarrow [];$ 
10   $datos_i \leftarrow datos(renglones(i), :);$ 
11   $datos_i(:, columnas(i)) \leftarrow [];$ 
12   $distancias \leftarrow sqrt(sum(datos_i - datos_3)^2);$ 
13   $[\sim, idx] \leftarrow sort(distancias);$ 
14   $value \leftarrow mean(datos_2(idx(1:k), col(i)));$ 
15   $res(i) \leftarrow value;$ 

```

3.4 Filtros

Los filtros son utilizados para limpiar las señales y trabajar de una manera regularizada. En este trabajo se usan dos filtros principales. Un filtro de mediana que permite generar un suavizado en la señal, y un filtro de Hampel con el cual se eliminan valores atípicos que se presentan en la misma.

3.4.1 Filtro de mediana

El filtro de mediana [29] realiza un suavizado en la señal, ingresando una ventana de tiempo de n datos a la izquierda y derecha del valor al que se le aplica el filtro. Con esta ventana se calcula la mediana de los valores y el resultado se usa para reemplazar el valor original.

Un ejemplo del funcionamiento de este filtro se puede ver en la figura 3.2, en estas gráficas se muestra una señal con ruido, y la misma señal después de haber aplicado un filtro de mediana. Como resultado de este suavizado, los cambios que presenta la nueva señal pasan a ser menos pronunciados en comparación de la señal original.

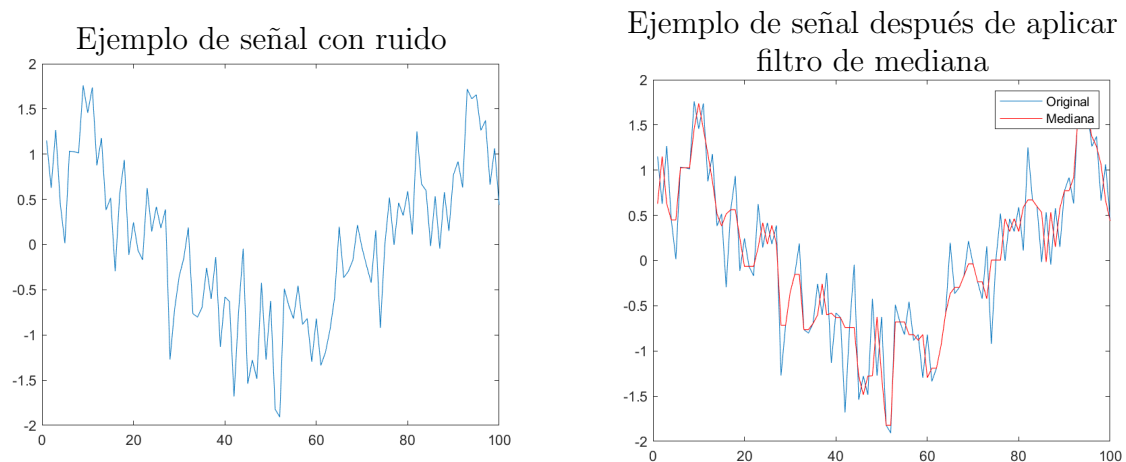


Figura 3.2: Ejemplo de uso del filtro de mediana

3.4.2 Filtro de Hampel

El filtro de Hampel [30] se encarga de eliminar valores atípicos presentados en la señal, sin la necesidad de realizar un suavizado en los datos.

El funcionamiento de Hampel es similar al de mediana desde el punto de vista donde se utiliza una ventana de n vecinos a la izquierda y derecha de cada valor de la señal. Con esta ventana se calcula la mediana y además la desviación estándar. Si la muestra difiere en más de 3 desviaciones estándar se sustituye el valor en dicho punto por la mediana calculada [31].

En la figura 3.3 se puede observar el funcionamiento de este filtro, el cual se encarga de eliminar valores atípicos. Si bien ambas señales se ven muy similares, se puede observar cómo en la filtrada se presentan menos picos a comparación de la primer señal.

3.5 Selección de subportadoras

Una tarea importante dentro del preprocesamiento de datos es la reducción de dimensionalidad. Si un conjunto de datos posee un alto número de dimensiones esto aumenta los recursos necesarios para aplicar algoritmos de clasificación, y si la información que se encuentra dentro del conjunto no es relevante, puede provocar confusión en el sistema y que el resultado no sea el esperado.

3.5.1 minimum Redundancy Maximum Relevance (mRMR)

El algoritmo de mínima Redundancia Máxima Relevancia [32, 33] es un algoritmo de reducción de dimensionalidad que, como su nombre lo indica, se encarga de

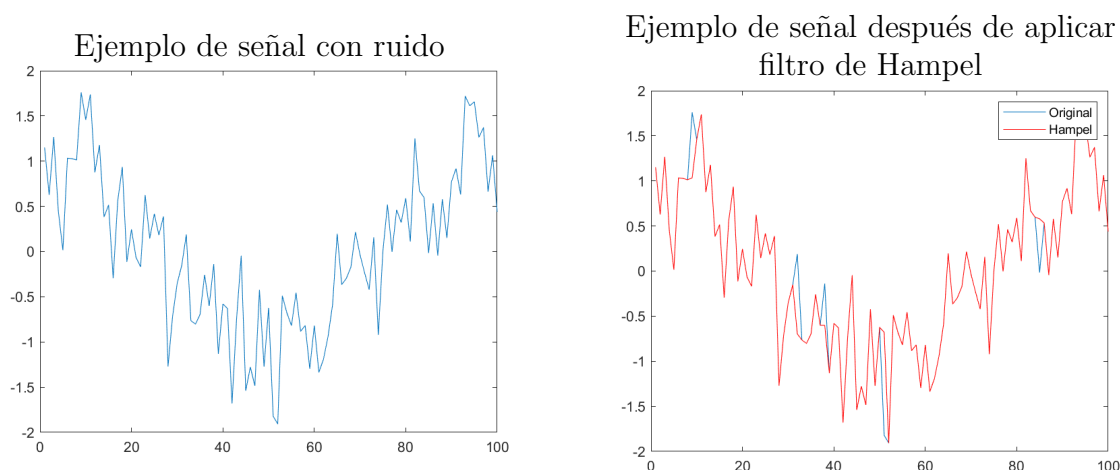


Figura 3.3: Ejemplo de uso del filtro de Hampel

minimizar la redundancia en un conjunto de datos eliminando información, mientras maximiza la relevancia de las variables dentro del mismo asegurando que la información eliminada es irrelevante o duplicada.

Mediante algoritmos heurísticos se cuantifica la importancia de una característica, permitiendo asignar una puntuación a cada característica dentro de un conjunto de datos para predecir la importancia de la misma. En base a esta caracterización se puede reordenar un conjunto de mayor a menor relevancia.

3.5.2 Selección por Forward y Backward Selection

Se implementó un método de selección de subportadoras para reducir la dimensionalidad del conjunto de datos basado en una combinación de estrategias de Forward Selection y Backward Selection [34].

Para el punto en el que se aplicó este método se utilizaron un total de 52 subportadoras para buscar la combinación ideal. Por cada una de las subportadoras se realizó una iteración en la etapa de Forward Selection, donde se agregó 1 subportadora al conjunto, se analizó el comportamiento y si mejoraba el reconocimiento se dejaba dicha subportadora dentro de las seleccionadas, si el rendimiento era igual o peor no se agregaba para no aumentar dimensionalidad.

Al revisar todas las subportadoras disponibles se pasa a la etapa de Backward Selection, donde ya se tiene un conjunto de datos con las subportadoras óptimas encontradas en la etapa anterior. La lógica para esta segunda etapa es, de las subportadoras encontradas es posible encontrar mejores rendimientos eliminando algunas repetidas o irrelevantes para el algoritmo. Por lo tanto se vuelve a analizar el comportamiento del sistema eliminando la primer subportadora para comparar los resultados. En caso de quitar una subportadora y mantener una respuesta igual

o mayor se mantiene el cambio de eliminar la subportadora y se continúa con la siguiente (la respuesta igual es para reducir dimensionalidad); por otro lado, si la respuesta disminuye, se mantiene la subportadora para mantener el rendimiento.

Si se encuentra un cambio durante la etapa de Backward Selection se activa una bandera que permite volver a la etapa de Forward Selection para comenzar a agregar únicamente las subportadoras que no se encuentran actualmente en la selección. Si en la etapa de Backward Selection no ocurre ningún cambio entonces no se activa la bandera y la selección de subportadoras termina con la lista actual.

En el algoritmo 2 se detalla el funcionamiento del mismo.

Algoritmo 2: Algoritmo forward y backward propagation

```

1 subportadoras ← [1, 2, 3, ..., 50, 51, 52];
2 existe_cambio ← true;
3 subportadoras_actual ← [];
4 tasa_mejor ← 0;
5 while existe_cambio do
6   existe_cambio ← false;
7   i ← 1;
8   while i ≤ subportadoras.length do
9     subportadoras_actual.add(subportadoras[i]) ; /* Agregar la
10      subportadora i para analizar su comportamiento */
11     tasa_actual ← evaluar_1DCNN() ; /* Evaluar el
12      comportamiento de la 1DCNN con la lista de
13      subportadoras actual */
14     if tasa_actual ≤ tasa_mejor then
15       subportadoras_actual.remove() ; /* Si no mejora la tasa,
16        se remueve la última subportadora agregada */
17     else
18       tasa_mejor ← tasa_actual ; /* Si se mejora la tasa, se
19        actualiza la mejor tasa */
20     i ← i + 1;
21   j ← 1;
22   while i ≤ subportadoras_actual.length do
23     subportadoras_aux = subportadoras_actual ; /* Hacer una copia
24      auxiliar de la lista de subportadoras para poder
25      eliminar y comparar */
26     subportadoras_aux.remove(j) ; /* Eliminar las subportadoras
27      por orden desde la primera a la última */
28     tasa_actual ← evaluar_1DCNN();
29     if tasa_actual ≤ tasa_mejor then
30       j ← j + 1 ; /* Si no se mejora la tasa, se aumenta j en
31        1 para evaluar eliminando la siguiente subportadora
32        */
33     else
34       tasa_mejor ← tasa_actual;
35       subportadoras_actual ← subportadoras_aux;
36       existe_cambio ← true;
37       ; /* Si sí se mejora la tasa, se actualiza la mejor
38        tasa, se toma como lista de subportadoras la
39        auxiliar, donde se elimina una de la lista, y se
40        cambia la bandera de existe_cambio */

```

3.6 Visualización de datos

Las técnicas de visualización de datos permiten tomar un conjunto de datos con múltiples dimensiones y generar un nuevo conjunto que conserve similitudes entre los datos, pero con un menor número de dimensiones para fines de visualización.

3.6.1 T-distributed Stochastic Neighbor Embedding

El algoritmo T-distributed Stochastic Neighbor Embedding [35] o T-sne por sus siglas en inglés, se encarga de generar un conjunto de datos con bajas dimensiones que conserve las distancias entre puntos que representa el conjunto original de altas dimensiones. De este modo es posible analizar gráficamente el comportamiento que presenta un conjunto de datos.

3.7 Algoritmos de Machine Learning

3.7.1 Clasificador Naive Bayes

Este algoritmo de clasificación es un procesamiento estadístico basado en la Teoría Bayesiana [36], donde se calcula la probabilidad que una muestra pertenezca a cierta clase basado en uniones de la distribución de las probabilidades que la muestra pertenezca a cada clase.

3.7.2 K Nearest Neighbor (KNN)

El clasificador KNN es uno de los métodos más simples para clasificar [37]. Este se basa en asignar etiquetas de clase a todos los datos dentro de un conjunto, y se puede predecir el valor de un punto de acuerdo al valor de los k vecinos que se encuentren más cerca de dicho punto. Dependiendo de la etiqueta de clase que tenga la mayor cantidad de vecinos, es como se asigna una clase a cada valor.

3.7.3 Clasificadores Discriminantes

Estos clasificadores calculan una función que permite dividir un conjunto de datos en 2 clases. Las funciones utilizadas en este experimento son de orden lineal y cuadrático.

Linear Discriminant Classifier (LDC)

El algoritmo LDC permite obtener una función lineal, la cual proporciona la capacidad de dividir un conjunto de datos con una línea recta generando así 2 sub-

conjuntos. Cada dato se clasifica de acuerdo al subconjunto en el que se encuentra después de la división.

Quadratic Discriminant Classifier (QDC)

Por otro lado, el algoritmo QDC arroja una función cuadrática, la cual otorga la capacidad de dividir el conjunto por una línea curva entre las clases. De igual manera, la curva se encarga de dividir un conjunto en dos subconjuntos y dependiendo de en dónde se encuentra cada punto es como se realiza la clasificación.

3.7.4 Support Vector Machines (SVM)

Las Support Vector Machines (SVM) [38] son algoritmos de Machine Learning diseñados para trabajar en conjuntos de datos de dos clases. Este algoritmo se basa en crear hiperplanos que permitan dividir el conjunto de datos en dos regiones.

Debido a que este trabajo utiliza más de dos clases es necesario aplicar técnicas que permitan clasificar con las SVM usando tres o más clases.

A continuación se muestra una descripción del funcionamiento de ambas herramientas suponiendo que se utiliza un conjunto de datos de 3 clases para simplificar el funcionamiento. Para este ejercicio donde se usan 9 clases, sólo hay que extender el funcionamiento de 3 hasta 9.

La primer técnica es *one vs. all* (uno contra todos) donde se clasifica si un elemento es de clase 1 o no pertenece a la clase 1, después si es de clase 2 o de cualquier otra clase que no sea la 2, y así para cada una de las clases presentes en el conjunto.

La segunda estrategia es *one vs. one* (uno contra uno) donde se clasifica si un dato pertenece a la clase 1 o la clase 2, después se compara si ese mismo elemento es de la clase 1 o clase 3, y por último se vuelve a comparar ahora con la clase 2 y clase 3. Esto permite comparar todas las clases, para saber cuál es la clase más propensa a representar a un dato.

Además de las estrategias se utilizan Kernels para generar el hiperplano. Esto permite realizar la división del conjunto de datos en un hiperplano Lineal, Polinomial o Función de Base Radial. En este experimento se probaron los 3 distintos Kernels para dividir un conjunto de datos.

3.8 Algoritmos de Deep Learning

En contraste con los métodos de Machine Learning, sólo se aplicó un método de Deep Learning, una Red Convolutiva Neuronal de 1 Dimensión (1 Dimension Convolutional Neural Network o 1DCNN).

3.8.1 One Dimensional Convolutional Neural Network - (1DCNN)

Una Red Neuronal Convolutiva (CNN) extrae características de un conjunto de datos a partir de estructuras convolucionales [39]. En una red convencional se utilizan imágenes en 2 dimensiones para extraer las características y poder predecir cada imagen.

Para este trabajo se utilizó la 1DCNN, que es una red que funciona con señales como entrada en lugar de imágenes. Esta no realiza convoluciones en 2 dimensiones, sino que sólo realiza convoluciones en una dirección. Eliminar una dimensión de la convolución hace que la red sea más rápida y ligera [40], permitiendo la capacidad de generar aplicaciones de tiempo real donde un microcontrolador se encarga de realizar el procesamiento de la red cargando un modelo preentrenado.

3.9 Jaula de Faraday

La jaula de Faraday es una caja recubierta por una malla de alambre o película metálica, cuya función principal es bloquear los campos eléctricos y ondas electromagnéticas [41]. La manera en que funciona la jaula es que debido a las paredes de la misma, el campo eléctrico en el interior es nulo, protegiendo así ante descargas eléctricas. Este funcionamiento se basa en las propiedades que presenta un conductor en equilibrio electrostático [42].

Las redes externas como telefonía e internet generan campos electromagnéticos que son recibidos por la caja metálica de la Jaula. Cuando se presenta este campo externo las cargas positivas se mantienen en las posiciones alrededor de la jaula, mientras que las negativas pueden moverse libremente, por lo que se desplazan hacia la fuerza del campo externo.

El movimiento de las cargas negativas hace que un lado de la jaula quede cargado negativamente, mientras que el otro positivamente, generando de esta manera un campo eléctrico en sentido opuesto. El resultante del campo externo y el campo interno en sentido opuesto ocasiona un campo nulo, y sin un campo ninguna carga puede atravesar la caja.

En este capítulo se explicó el funcionamiento de las herramientas y técnicas usadas para este trabajo. Desde qué es y cómo se obtiene el CSI, hasta los distintos algoritmos de filtros, reducción de dimensionalidad y clasificación. El funcionamiento del sistema desde la recolección del CSI, hasta cómo se llegó a aplicar los algoritmos de clasificación se encuentran en el siguiente capítulo.

4. Materiales y métodos

En este trabajo de conteo de personas se realizaron dos experimentos principales. Cada experimento se encarga de realizar las tareas de recopilación de datos, generación de conjuntos de datos para cada experimento, limpiar los conjuntos con preprocesamiento y aplicar algoritmos tanto de Machine Learning como Deep Learning. Para cada experimento se aplicaron dos grupos de actividades separadas por un primera configuración donde aparecieron áreas de mejora, y una segunda configuración donde se mejoraron los resultados de clasificación de manera considerable en relación a la primer configuración de cada experimento.

Las dos configuraciones realizadas en el primer experimento se encuentran enfocados hacia la utilización de algoritmos de Machine Learning, y un algoritmo de Deep Learning. Por otro lado, el segundo experimento se utilizó únicamente un algoritmo de Deep Learning, con la ventaja que al ser sólo un algoritmo, el conjunto de datos se encuentra optimizado para dicho algoritmo.

4.1 Primer experimento

4.1.1 Metodología del sistema

El primer experimento utilizó la comunicación de dos antenas para enviar paquetes desde transmisor hacia receptor mediante una red WiFi. Durante el envío de datos en la red WiFi se genera la Información del Estado del Canal (CSI), la cual sirve para describir el ambiente donde se realiza la conexión y optimizar el uso del canal de la red.

Con los valores del CSI que se obtienen en un periodo determinado de muestreo es posible generar señales que describan el nivel de ruido durante la comunicación de la red. Como el CSI informa la respuesta de magnitud y fase de cada subportadora, es cuestión de seleccionar la información importante, en este caso, sólo se tomó la amplitud para generar la señal a partir del CSI.

Una vez que se tienen listas las señales a partir del CSI se aplicaron los algoritmos de preprocesamiento de datos para limpiar la señal y poder mejorar las tasas de reconocimiento. Después de limpiar las señales se realizó una etapa de extracción de características a partir de medidas estadísticas. La etapa de extracción de características ayuda para alimentar los algoritmos de Machine Learning y Deep Learning, donde se introduce cada característica extraída y se obtiene de resultado una etiqueta perteneciente a cada clase.

Dependiendo del número de personas que se encuentre dentro de la habitación se genera un nivel de ruido distinto. De acuerdo a la intensidad del nivel de ruido, es posible conocer a través de algoritmos de clasificación el total de personas que se encuentra dentro.

En la figura 4.1 se muestra el diagrama de bloques que representa el flujo de trabajo para el experimento inicial.

4.1.2 Recursos

Para este primer experimento se creó un conjunto de datos utilizando una habitación dentro del Instituto de Ingeniería con medidas de 3.5m x 6.0m. La figura 4.2 muestra un diagrama de ubicación física de la habitación y la figura 4.3 muestra una imagen real de la habitación. Las antenas se colocaron de esquina a esquina con una distancia de 6.5m entre ellas. Las paredes dentro de la misma se encuentran

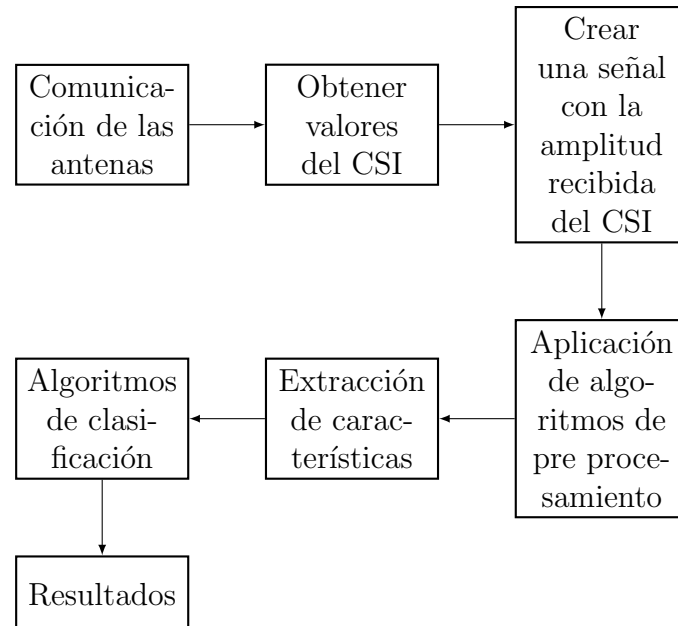


Figura 4.1: Primer experimento

4.1 Primer experimento

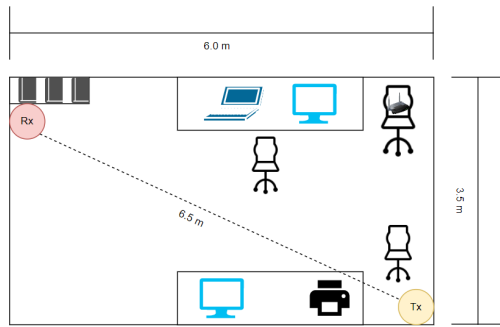


Figura 4.2: Diagrama de ubicación física de las antenas

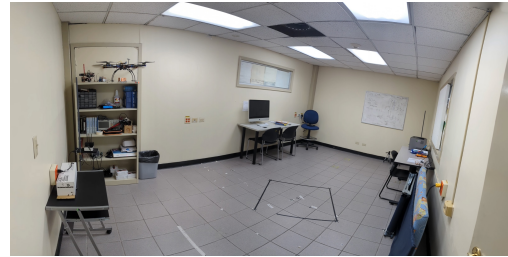


Figura 4.3: Imagen de la habitación

construidas de hojas de cartón de yeso reforzadas con metal galvanizado.

Dentro de esta habitación se colocaron dos computadoras Intel NUC Kit equipadas con un procesador Intel® Core™ i5-4250U y 4 GB de RAM. La tarea principal de estos equipos es enviar información de transmisor a receptor mediante una red WiFi para obtener el CSI de la comunicación entre éstas. Una intel NUC se configuró como transmisor (figura 4.4) contando con una antena, mientras que la segunda Intel NUC se configuró como receptor (figura 4.5) con un total de tres antenas, produciendo tres canales de comunicación. La información enviada son paquetes de información, configurados a una tasa de @25 Hz, con un total de 400 paquetes por observación, durando aproximadamente 16 segundos en realizar cada observación.

La finalidad de estas Intel NUC sólo fue recopilar la información, por lo que una vez obtenidas las observaciones del CSI, se extrajeron los resultados para ser analizados dentro de otro equipo distinto. Para la tarea del preprocesamiento de datos, así como aplicar los algoritmos de Machine Learning y Deep Learning se utilizó una laptop personal con un procesador Intel® Core™ i7- 7820HQ y 16 GB de RAM.

Las Intel NUC Kit implementan el estándar IEEE 802.11n, permitiendo obtener 30 subportadoras por cada par de antena. En este caso se utiliza una antena de transmisor y tres de receptor lo que permite aumentar las subportadoras desde 30 hasta 90.

El alcance para este primer experimento fue identificar tanto el nivel de ocupación de una habitación, como el número total de personas que se encuentran dentro de la misma. Para el nivel de ocupación se realiza tomando en cuenta 2 clases principales: la habitación vacía que representa 0, 1 o 2 personas dentro de la habitación, y la habitación llena, donde se tiene un total de 3 o más personas (desde 3 hasta 9).

Para el total de personas se utilizaron nueve clases distintas, cada clase re-

4.1 Primer experimento



Figura 4.4: Intel NUC transmisor con una antena



Figura 4.5: Intel NUC receptor con tres antenas

presenta un total distinto de personas dentro de la habitación, donde la clase 1 significa que son 0 personas, la clase 2 es 1 persona dentro, la clase 3 son 2 personas, hasta llegar a la clase 9 donde se tienen 8 personas dentro de la misma. En la tabla 4.1 se observa la relación entre el total de personas y las clases definidas tanto para clasificar el nivel de ocupación como para contar a las personas dentro de la habitación.

Las señales obtenidas a partir del CSI se realizaron con la herramienta Linux 802.11n CSI Tool kit [43]. Esta herramienta arroja un timestamp para representar la señal, la amplitud y la fase de cada subportadora. De acuerdo al total de pares de antena y el protocolo de la herramienta utilizados para la recolección de CSI, se consiguió un total de 90 subportadoras, por lo que la información obtenida es el timestamp, 90 valores de amplitud y otros 90 de fase de las subportadoras. Para este trabajo, sólo es relevante la información obtenida a través de la amplitud de cada subportadora, ignorando la fase debido a que la fase del CSI presenta errores constantes producidos por un offset que se encuentra presente a causa de

Total de personas	0	1	2	3	4	5	6	7	8
Clase en nivel de ocupación	1			2					
Clase para conteo de personas	1	2	3	4	5	6	7	8	9

Tabla 4.1: Relación con el total de personas y las clases para las actividades

la desincronización de los relojes del par de equipos utilizados para la transmisión [44].

Las señales generadas a partir del CSI tienen un nivel de ruido distinto dependiendo del entorno donde se obtuvieron las muestras. Este entorno incluye tanto el material del entorno donde se realiza la lectura (un cuarto, al aire libre, etc.) como los muebles, objetos e incluso personas que se encuentren dentro en el momento que se lleva a cabo la lectura entre el par de antenas. Para realizar el conteo se utilizó una habitación donde lo único que cambiaba era la cantidad de personas dentro, ya que así se puede asegurar que el ruido generado por los muebles y objetos sea el mismo, y lo único que varía es el ruido provocado por el factor humano.

Se realizaron un total de 10 observaciones por clase, es decir, 10 observaciones de la habitación vacía, 10 observaciones con 1 persona dentro, continuando hasta tener las últimas 10 observaciones con 8 personas dentro de la misma. Durante cada observación, las personas dentro de la habitación se posicionaron en un lugar específico sin moverse durante los 16 segundos que se realizaba la comunicación entre los pares de antenas, una vez finalizada una observación, todos los participantes cambiaban de lugares a otros que no habían sido utilizados, teniendo en cuenta que en un ambiente real, las personas se pueden posicionar en cualquier lugar dentro de una habitación.

En la figura 4.6, se puede observar el comportamiento que posee una señal construida a partir del CSI para cada uno de los distintos totales de personas que se encuentren dentro de la habitación. Las señales que se muestran en las gráficas son las 6 amplitudes de las subportadoras más relevantes de las 90 que se recaban en total.

Estas 6 señales principales se seleccionaron de entre las 90 como resultado del algoritmo de reducción de dimensionalidad mRMR. Las señales son sólo representativas para ver el comportamiento que se obtienen dependiendo del número de personas, ya que en cada clasificación se vuelve a buscar y seleccionar las señales más representativas para la clasificación.

4.1.3 Construcción del primer experimento - configuración A (E1A)

Para identificar los experimentos que estamos evaluando se identifican con una etiqueta, este es el experimento 1 configuración A, etiquetado como E1A.

En este primer experimento se cuenta con 90 pares de amplitud y fase de las subportadoras, sin embargo, en E1A sólo se tomaron en cuenta las 30 subportadoras pertenecientes a el primer par de antenas de los 3 pares disponibles, es decir, de 90 subportadoras recabadas del CSI, sólo se tomaron 30 lo que ocasionó pérdida

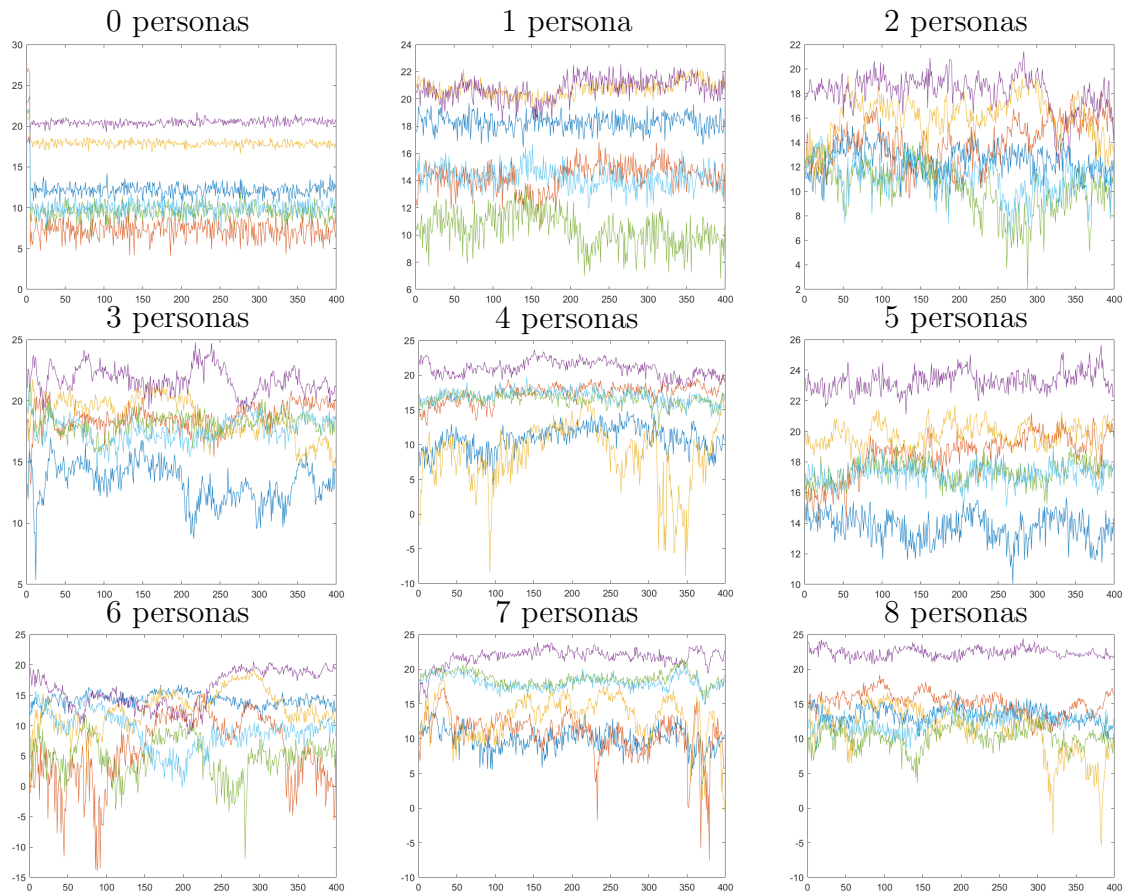


Figura 4.6: Subportadoras principales para el primer conjunto de datos

de información relevante de los datos, abriendo áreas de oportunidad para realizar una nueva configuración en este experimento.

Preprocesamiento de datos

De acuerdo a la configuración de la comunicación de las antenas, se esperan recibir los paquetes a una frecuencia de 25 Hz, es decir, 25 paquetes por segundo, con un total de 400 paquetes por observación, sin embargo, durante el proceso de recolección de datos, no todas las observaciones cumplieron el total de 400, sino que hubo algunas con 399 paquetes registrados. Por esta razón, se regularon todas las observaciones para cumplir con un total de 399 paquetes por observación.

A partir de estos treinta valores de amplitud se realizaron técnicas de preprocesamiento de datos. Durante la recopilación de datos algunos paquetes se perdieron en la comunicación de transmisor a receptor, los cuales eran representados con un valor -Inf. Debido a esta pérdida de información se decidió aplicar una técnica de imputación de datos basado en vecinos cercanos, sustituyendo los valores -Inf por el valor calculado mediante la técnica de imputación de datos.

Para cada cambio dentro de la etapa de preprocesamiento el conjunto de datos se fue modificando de manera escalonada. El conjunto de datos inicial se obtuvo después de regular el total de paquetes y corregir los valores perdidos para evitar discrepancia dentro del conjunto. Las dimensiones del conjunto resultante se muestran en la figura 4.7, donde se tienen 30 subportadoras de información, con un total de 399 paquetes por subportadora, para 90 observaciones (10 observaciones para cada clase, por un total de 9 clases).

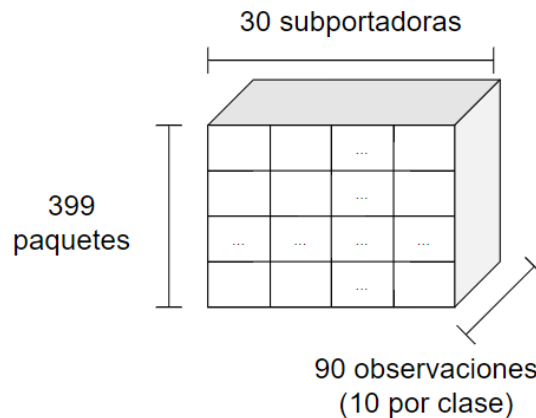


Figura 4.7: Primera versión del conjunto de datos de E1A

Valores atípicos y eliminación de ruido

Para poder eliminar valores especiales que se encuentren muy distantes del resto del conjunto se utilizó el filtro de Hampel, eliminando así los valores atípicos que pueda presentar la señal. Después de ello, para poder suavizar la señal, se aplicó un filtro de mediana para reducir la potencia del ruido que se pueda presentar.

Para optimizar los parámetros de Hampel y filtro de mediana, se utilizaron valores desde 1 hasta 25 con incrementos de 2 en 2 (1, 3, 5, ..., hasta 25) para encontrar una configuración óptima para los filtros de Hampel y mediana. Con esta lista de parámetros se tienen 13 opciones para Hampel y otras 13 para el filtro de mediana, dando un total de 169 combinaciones posibles para buscar la configuración óptima de los filtros de Hampel y mediana.

En la figura 4.8 se muestra la señal para 0 personas, así como el resultado de aplicar ambos filtros tanto de Hampel como el de mediana. El resto de señales se muestra en la sección 7, Apéndice A. Las gráficas se muestran por tercias dentro de la misma fila, mostrando primero la señal original, la señal después del filtro Hampel y por último la señal después de filtros Hampel y mediana. La fila siguiente es la medición del siguiente número de personas.

Extracción de características

Para cada una de las 30 subportadoras que sirvieron de base para generar cada señal se realizó una extracción de características de dos medidas estadísticas, únicamente media y mediana, para cada señal construida por 399 paquetes.

Con la extracción se obtuvo de resultado el conjunto de datos que se muestra en la figura 4.9, un total de 30 amplitudes a partir de las subportadoras, dos medidas estadísticas y 90 observaciones (10 observaciones por clase, con 9 clases).

Reescribiendo el conjunto de datos anterior se modificaron las dimensiones

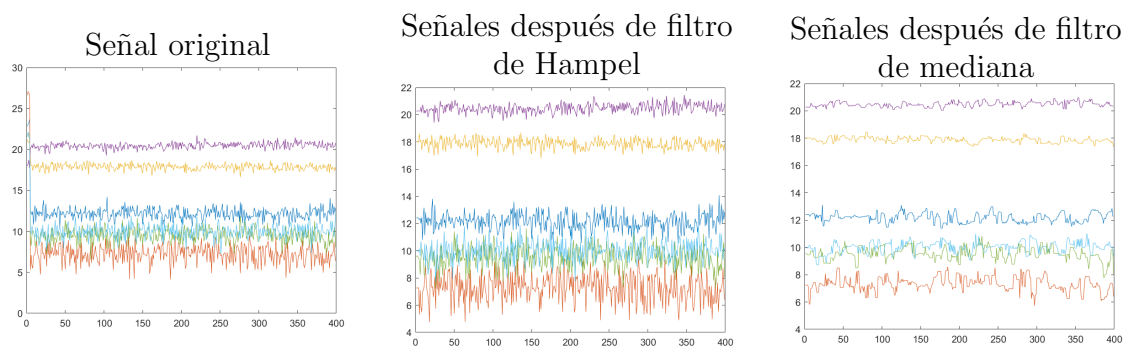


Figura 4.8: Señal para 0 personas antes y después de aplicar los filtros de Hampel y mediana en el primer conjunto de datos

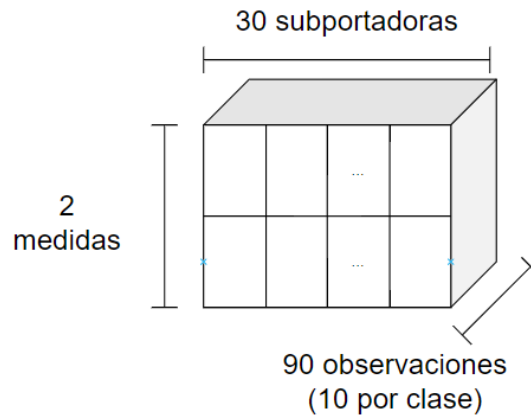


Figura 4.9: Segunda versión conjunto de datos de E1A

del mismo, uniendo las 30 subportadoras de media con las 30 subportadoras de la mediana y permitir formar las 60 características extraídas. Esto devuelve el conjunto a evaluar de la figura 4.10, el cual posee 60 características extraídas, 10 observaciones por clase, y un total de 9 clases.

Para mejorar los resultados del primer experimento E1A se modificó la estructura de los datos evaluados con una nueva configuración. Esta segunda configuración tiene como objetivo aumentar el volumen de la información evaluada aumentando el total de las subportadoras, además de seleccionar un mayor número de medidas estadísticas para realizar la extracción de características. Con este aumento de datos en el conjunto se planea buscar una mayor tasa de reconocimiento.

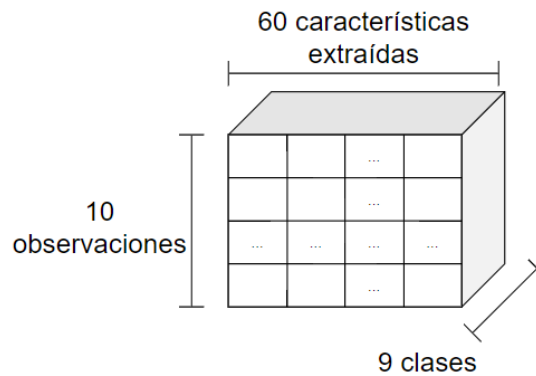


Figura 4.10: Tercera versión conjunto de datos de E1A

4.1.4 Construcción del primer experimento - configuración 2 (E1B)

A diferencia E1A, en esta segunda configuración etiquetada E1B sí se tomaron en cuenta las 90 subportadoras disponibles, por lo que la información obtenida del CSI aumentó de manera considerable. Al modificar el total de subportadoras, también se presentaron cambios en las dimensiones del conjunto de datos de base, mostrándose ahora como en la figura 4.11 el cual es similar a E1A, con la diferencia del aumento de subportadoras.

Preprocesamiento de datos

El conjunto de datos para el experimento E1B se creó con la misma información recopilada en el experimento E1A. Debido a esto, el conjunto de datos presenta los mismos detalles relacionados al preprocesamiento. Lo primero fue regular de misma manera el total de paquetes que contiene cada señal desde 400 hasta 399 debido a la pérdida del primer dato durante la transmisión.

Asimismo es necesario aplicar un algoritmo de imputación de datos por los valores perdidos en la recolección de datos que se representan como -Inf. La diferencia en este método de imputación, es que al tener ahora 90 subportadoras en lugar de 30, ocurren dos nuevos detalles. El primero de ellos es que es posible encontrar nuevas subportadoras que sean más parecidas a la que tienen los datos perdidos, permitiendo aumentar la confianza en este método. El otro inconveniente es que al aumentar las subportadoras también aumentan los posibles datos perdidos.

El conjunto de datos consistió de 10 observaciones por clase (90 observaciones en total), con dimensiones de 399 por 90 (399 paquetes por observación y 90 amplitudes de las subportadoras). Para tener un conjunto de datos más extenso,

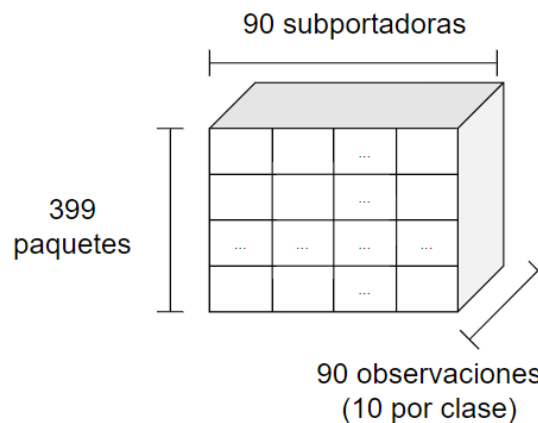


Figura 4.11: Primera versión del conjunto de datos de E1B

se divide cada observación en 3 subseñales, utilizando ventanas de 133 paquetes, permitiendo así modificar el conjunto de datos como se ve en la figura 4.12 el cual consta de 90 subportadoras de 133 paquetes de información con un total de 270 observaciones (30 observaciones por clase, con un total de 9 clases).

Valores atípicos y eliminación de ruido

En esta segunda creación de un conjunto de datos, también se implementaron los filtros de Hampel y mediana, para eliminar valores atípicos y ruido, respectivamente. Sin embargo, no se realizó una búsqueda por los parámetros óptimos para ambos filtros, sino que se aplicó la misma configuración encontrada en la creación del primer conjunto.

Extracción de características

Se realizó una extracción de características de 8 medidas estadísticas para cada una de las 90 subportadoras que se tiene en cada señal. En cada extracción se utilizó como entrada una señal con un total de 133 paquetes. Las medidas estadísticas usadas fueron la media, mediana, varianza, desviación estándar, curtosis y el momento central en segundo, tercero y cuarto orden.

Con estas 8 medidas estadísticas extraídas se modificó el conjunto de datos tal como se ve en la figura 4.13 contando con 90 subportadoras, de 8 medidas estadísticas y 270 observaciones. Este mismo conjunto se reordenó para combinar las subportadoras con las medidas estadísticas y obtener así las características extraídas, obteniendo un total de 720 características extraídas, 30 observaciones por clase y 9 clases, como se ve en la figura 4.14.

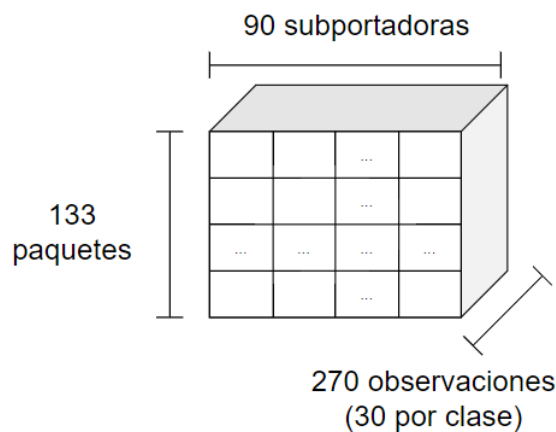


Figura 4.12: Segunda versión conjunto de datos de E1B

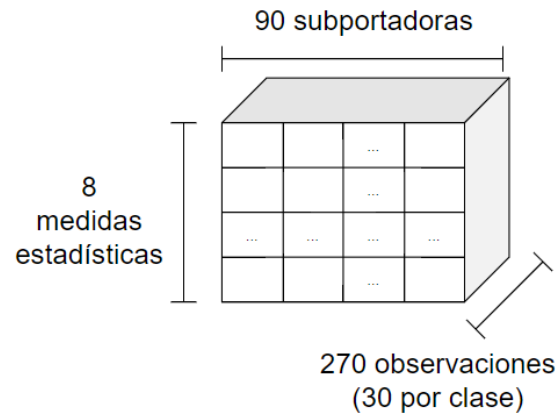


Figura 4.13: Tercera versión del conjunto de datos de E1B

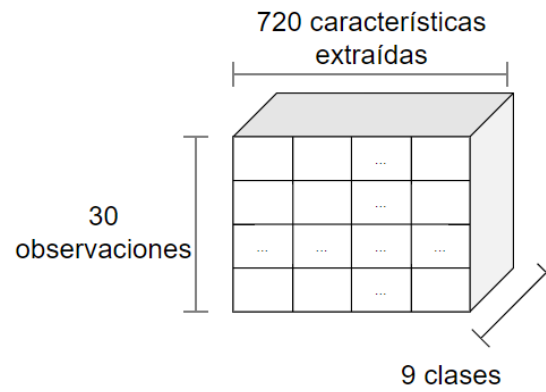


Figura 4.14: Cuarta versión del conjunto de datos de E1B

Reducción de dimensionalidad

Como resultado de la extracción de características, el conjunto de datos fue transformado para tener 270 observaciones por 720 características extraídas. Debido al gran número de dimensiones, el resultado de los algoritmos de clasificación puede contener un alto nivel de ruido. Por esta razón, se utilizó el algoritmo *minimum Redundancy - Maximum Relevance (mRMR)*, permitiendo reducir las dimensiones del conjunto de 720 a 30 principales características.

Como resultado después de aplicar la reducción de dimensionalidad en el conjunto de datos, se tiene un conjunto con 30 características extraídas, 30 observaciones por clase y un total de 9 clases, como se ve en la figura 4.15.

Conjunto de datos para nivel de ocupación

Para medir el nivel de ocupación en la habitación se utilizó el mismo conjunto de datos, es decir, las mismas técnicas aplicadas de preprocesamiento de datos donde se incluye imputación de datos, filtros de valores atípicos y reducción de ruido, extracción de características y reducción de dimensionalidad.

La diferencia con este conjunto de datos, es que en lugar de tener un conjunto de 9 clases, únicamente se tienen 2 clases, en la primera clase se agruparon todos los resultados de extracción de características de las clases 1, 2 y 3, es decir, 0, 1 y 2 personas en la habitación para tener la clase de "habitación vacía". Dando un total de 90 muestras en la nueva clase 1.

Para la segunda clase se utilizaron los resultados de extracción de características de las clases 4, 5, 6, 7, 8 y 9, es decir, 3, 4, 5, 6, 7 y 8 personas dentro de la habitación para tener una clase de "habitación llena". Al agrupar todas estas clases se obtiene de resultado un total de 180 muestras para la clase 2, lo cual genera una desigualdad de 1:2 con respecto a la clase 1. Por esta razón no se

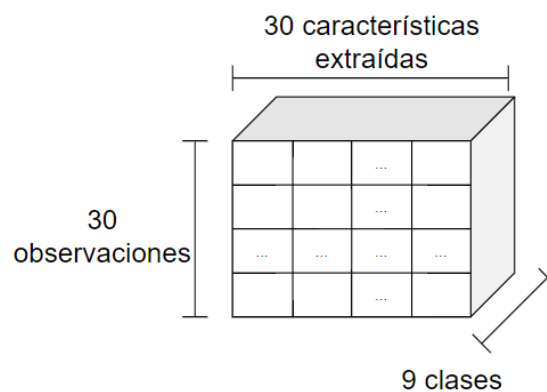


Figura 4.15: Versión final conjunto de datos de E1B

tomaron en cuenta todas las muestras para generar la clase 2, sino que se tomaron la mitad de muestras de la clase 4, la mitad de la 5, y así para cada clase, dando un total de 90 muestras para la clase 2, permitiendo equilibrar el conjunto de datos.

El conjunto de datos generado al final se ve en la figura 4.16 es un total de 30 características principales extraídas, 90 observaciones por clase y un total de sólo 2 clases (lleno y vacío).

Con esto ya se tienen listos los conjuntos de datos a los cuales aplicar métodos de clasificación tanto de Machine Learning como Deep Learning, para obtener el número de personas dentro de la habitación y el nivel de ocupación de la misma.

4.2 Jaula de Faraday

El primer experimento en sus dos configuraciones se llevó a cabo en una habitación dentro del Instituto de Ingeniería donde las paredes del mismo están construidas de hoja de cartón de yeso reforzado con metal galvanizado.

Las antenas utilizadas se encuentran conectadas por medio de una red con frecuencia de 2.4 GHz. Al trabajar con esta red WiFi es posible que ocurran interferencias causadas por dispositivos o redes externas. El ruido generado en la señal de CSI puede provocar que las mediciones no sean precisas.

Para cubrir la necesidad de analizar señales de CSI en un espacio controlado, se decidió construir una jaula de Faraday con materiales proporcionados por el Instituto de Ingeniería, en el que se incluyen tubos de acero para construir la estructura de la jaula, y mallas de cobre para cubrir las paredes, techo y piso de la jaula.

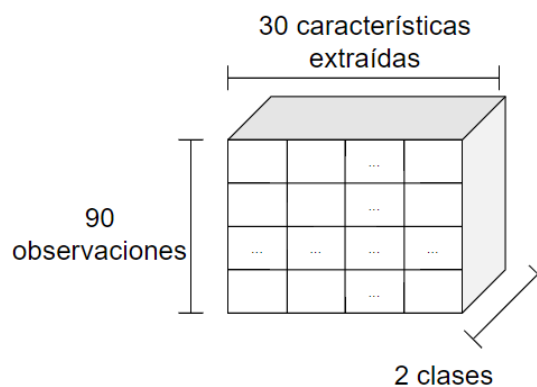


Figura 4.16: Conjunto de datos para 2 clases de E1B

4.2.1 Cálculos de la malla

De acuerdo a Eliot Eggers [45] para poder diseñar una jaula de Faraday que se mitigue las señales WiFi es necesario que la frecuencia de bloqueo de la jaula sea mayor con respecto a la frecuencia de la señal que desea ser bloqueada. De acuerdo a los textos de 1940 del físico Hans Bethe, el tamaño de los agujeros que debe tener una jaula no deben ser mayores a un décimo del tamaño de longitud de onda (λ) de la señal para funcionar idealmente.

Con esto dicho, se puede calcular la frecuencia de la cual es capaz de atenuar la jaula con la fórmula de frecuencia:

$$f = c/\lambda \quad (4.1)$$

donde f es la frecuencia que se está calculando, c es la velocidad de la luz (aproximadamente 3×10^8), y λ corresponde a la longitud de onda, el cual se obtiene al multiplicar por un valor de 10 el tamaño de los agujeros de la malla.

En este caso, las mallas presentan un tamaño de 2 mm, con el que se obtiene una longitud de onda de 20 mm (0.020 m).

$$\lambda = 0.002m * 10 = 0.020m \quad (4.2)$$

Conociendo la longitud de onda se calcula la frecuencia que atenúa la malla dividiendo la velocidad de la luz sobre la longitud de onda.

$$f = \frac{3 \times 10^8 m/s}{0.020m} = 15 \times 10^9 Hz = 15 GHz \quad (4.3)$$

Teóricamente, la jaula es capaz de atenuar señales de hasta 15 GHz, lo cual es más que suficiente para las señales de 2.4 GHz y 5 GHz que producen las redes WiFi.

4.2.2 Construcción de la Jaula

La jaula de Faraday se construyó con dimensiones de 2.85 m x 1.95 m x 3.43 m. El primer paso fue cortar los tubos que actúan como esqueleto de jaula, debido a que las dimensiones que presentaban los tubos sobrepasaban las dimensiones del cuarto en donde se planeó construir la jaula. Para cortar todos los tubos sin problemas se utilizó un esmeril para cortar los tubos y después lijarlos, permitiendo así usar los conectores sin problemas.

Una vez acertadas las dimensiones de los tubos, el siguiente paso fue construir la estructura de la jaula (figura 4.17), la cual es el conjunto de tubos de metal unidos por conectores en sus 8 vértices (figura 4.18).

Después de construir la estructura se colocaron las puertas, el techo y por último el suelo de la jaula, utilizando la malla de cobre para los 6 lados de la jaula.



Figura 4.17: Construcción de la estructura de la jaula



Figura 4.18: Estructura de la jaula

Dicha malla se encontraba en un estado no óptimo, presentando algunos agujeros. Para colocar la malla dentro de la jaula fue necesario menos material del que se tenía a disposición, por lo que de los trozos excedentes de malla se utilizaron para agregar parches en las zonas que se encontraban con agujeros (figuras 4.19 y 4.20).

Una vez construida la jaula, es necesario tener la certeza que exista continuidad en toda la jaula para poder bloquear los campos eléctricos de una manera efectiva. Para verificar la continuidad, se utilizó un multímetro para ver los distintos puntos de la jaula, como en la figura (figura 4.21) donde se está revisando la continuidad de una pared con el techo de la jaula.

4.2.3 Reforzamiento de la jaula

Una vez finalizada la construcción de la jaula se realizaron pruebas en la misma con una radio, donde las señales van desde 88 hasta 108 MHz y la señal no pudo ser mitigada ya que el programa en la radio se continuaba escuchando sin ningún cambio en el sonido.

Para mejorar el resultado de la atenuación de la jaula se propuso realizar una prueba forrando una caja pequeña de aluminio para meter un teléfono celular e intentar realizar una llamada buscando que la señal no pueda encontrar al teléfono.



Figura 4.19: Ejemplo de un parche en la malla



Figura 4.20: Ejemplo de un segundo parche en la jaula



Figura 4.21: Medición de continuidad en la jaula

Durante esta prueba se recubrió capa por capa, y en cada vuelta se revisaba si el teléfono de celular podía aún recibir llamadas. Al final de 8 vueltas, la comunicación del teléfono con el exterior dejó de ser posible, se probó tanto con llamadas directa de telefonía móvil, como con mensajes y llamadas a través de internet. En ambas pruebas el resultado fue el mismo, donde no se logró concretar las llamadas debido a que la caja forrada de aluminio bloqueaba con éxito tanto las señales de telefonía móvil como internet externas.

Debido al éxito presentado en las pruebas con la caja forrada, se decidió realizar modificaciones a la jaula. Primeramente se realizaron cálculos para saber cuánto material es necesario. Se optó por comprar 3 paquetes de papel aluminio de 400 m, con los que, idealmente alcanza 1 paquete para cubrir 3 vueltas tanto a paredes como suelo y techo en la jaula.

Con el aluminio se recubrió la jaula para que de esta manera se apliquen tanto los efectos de la malla como del papel aluminio y se logre una atenuación mucho mayor en la jaula. Después de tener las primeras 3 capas listas se hizo una prueba más con la radio, para verificar si el reforzamiento presentaría atenuaciones en la jaula. Como resultado de dicha prueba la señal presentó interferencia al momento de escuchar la radio, razón por la que se continuó con el reforzamiento.

Otra prueba realizada fue medir las unidades de decibelios que presenta la red WiFi en la jaula, donde al momento de únicamente entrar se tiene un valor de 100, y al cerrar la puerta y aterrizar la jaula se presenta una atenuación de hasta 50 decibelios en la señal de WiFi.

Debido al tiempo que consumió la creación de la jaula, su reforzamiento finalizó al tener listas 5 capas tanto en paredes como en suelo y techo. Al tener listo el recubrimiento se prosiguió a realizar una segunda recolección de datos, para generar un conjunto de datos dentro de un ambiente controlado.

4.3 Segundo experimento

4.3.1 Metodología del experimento

El segundo experimento es similar al primero, con la diferencia que se eliminó la etapa de extracción de características.

La comunicación de las antenas se llevó a cabo a través de un par de antenas transmisor-receptor. Esta comunicación permite obtener el CSI de la red WiFi, y a partir de la misma, se generó una señal con la información recibida en la amplitud del CSI. Una vez generadas las señales se implementan algoritmos de preprocesamiento de datos para limpiar las señales.

La principal diferencia radica en el paso e) del flujo de trabajo representado en el diagrama a bloques de la figura 4.22. En los dos experimentos siguientes (E2C

y E2D) no se realizó una extracción de características, sino que se utilizó la señal completa como entrada después de un preprocesamiento. El algoritmo utilizado es una 1DCNN, la cual trabaja con señales, por esta razón no se realizó la etapa de extracción de características, sino que se usó la señal como entrada para obtener una etiqueta en cada señal correspondiente a cada clase.

4.3.2 Recursos del experimento

El proceso de muestreo para este segundo conjunto de datos llevó algo de tiempo en comenzar, pues fue necesario terminar el trabajo de la jaula de Faraday, además de comprender el uso del Toolkit para generar señales a partir del CSI de los dispositivos ESP32 para realizar un cambio en los dispositivos de recolección de datos. Una de las razones del cambio entre equipo de medición es que los ESP32 permiten tomar muestras a una mayor frecuencia en comparación con las computadoras Intel NUC Kit utilizadas en la creación del conjunto anterior.

Los ESP32 usados para realizar la recolección del CSI se colocaron dentro de la jaula de Faraday. Se utilizó únicamente 1 antena en estación (figura 4.23, este ESP32 se energiza a la corriente para comunicarse con el segundo ESP32) y 1 en el punto de acceso (figura 4.24, este ESP32 se energiza conectando a una PC para obtener el CSI y almacenarlo como un archivo .txt), obteniendo 52 subportadoras en total.

La idea de este segundo conjunto de datos fue tener un sistema en el que no es necesario aplicar extracción de datos, sino que las señales obtenidas puedan ser

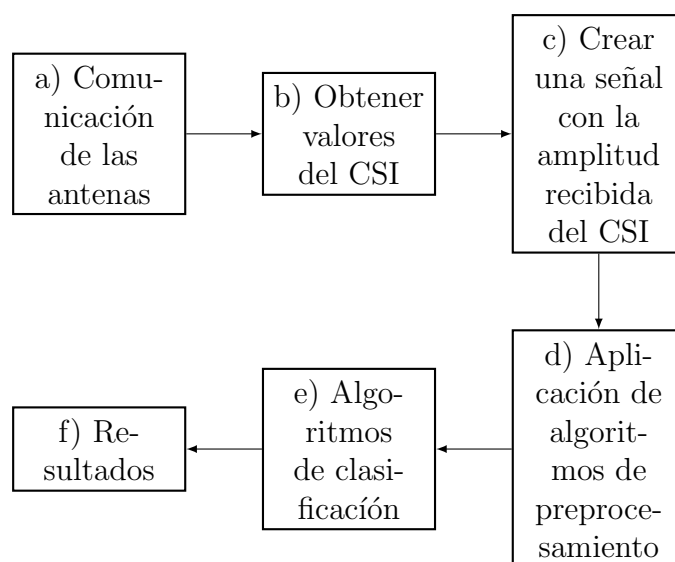


Figura 4.22: Diagrama a bloques del segundo experimento



Figura 4.23: ESP32 que actúa como estación



Figura 4.24: ESP32 que actúa como punto de acceso

introducidas en un algoritmo de Deep Learning para crear un sistema que pueda calcular en tiempo real el total de personas dentro de la habitación.

En este segundo conjunto de datos se redujo el total de clases de 9 hasta 6, es decir, el total de personas dentro de la habitación en el conjunto usado para los experimentos E1A y E1B fue de 0 a 8 y en este conjunto para los experimentos E2C y E2D es de 0 a 5 personas dentro.

La comunicación de los dispositivos se configuró con una frecuencia de 50 Hz durante 4 minutos, es decir, durante 240 segundos se enviaron 50 paquetes cada segundo, dando un total de 12,000 paquetes por observación. En total, se realizaron 5 observaciones de 4 minutos para cada una de las 6 clases, durando alrededor de 120 minutos en todas las muestras. El conjunto de datos que se logró conseguir en este proceso es de 52 subportadoras de información, con un total de 12,000 paquetes en cada subportadora, para 30 observaciones (5 observaciones por cada

4.3 Segundo experimento

clase, con un total de 6 clases). En la figura 4.25 se expresa este conjunto de datos.

Durante la conexión se perdieron algunos paquetes, debido a que en las observaciones llegaban aproximadamente el 98% de los 12,000 paquetes, es decir, se tenían aproximadamente 11,800 paquetes por observación. Debido a esto, se decidió establecer el total de paquetes en el 95% de la capacidad esperada, en 11,400. Este ajuste en el total de datos recibidos para cada subportadora modificó el conjunto de datos actual, contando ahora con el que se muestra en la figura 4.26.

Para cada una de las 6 clases, se realizaron un total de 5 observaciones, de las cuales se capturaron 11,400 paquetes por observación. Para incrementar el número de observaciones del conjunto, se utilizaron ventanas de tiempo para dividir cada observación en 12 muestras, es decir, de los 4 minutos que dura 1 observación, se

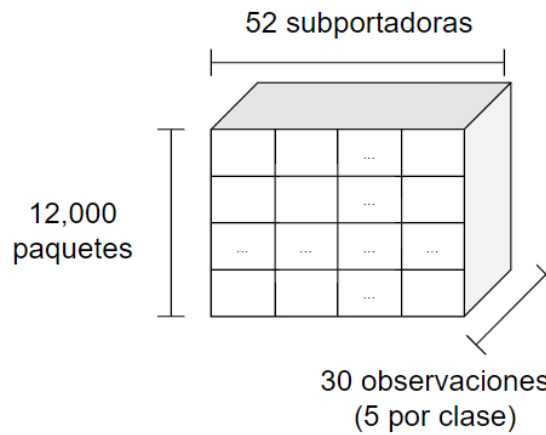


Figura 4.25: Primera versión del conjunto de datos de E2C

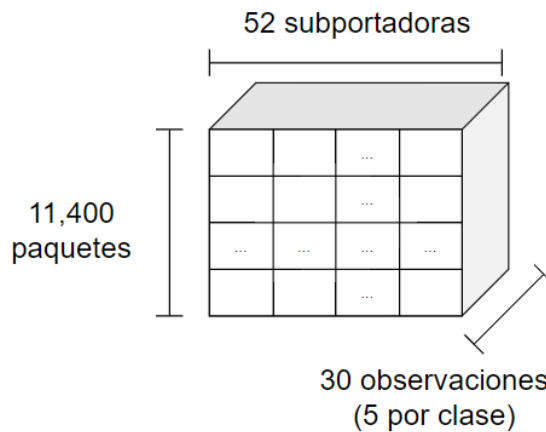


Figura 4.26: Segunda versión del conjunto de datos de E2C

tomaron 20 segundos para crear cada ventana.

Como resultado se obtuvo un conjunto de datos con las mismas 52 subportadoras, pero con un total de paquetes por observación menor disminuyendo desde 11,400 hasta 950, lo que permitió aumentar doce veces el número de observaciones desde 30 observaciones (5 por cada clase) hasta 360 observaciones en total (60 observaciones por cada una de las 6 clases). En la figura 4.27 se ve la representación del conjunto de datos obtenido.

Se puede reordenar el conjunto de datos, de tal manera que, al saber que una subportadora contiene 950 paquetes es posible considerar esto como una señal. Esto permite ver el conjunto desde el punto de vista donde se cuenta con 52 señales por observación, con un total de 60 observaciones por clase y con 6 clases, tal como se muestra en la figura 4.28.

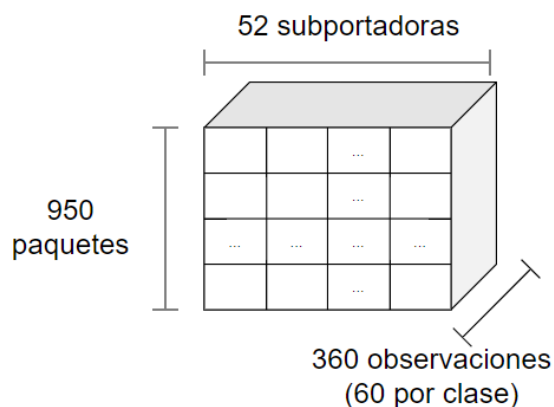


Figura 4.27: Tercera versión del conjunto de datos de E2C

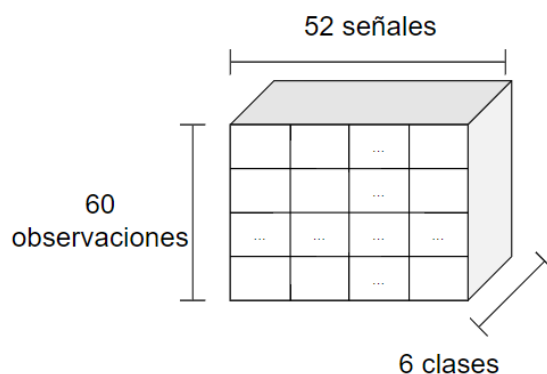


Figura 4.28: Cuarta versión del conjunto de datos de E2C

4.3.3 Configuración de la Red Neuronal Convolutiva de 1 Dimensión

Para la configuración de la arquitectura de la 1DCNN se utilizaron un total de 8 capas las cuales se presentan en la figura 4.29. La creación del modelo de 1DCNN requiere ingresar pesos iniciales para comenzar a entrenar a partir de estos. Los pesos seleccionados para nuestra red fueron aleatorios, realizando múltiples iteraciones de la red para evaluar con distintas configuraciones de pesos, hasta encontrar una configuración óptima entre el resto de evaluadas. En la generación de el modelo para la 1DCNN se utilizó configuraron los parámetros de entrenamiento de optimización y pérdida con los algoritmos Adam y Cross Entropy Loss, respectivamente.

4.3.4 Construcción del segundo experimento - configuración C (E2C)

Preprocesamiento de datos

Una vez dividido el conjunto de datos, como parte del preprocesamiento de la señal, se aplicaron filtros únicamente para reducir el ruido en la señal y eliminar valores atípicos que pueda presentar la misma, es decir, se aplicaron filtros de mediana y Hampel al igual que en la creación del primer conjunto.

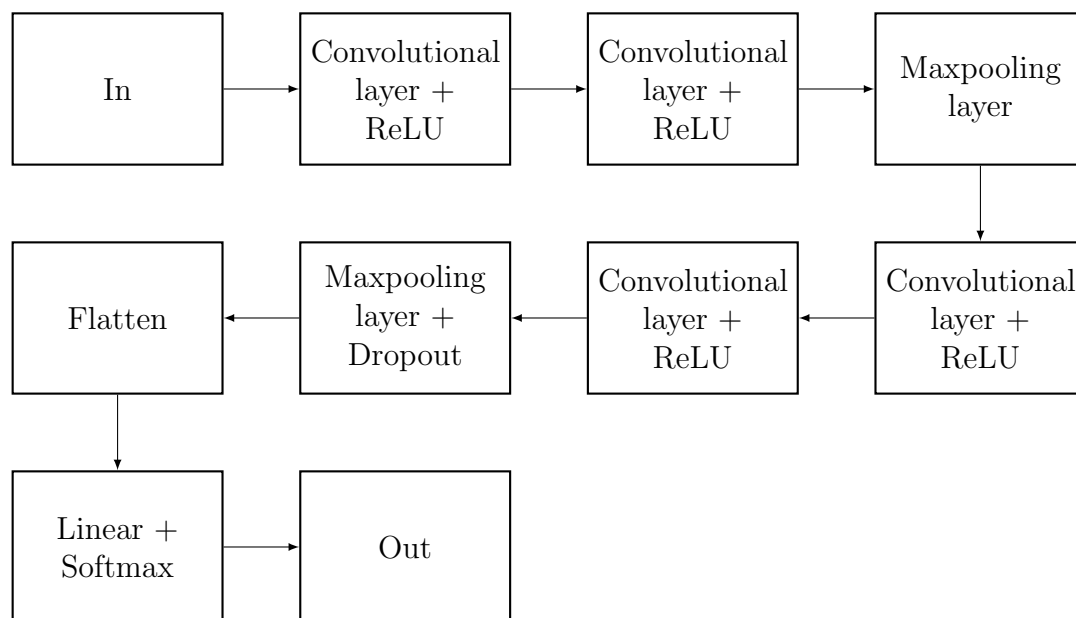


Figura 4.29: Arquitectura de la Red Neuronal Convolutiva

En la figura 4.30 se observa el comportamiento que presentan las señales más relevantes con 0 personas para este conjunto antes y después de aplicar los filtros de Hampel y mediana. El orden que se muestra son la señal en su estado original, la señal después de aplicar filtro de Hampel y la señal con filtro de mediana. Esto se aplica para 0 personas hasta 5 personas dentro de la aplicación. El resto de señales para 1 hasta 5 personas se encuentra en la sección 7, Apéndice A.

4.3.5 Construcción del segundo experimento - configuración D (E2D)

Los resultados obtenidos con las señales para en E2C no fueron los esperados debido a que en la creación del conjunto de datos anterior se identificó un filtro pasa bajas, lo que afectó de manera directa los resultados. Debido a esa configuración se propuso una revisión sobre cómo se obtuvo el conjunto de datos a partir de la señal de CSI obtenida.

Los ESP32 realizan la tarea de recolección del CSI y la herramienta que se utilizó arroja el valor del CSI que consiste en datos binarios, al cual es necesario aplicarle ciertos algoritmos para obtener su valor en amplitud y/o fase, en este caso sólo es relevante la amplitud del CSI.

Para obtener la información necesaria para el conteo de datos es necesario convertir la información de binario a entero. Durante este proceso de conversión se encontraba activo un filtro pasa bajas, el cual es importante para otras actividades de reconocimiento con señales de CSI, sin embargo, para el conteo de personas elimina información importante. Por esta razón se desactivó el filtro para realizar una segunda configuración con este conjunto de datos con las mismas señales recopiladas dentro de la jaula.

Una vez obtenido los valores de amplitud del CSI sólo se realizó la construcción del conjunto de datos al igual que en el configuración C, es decir, se obtuvo un

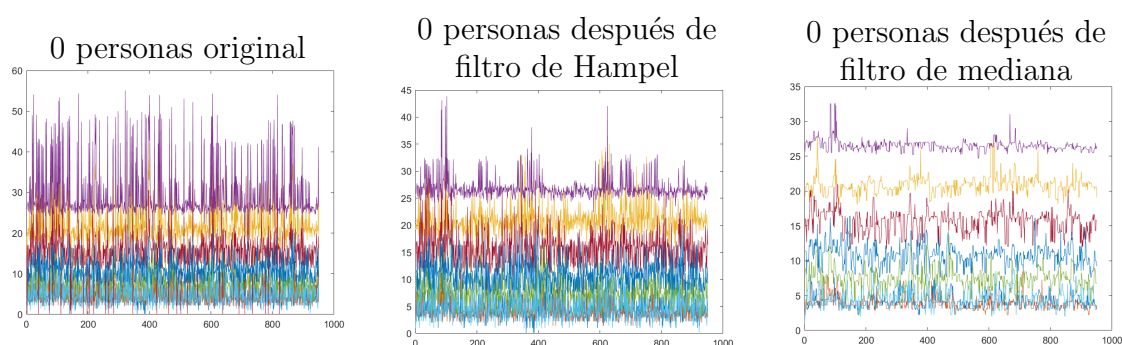


Figura 4.30: Señales principales para 0 personas antes y después de aplicar los filtros de Hampel y mediana en el segundo conjunto de datos

conjunto de 52 señales x 60 observaciones x 6 clases de la figura 4.28.

Preprocesamiento de datos

Como parte del preprocesamiento sólo se eliminaron valores atípicos y ruido en la señal, evitando realizar preprocesamiento excesivo para aumentar el rendimiento de una futura aplicación de conteo en tiempo real.

Reducción de dimensionalidad

Debido al aumento en la tasa de reconocimiento presentado en este E2D con respecto a E2C, se aplicó un algoritmo para reducir las dimensiones del conjunto de datos, más específicamente reduciendo el total de señales. El algoritmo utilizado está basado en las estrategias Forward Selection y Backward Selection, lo que permite buscar dentro de las 52 subportadoras, una combinación óptima para generar la mayor tasa de reconocimiento utilizando el menor número de señales.

Esto permitió reducir el conjunto de datos hasta tener 8 señales con la información relevante. En la figura 4.31 se muestra la versión final del conjunto de datos.

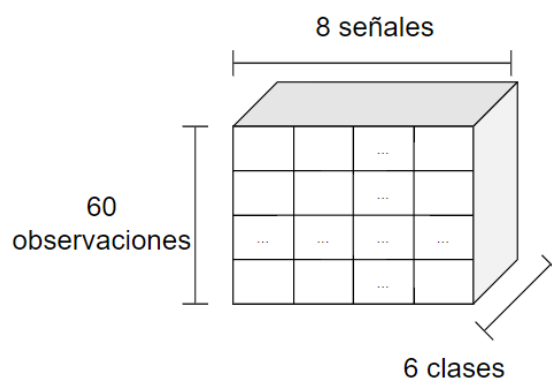


Figura 4.31: Versión final del conjunto de datos de E2D

5. Resultados

Para generar veracidad en los resultados de utilizar los distintos algoritmos de clasificación se aplicó una validación con K-Fold cross validation. Los porcentajes que se muestran en esta sección fueron generados con dicha técnica, con excepción de la configuración A del primer experimento, donde los resultados eran tan bajos que no fue relevante aplicar la validación.

5.1 Primer experimento

En este primer experimento se utilizaron algoritmos de Machine Learning y Deep Learning para evaluar el conjunto de datos. Con los resultados del experimento E1A fue posible encontrar áreas de oportunidad, los cuales, una vez corregidos, permitieron mejorar considerablemente los resultados de clasificación para el experimento E1B.

5.1.1 Configuración A

Entre los algoritmos utilizados se encuentran Naive Bayes, LDC y QDC. Dichos algoritmos tienen una implementación muy sencilla, debido a que no necesitan configuración de parámetros. Por esta razón se utilizaron como un primer acercamiento para trabajar con algoritmos de clasificación.

Para evaluar el conjunto de datos con estos algoritmos se realizaron 100 ejecuciones para cada uno de los algoritmos, donde las tasas de reconocimiento fluctuaban con valores entre 16% y 55%, mostrando resultados no confiables debido a la diferencia de los valores mínimo y máximo, además que el porcentaje se mantiene debajo del 60%.

Comportamiento de los datos

Al tener un bajo rendimiento y una alta variación en los resultados de los algoritmos Naive Bayes, LDC y QDC, se optó por analizar el comportamiento del conjunto de datos utilizando la técnica de reducción de dimensionalidad T-sne. En la figura 5.1 se observa el comportamiento del conjunto.

Si bien, hay algunas clases que parecen agruparse, como es el caso de la clase 1, no es posible agrupar de manera sencilla todas las clases. Debido a esto se realizaron correcciones en la construcción del conjunto para optimizar el comportamiento.

El primer cambio fue aumentar el número de subportadoras. Hasta el momento sólo se utilizaban las 30 subportadoras correspondientes a la comunicación entre 1 par de antenas, sin embargo, se cuenta con 3 pares, por lo tanto, el total de subportadoras se triplicó de 30 hasta 90.

El segundo cambio fue aumentar las características extraídas. En esta configuración A sólo se extrajeron media y mediana, sin embargo, para crear un nuevo conjunto se contemplaron, además de las ya mencionadas, las medidas estadísticas de varianza, desviación estándar, curtosis, momento central de segundo, tercero y cuarto orden.

El tercer cambio se realizó para aumentar la cantidad de observaciones. Con esta configuración se utilizaron 10 observaciones de cada clase (desde la clase 1 hasta la clase 9), con un total de 90 observaciones en el conjunto de datos. Para aumentar el total de datos se dividió cada observación utilizando ventanas de tiempo de $1/3$ del tamaño original, es decir, de 1 observación se aumentó a 3 observaciones, por lo tanto, de las 10 observaciones por clase aumentó hasta 30

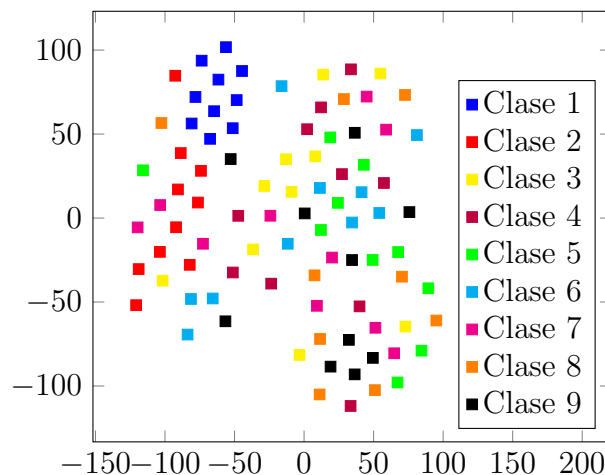


Figura 5.1: Comportamiento del sistema 1 con la técnica T-sne

observaciones por clase, y de 90 observaciones en el conjunto se aumentó hasta 270 observaciones.

Debido a estos cambios, el comportamiento del conjunto de datos se vio alterado. En la figura 5.2 se muestra el resultado de las correcciones en el conjunto de datos.

Con esta corrección en el sistema es posible ver que en el centro del conjunto se tiene un mayor número de etiquetas de clases 1, 2 y 3, mientras que en el exterior se tienen las demás clases. Por esta razón, es posible segmentar el conjunto en 2 clases para medir el nivel de ocupación de la habitación. En la figura 5.3 se puede ver el comportamiento del conjunto aplicando la reducción de 9 clases hasta 2 clases.

Con estas 2 clases, se puede apreciar con más claridad el patrón en el comportamiento del conjunto de datos, donde al centro se encuentran aquellos clase 1, y en los alrededores se encuentran los clase 2. Todos estos cambios permitieron analizar el nuevo conjunto desde el punto de vista de estimar el nivel de ocupación de la habitación, es decir, si se encuentra vacía o llena.

5.1.2 Configuración B

Para el experimento E1B se aplicaron los siguientes algoritmos de Machine Learning: Naive Bayes, KNN, LDC, QDC y Support Vector Machine, mientras que por el lado de los algoritmos de Deep Learning se aplicó una Red Neuronal Convolutiva de 1 Dimensión.

Se aplicaron dos pruebas, una para clasificar el nivel de ocupación de la habitación (2 clases) y la segunda prueba para contar el total de personas dentro de la

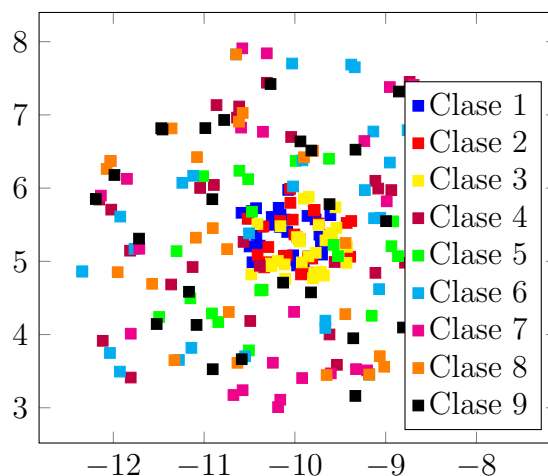


Figura 5.2: Comportamiento del conjunto de datos corregido

5.1 Primer experimento

misma (9 clases). En la prueba de dos clases se utilizaron los algoritmos mostrados en la tabla 5.1, donde se pueden ver desglosadas las tasas de reconocimiento y las matrices de confusión obtenidas al utilizar los distintos algoritmos de Machine Learning.

Utilizando los algoritmos tradicionales de Machine Learning se obtuvo un valor

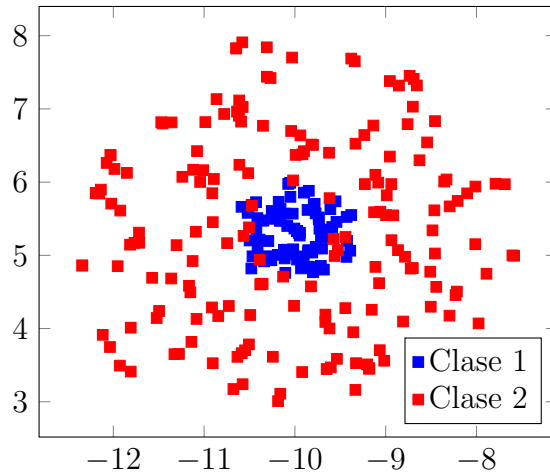


Figura 5.3: Comportamiento del conjunto de datos con sólo 2 clases

Algoritmo	Reconocimiento	Matriz de confusión				
Naive Bayes	77.78%	Predichos Verdaderos <table border="1"> <tr> <td>14</td> <td>4</td> </tr> <tr> <td>4</td> <td>14</td> </tr> </table>	14	4	4	14
14	4					
4	14					
KNN	86.11%	Predichos Verdaderos <table border="1"> <tr> <td>17</td> <td>1</td> </tr> <tr> <td>4</td> <td>14</td> </tr> </table>	17	1	4	14
17	1					
4	14					
LDC	80.56%	Predichos Verdaderos <table border="1"> <tr> <td>16</td> <td>2</td> </tr> <tr> <td>5</td> <td>13</td> </tr> </table>	16	2	5	13
16	2					
5	13					
QDC	77.78%	Predichos Verdaderos <table border="1"> <tr> <td>13</td> <td>5</td> </tr> <tr> <td>3</td> <td>15</td> </tr> </table>	13	5	3	15
13	5					
3	15					

Tabla 5.1: Métricas de algoritmos de Machine Learning con el conjunto de 2 clases

5.1 Primer experimento

máximo de 86.11% de reconocimiento estimando la capacidad de la habitación.

Para poder contar el total de personas desde 0 hasta 8, utilizando más clases se decidió utilizar un algoritmo más potente de Machine Learning, como lo es Support Vector Machine. Este algoritmo está diseñado para trabajar utilizando conjuntos de datos que diferencian entre 2 clases, debido a que el conjunto de datos aplica tanto 2 clases, como un total de 9, es necesario aplicar algunas técnicas para clasificar un mayor número de etiquetas.

Utilizando sólo 2 etiquetas, las Support Vector Machine fueron capaces de obtener una tasa de reconocimiento de hasta 91.67%. En la figura 5.2 se puede ver la tasa de reconocimiento obtenida al utilizar los distintos kernel de configuración de las SVM.

Para clasificar más de 2 clases utilizando las SVM es necesario aplicar las técnicas de *one vs. all* o *one vs. one*. En la tabla 5.3 se muestra la tasa de reconocimiento aplicando la técnica *one vs. all* clasificando cada observación del conjunto y agrupando los porcentajes por clases. Al final se calcula el promedio de todas las clases para obtener la tasa de reconocimiento aplicando dicha técnica.

Por otro lado, en la tabla 5.4 se muestra la tasa de reconocimiento agrupada por clases para cada una de las 9 que se encuentran dentro del conjunto de datos, y al final se muestra el promedio de la técnica *one vs. one*.

El algoritmo para clasificar con las SVM, tanto para 2 clases, como con 9 clases

Lineal	Polinomial	RBF
77.78%	88.89%	91.67%

Tabla 5.2: Resultados de SVM usando el conjunto de 2 clases

Número de personas	Lineal	Polinomial	RBF
0	100%	100%	100%
1	100%	98.14%	98.14%
2	88.89%	88.89%	88.89%
3	88.89%	88.89%	92.59%
4	88.89%	88.89%	90.74%
5	88.89%	92.59%	90.74%
6	90.74%	98.14%	94.44%
7	88.89%	88.89%	90.74%
8	88.89%	96.29%	94.44%
Tasa de reconocimiento	91.56%	93.41%	93.41%

Tabla 5.3: Resultados de SVM usando el conjunto de 9 clases y la estrategia *one vs. all*

5.1 Primer experimento

Número de personas	Lineal	Polinomial	RBF
0	100%	89.58%	100%
1	100%	95.83%	100%
2	79.17%	78.13%	84.87%
3	77.08%	79.17%	85.41%
4	82.29%	81.25%	84.38%
5	80.21%	86.46%	86.46%
6	82.29%	80.21%	87.50%
7	80.21%	75.00%	83.33%
8	68.75%	71.88%	73.96%
Tasa de reconocimiento	83.33%	81.94%	87.27%

Tabla 5.4: Resultado de SVM usando el conjunto de 9 clases y la estrategia *one vs. one*

aplicando ambas estrategias arrojó mejores resultados utilizando la configuración del Kernel Radial Basis Function.

Trabajando con el conjunto de datos de 2 clases, se pudo obtener el mejor rendimiento de la SVM, ya que se encuentra diseñada para 2 clases, y cada observación se compara 1 vez con todo el conjunto. Sin embargo, al realizar un análisis para el conjunto con más de 2 clases se hacen un distinto número de comparaciones.

Trabajando con 2 clases, se tiene un conjunto de datos de 90 observaciones por clase x 30 características extraídas x 2 clases. Para evaluar el conjunto se utiliza una partición 80-20, donde el 80% de los datos se usan para training y el 20% restante se utiliza para test. Esta separación hace que se cuente un total de 72 observaciones por clase en el conjunto de training y 18 observaciones para test. Para cada una de las observaciones en test se clasificó en base al conjunto de training de 72 observaciones, analizando si la observación pertenece a la clase 1 o a la clase 2 todo en 1 única iteración.

En la estrategia *one vs. all* se compara una observación con todo el conjunto, para identificar si dicho valor pertenece o no a una clase en específico. Dicha acción la verifica comparando una observación, con todo el conjunto dividido en dos clases, una que indica que la observación pertenece a la clase n y otra que indica que no pertenece a esta clase n. Como se tiene un total de 9 clases, entonces es necesario que se verifique para cada clase, es decir, se verifica que la observación pertenezca o no a la clase 1, después se verifica que pertenezca o no a la clase 2, después que pertenezca o no la clase 3, realizando esta acción para cada una de las 9 clases del conjunto. Al finalizar, dependiendo de la clasificación que le dio a cada elemento se obtiene la tasa de reconocimiento para cada clase en particular,

con lo cual se obtiene el promedio de todas las clases, para generar la tasa de reconocimiento de Support Vector Machine con la estrategia *one vs. all*.

Esta tarea implica realizar un mayor número de comparaciones. El conjunto que se usó para las 9 clases es aquel que tiene las dimensiones de 30 observaciones por clase, por 30 características, por 9 clases, y se usó una partición 80% training y 20% test. Como resultado, se analizan 6 observaciones de cada una de las 9 clases en test, con respecto a 24 observaciones de cada una de las 9 clases en train. Con esta estrategia se clasifican las observaciones en test para saber si pertenecen a la clase 1 o no, después para saber si pertenece a la clase 2 o no, después para saber si pertenece a la clase 3 o no, realizando esta acción una vez para cada clase, es decir, realizando un total de 9 iteraciones, aumentando el tiempo de respuesta de este algoritmo al menos unas 9 veces mayor al de 2 clases.

La segunda estrategia utilizada es *one vs. one*, donde se compara una observación con respecto a una clase en específico, para ver si dicha observación pertenece o no a cada una de las clases. Esta técnica se basa en dividir el conjunto de datos en un total de n subconjuntos equivalentes, donde n es el total de clases dentro del conjunto, es decir, como se trabajó con 9 clases se obtuvieron 9 subconjuntos, donde cada uno representa a 1 clase en particular. La manera de evaluar este conjunto fue utilizando una observación de una clase específica, por ejemplo la clase 1, y clasificar si esa observación pertenece a la clase 1 o a la clase 2, después se verifica si pertenece a la clase 1 o la clase 3, después se compara con clase 1 y clase 4, y de esta manera se compara la clase a la que pertenece, con todas las otras clases, obteniendo una tasa de reconocimiento para cada escenario donde es posible encontrar la clase correcta.

En este conjunto se usó la misma partición, un 80% de las observaciones para training y el 20% restante para test. Para cada una de las observaciones de cada clase en test se realiza la clasificación con respecto a cada una de las otras clases, es decir, se aplica el algoritmo 8 veces para cada una de las clases, teniendo 9 clases en total.

Para no aplicar comparaciones extras, lo que se hizo con el algoritmo es obtener la tasa de reconocimiento de todos los subconjuntos de la clase comenzando con la clase 1, y después en cada clase siguiente se realiza 1 comparación menos que en la anterior, es decir, se realizan las 8 comparaciones de clase 1, desde clase 2 hasta clase 9, al trabajar con clase 2 se hacen 7 comparaciones, desde clase 3 hasta 9, esto debido a que SVM trabaja dividiendo los subconjuntos en hiperplanos, donde es lo mismo comparar clase 1 y 2, que comparar clase 2 y 1. En la tabla 5.5 se muestra a detalle cuántas iteraciones de SVM son requeridas para obtener la tasa de reconocimiento de cada clase.

A partir la clase 2 en adelante se reutilizan las comparaciones de la clase anterior para ahorrar recurso computacional, por ejemplo, con la etiqueta de clase 2 se hace

Etiqueta de clase	Iteraciones por clase
1	8
2	7
3	6
4	5
5	4
6	3
7	2
8	1
9	0
Total de iteraciones:	36

Tabla 5.5: Total de iteraciones por clase para aplicar SVM con estrategia one vs. one

1 iteración menos, ya que en la anterior se tiene lista la comparación de clase 1 con la misma clase 2.

Si bien los resultados que arrojó utilizar las SVM están alrededor del 90 %, el esfuerzo computacional que se requiere es elevado, ya que es necesario realizar varias iteraciones del mismo algoritmo, lo cual puede reducir las posibilidades de crear una aplicación de "tiempo real".

Para combatir este problema del tiempo de ejecución de las SVM, se aplicó un algoritmo más enfocado a la velocidad, las Redes Neuronales Convolucionales. Una Red Neuronal Convolutiva es un algoritmo de Deep Learning el cual, habitualmente, se enfoca en trabajar con imágenes, realizando convoluciones en 2 dimensiones (debido a las dimensiones de una imagen de alto y ancho), permitiendo así extraer características de la imagen para poder clasificar en alguna etiqueta. Sin embargo, de acuerdo a S. Kiranyaz [40], es posible utilizar Redes Neuronales Convolucionales que trabajen de manera más rápida y permita crear aplicaciones más ligeras, utilizando Redes Neuronales Convolucionales de 1 Dimensión (1DCNN).

Las redes de tipo 1DCNN se caracterizan por realizar convoluciones en 1 sola dimensión en lugar de 2 dimensiones, ya que no utilizan imágenes como entrada para la red como se hace en las CNN, sino que 1DCNN usa señales para alimentar la red. Las imágenes poseen 2 dimensiones (alto y ancho) y para cada celda dentro de estas dimensiones se posee un valor que representa el color de la imagen; por otro lado, para las señales únicamente se posee el tiempo o frecuencia de la señal y para cada celda dentro de la misma se obtiene un valor que representa la amplitud. Entonces, la 1DCNN permite realizar las convoluciones únicamente para el tiempo de la señal, permitiendo así extraer las características de la misma sin la utilización de procesamiento para una dimensión extra.

5.1 Primer experimento

Para este experimento se realizó la evaluación de la 1DCNN tanto para estimar el nivel de ocupación de la habitación como para contar el total de personas dentro de la misma. El conjunto de datos para evaluar el experimento en ambos casos es aquel al que ya se le aplicó la etapa de extracción de características utilizando las 8 medidas estadísticas. Debido a esto, la entrada para la 1DCNN no son señales, sino son las 8 características extraídas.

La primer clasificación utilizando la 1DCNN es etiquetar el nivel de ocupación que representa la habitación, con el cual se obtuvo una tasa de reconocimiento del 85.18%. En la tabla 5.6 se observa la matriz de confusión.

La segunda clasificación con la 1DCNN es contar a las personas utilizando el conjunto de datos con la habitación vacía hasta tener 8 personas dentro de la misma. Para esta actividad de conteo se obtuvo una tasa de reconocimiento del 48.14% contando personas en la habitación, desde 0 hasta 8 personas. En la tabla 5.7 se muestra la matriz de confusión utilizando 1DCNN con 9 personas.

Debido al comportamiento que presenta la matriz de confusión, donde en la habitación vacía tiene 0 errores, tanto Falsos Positivos como Falsos Negativos, y conforme aumenta el número de la etiqueta de clase empeora el comportamiento del algoritmo, se realizó la hipótesis que este comportamiento puede mejorar disminuyendo el número de etiquetas dentro del conjunto de datos.

Para corroborar esta hipótesis se realizó una segunda prueba de conteo de personas utilizando el algoritmo 1DCNN aplicando algunas variantes a este conjunto de datos. Ya que la hipótesis plantea un mejor resultado disminuyendo las etiquetas de clase, lo que se hizo fue eliminar aquellas observaciones que pertenecen a la clase 8 y clase 9, dejando en el conjunto de datos las observaciones que se encuentran desde la clase 1 hasta la clase 7.

		Predicted	
		1	2
True	Class	1	2
		12	6
	2	2	34

Tabla 5.6: Matriz de confusión para 1DCNN usando 2 clases

		Predicted								
		1	2	3	4	5	6	7	8	9
True	Class	1	2	3	4	5	6	7	8	9
	1	6	0	0	0	0	0	0	0	0
	2	0	5	0	0	0	1	0	0	0
	3	0	0	2	1	0	2	1	0	0
	4	0	0	3	1	0	0	2	0	0
	5	0	0	2	0	3	0	1	0	0
	6	0	0	0	1	0	2	2	1	0
	7	0	0	0	1	0	0	4	1	0
	8	0	0	1	1	1	0	0	3	0
9	0	0	3	0	0	2	1	0	0	

Tabla 5.7: Matriz de confusión para la 1DCNN contando 9 clases

Este subconjunto cuenta con dimensiones similares, ya que son 8 características extraídas por 30 subportadoras seleccionadas por 7 clases. Para este conjunto, en comparación con la evaluación anterior de 9 clases, se logró aumentar la tasa de reconocimiento hasta un 61.90%. En la tabla 5.8 se muestra la matriz de confusión de clasificar con 7 etiquetas de clase.

Debido al aumento de la precisión en la clasificación eliminando las últimas 2 clases, se pudo confirmar la hipótesis para este conjunto de datos, donde, utilizando un menor número de etiquetas de clase es posible obtener mejores resultados de clasificación usando las 1DCNN.

Con esto en mente, se optó por realizar un segundo experimento de conteo de personas, utilizando no 8 personas para el conteo esta vez, sino que se redujo a 5 personas, para ver el comportamiento de un nuevo conjunto de datos.

5.2 Segundo experimento

Los experimentos E1A y E1B se realizaron en una habitación dentro del instituto de ingeniería para simular un ambiente más natural, sin embargo, para cubrir la necesidad de realizar experimentos en un ambiente controlado, se decidió por construir una jaula de Faraday para eliminar señales que puedan provocar interferencia al momento de realizar la comunicación WiFi.

5.2.1 Configuración C

En los experimentos E1A y E1B, se realizó el conteo de personas en un conjunto de datos el cual contaba con características extraídas, sin embargo, debido a la naturaleza de las 1DCNN, para los experimentos E2C y E2D se decidió no aplicar una etapa de extracción de características, permitiendo así ingresar como entrada a la 1DCNN la señal generada a partir del CSI de la comunicación WiFi.

El conjunto de datos que se utilizó para esta clasificación cuenta con un total de 52 señales de 950 paquetes x 60 observaciones en cada clase x 6 clases.

		Predicted							
		1	2	3	4	5	6	7	
True	Class	1	6	0	0	0	0	0	0
	2	0	5	0	0	0	1	0	
	3	0	0	4	1	0	0	1	
	4	0	0	2	0	1	1	2	
	5	0	0	1	0	4	1	0	
	6	0	0	0	1	1	3	1	
	7	0	0	1	1	0	0	4	

Tabla 5.8: Matriz de confusión para la 1DCNN contando 7 clases

Con este conjunto de datos se aplicó el algoritmo 1DCNN, sin embargo, los resultados que se obtenían estaban alrededor del 50% al 55%. Debido a esto fue necesario dar pasos hacia atrás en la elaboración del conjunto de datos.

Al momento de obtener los datos del CSI para generar una señal se obtiene la información que corresponde a cada subportadora como un número complejo. A dicho número es necesario aplicarle operaciones para poder obtener los valores de amplitud y fase. En este caso, únicamente se obtuvo el valor de amplitud, sin embargo, durante el momento de la conversión de número complejo a amplitud, se encontraba activado un algoritmo de filtro pasa bajas, el cual se usa para limpiar la señal y optimizar los resultados del CSI para actividades humanas como caminar, correr o caer. Debido a que la actividad que se trabajó es conteo de personas, el filtro pasa bajas elimina información con la que se puede realizar la clasificación.

Antes de encontrar el error de tener activado un filtro pasa bajas para una actividad que no lo necesita se dio un paso hacia atrás para buscar qué pasos se están realizando de manera distinta del primer sistema con respecto a este segundo. Dentro de los cambios entre sistemas se pueden destacar los siguientes:

- El ambiente de recolección. En el primer sistema todos los datos que se recolectaban eran simplemente dentro de una habitación, mientras que en el segundo sistema se construyó una jaula de Faraday para aislar los campos electromagnéticos y tener un ambiente más controlado.
- Los dispositivos de recolección. En el sistema anterior se utilizaron computadoras Intel NUC Kit, mientras que en este segundo sistema se utilizan dispositivos ESP32, los cuales son mucho más accesibles y con menores capacidades de procesamiento.
- El número de antenas en la recolección. Con las Intel NUC Kit se tenía un total de 3 pares de antenas entre transmisor y receptor, mientras que con los ESP32 sólo se tiene 1 par de antenas.
- La creación del conjunto de datos. El conjunto de datos utilizado con el primer sistema es un conjunto al que se le aplicó una capa de extracción de características, por otro lado, el segundo sistema sólo recibe como entrada las señales generadas a partir del CSI con los algoritmos de preprocesamiento de eliminación de ruido y valores atípicos sin tener esta etapa de extracción.

Al tener todos estos cambios entre un sistema y otro, no es posible definir si un cambio en específico contribuye ya sea positiva o negativamente para la tarea de clasificación.

5.2.2 Configuración D

Debido a la importancia de los datos perdidos por el filtro, se realizó una configuración distinta con el filtro desactivado. En este experimento E2D los resultados de conteo de persona aumentaron a tasas de reconocimiento que alternaba entre 65% y 70%.

Las subportadoras utilizadas para el conjunto de datos son las 52 que se obtienen a partir de la comunicación de los ESP32, sin embargo, no todas las subportadoras obtienen información relevante, o información única, ya que es posible que la información de varias subportadoras sea redundante y reducir la información de entrada aumente la velocidad y eficiencia del sistema.

Para optimizar las subportadoras que se utilizan en el sistema, una vez obtenido la mejora de apagar el filtro pasa bajas, se aplicó un algoritmo de selección de subportadoras basado en las estrategias Forward Selection y Backward Selection. Este algoritmo se dejó en búsqueda, alrededor de 12 horas, lo cual permitió reducir las subportadoras de 52, a únicamente 8 subportadoras.

A partir de la reducción de las subportadoras, las ejecuciones de 1DCNN rondaban entre 70% y 80%. Los modelos generados son con pesos aleatorios, así que para optimizar el modelo de 1DCNN se realizaron las iteraciones de 1DCNN durante 26 horas para buscar la configuración óptima de pesos en la red.

Al finalizar se encontró un modelo optimizado, el cual se exportó para poder cargar en cualquier otro equipo con el mismo conjunto de datos. Este modelo arrojó una tasa de reconocimiento del 83.33% contando con 6 clases, desde la habitación vacía hasta tener a 5 personas dentro de la misma. En la figura 5.9 se muestra la matriz de confusión del modelo optimizado.

		Predicted					
		1	2	3	4	5	6
True	Class						
	1	12	0	0	0	0	0
	2	0	12	0	0	0	0
	3	0	0	11	0	2	0
	4	0	0	1	7	0	0
	5	0	0	0	1	9	3
6	0	0	0	4	1	9	

Tabla 5.9: Matriz de confusión para la 1DCNN con el modelo optimizado

6. Conclusiones y trabajos futuros

6.1 Conclusiones

El CSI permite reconocer una variedad de actividades humanas como puede ser caminar, correr, monitorear la respiración, o para este caso, contar personas que se encuentren dentro de la difusión de la señal WiFi. En este trabajo de investigación se realizaron pruebas de desempeño de varios algoritmos tanto de Machine Learning como de Deep Learning. Al inicio se tenía en mente comenzar con los algoritmos de Machine Learning para tener un acercamiento con el CSI y las etapas necesarias para la minería de datos, como son el preprocesamiento de datos, la extracción de características y finalmente la clasificación a través de los distintos algoritmos. Sin embargo, durante el proceso de investigación se descubrió un tipo de algoritmos de Deep Learning que permiten trabajar con Redes Neuronales de una manera más veloz y ligero. Revisando el comportamiento presentado en este trabajo se puede inferir en las siguientes afirmaciones:

1. Analizando los resultados obtenidos por la 1DCNN, donde no presenta Falsos Positivos, ni Falsos negativos y presenta una tasa de reconocimiento del 100% trabajando únicamente con las clases que representan a la habitación vacía y la habitación con 1 persona, se puede afirmar que es posible crear una aplicación que trabaje en tiempo real, basada en dispositivos de bajo presupuesto como lo son los ESP32, que se encarguen de contar el total de personas que se encuentran dentro de una habitación pequeña con capacidad hasta 2 o 3 personas, puede funcionar de manera correcta, siempre y cuando se cumplan con las condiciones donde se realizó la toma de muestras de este trabajo.
2. En la construcción del segundo sistema existen algunas diferencias en la manera de obtención de datos y generación del conjunto. Debido a esta variedad de cambios que se aplicaron de un experimento a otro, hubo un

tiempo en el que el segundo sistema presentaba inconsistencias por tener activo un filtro pasa bajas, lo cual llevó a pensar en todos los cambios que hubo de un sistema a otro, y al no realizar los cambios de uno en uno para conocer si el cambio generado era positivo o negativo no se pudo rastrear si era ocasionado por la implementación de la jaula, por el cambio de los dispositivos de recolección, por el total de antenas, o por los algoritmos utilizados. Por esta razón es importante aplicar cambios pero de poco a poco en los sistemas, permitiendo así comparar los resultados para saber si algún cambio es positivo o negativo.

3. Con la 1DCNN fue posible incrementar la tasa de reconocimiento que presenta el algoritmo al disminuir el número de clases con las que se está trabajando, ya que al disminuir de 9 clases a 7 clases con el mismo conjunto de datos pudo aumentar la tasa de reconocimiento, mientras que en el segundo sistema, teniendo en cuenta todos los otros cambios que se aplicaron para la construcción del conjunto, disminuir la clases a 6 ayudó nuevamente para incrementar la tasa de reconocimiento.

6.2 Trabajos futuros

Si bien los resultados obtenidos con la elaboración del experimento E2D permitieron aumentar la exactitud de la clasificación el sistema presentado tiene áreas de mejora, las cuales pueden ser atacadas para buscar un mejor desempeño al actual obtenido.

1. Realizar pruebas con un distinto número de personas. En el primer sistema se aplicó una disminución del total de personas de 9 a 7, lo que permitió aumentar la tasa. Una línea de acción que se puede tomar es comenzar de menos a más, es decir, comenzar a clasificar con un conjunto de datos que cuente con un número bajo de personas, por decir unas 3 o 4 para comenzar a probar distintas configuraciones para la recolección y creación de un conjunto de datos.
2. Realizar pruebas de desempeño que permitan comparar el tiempo de ejecución de los distintos algoritmos utilizados para la clasificación, permitiendo así obtener la relación entre la exactitud de un algoritmo con su tiempo de ejecución.
3. Implementar más antenas para recolectar CSI que permitan obtener información más detallada del mismo debido a la diversidad espacial y corroborar si ayuda a incrementar las tasas de reconocimiento.

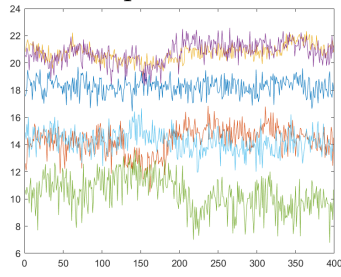
4. Diseñar un sistema que permita capturar el CSI, generar señales a partir de esta, aplicar algoritmos sencillos de preprocesamiento como reducción de ruido y valores atípicos, e ingresar la señal a un modelo previamente entrenado de Machine Learning o Deep Learning para obtener un valor del total de personas en tiempo real.
5. Jugar con las variables del entorno en el que se está realizando la medición. Un tercer conjunto de datos puede ser tomado en un ambiente que no tenga una Jaula de Faraday, pero sí se tome con los ESP32. En la tabla 6.1 se puede ver un ejemplo de las variables de entorno utilizadas en los 2 experimentos, así como algunas propuestas para modificar estas variables.
6. Otra prueba que se puede realizar es, con la Jaula de Faraday o con una habitación común, tener a personas dentro de la misma, y al mismo tiempo, tener a personas fuera de la habitación, para evaluar si el sistema presenta confusiones con el ruido que provocan estas personas que se encuentren en las cercanías de la habitación donde se realiza el conteo de personas.
7. Generar un conjunto de datos con personas en movimiento, los datos recolectados en este trabajo se realizaron con personas en una posición fija, sin embargo, en un ambiente real es normal que las personas puedan moverse libremente de un lugar a otro.

Número de experimento	Jaula de Faraday	Dispositivo de recolección	Pares de antenas	Máximo de personas
1	No	Intel NUC Kit	3	8
2	Sí	ESP32	1	5
Propuesta 3	Sí	Intel NUC Kit	3 o superior	4 o superior
Propuesta 4	No	ESP32	1 o superior	4 o superior

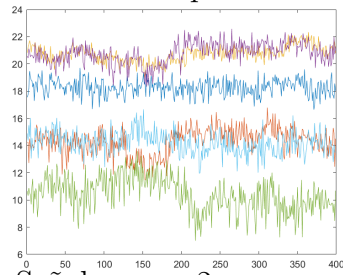
Tabla 6.1: Variables de entorno para la configuración de los sistemas

7. Apéndice A

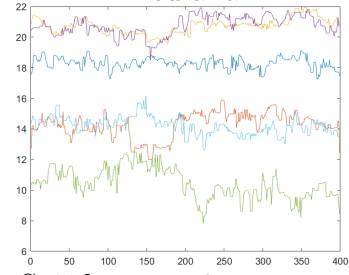
Señales originales para 1 persona



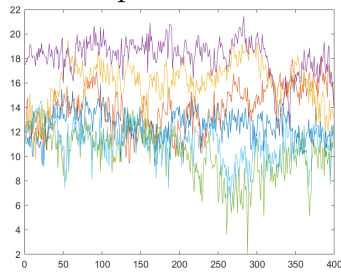
Señales para 1 persona después de filtro de Hampel



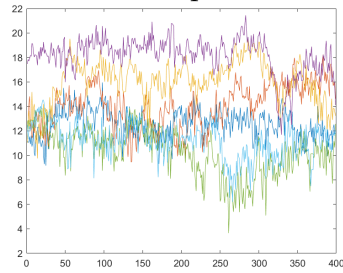
Señales para 1 persona después de filtro de mediana



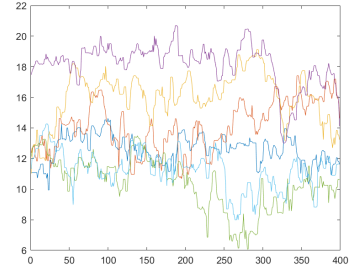
Señales originales para 2 personas



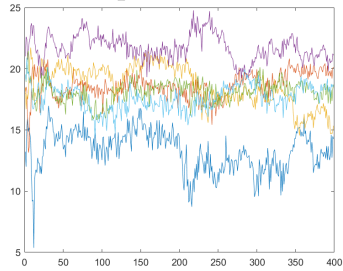
Señales para 2 personas después de filtro de Hampel



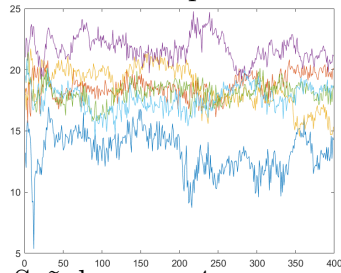
Señales para 2 personas después de filtro de mediana



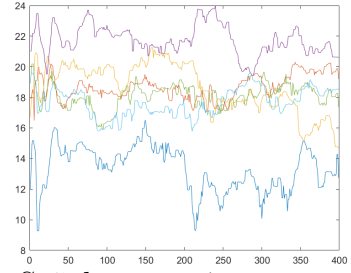
Señales originales para 3 personas



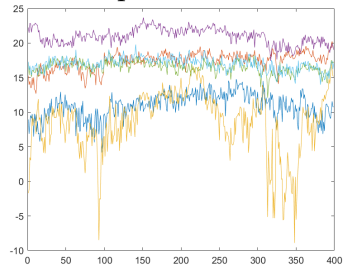
Señales para 3 personas después de filtro de Hampel



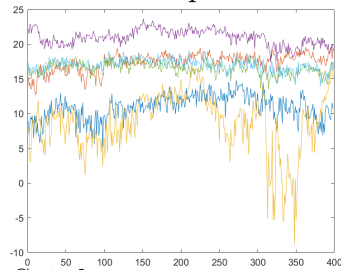
Señales para 3 personas después de filtro de mediana



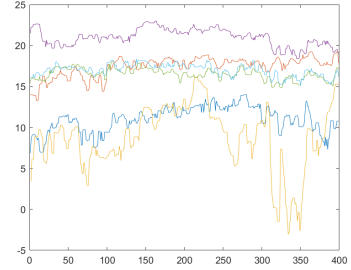
Señales originales para 4 personas



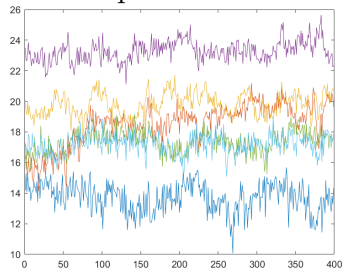
Señales para 4 personas después de filtro de Hampel



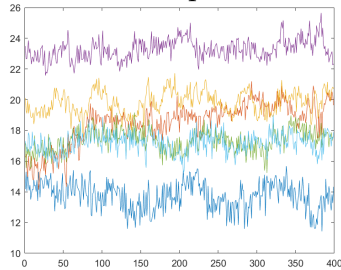
Señales para 4 personas después de filtro de mediana



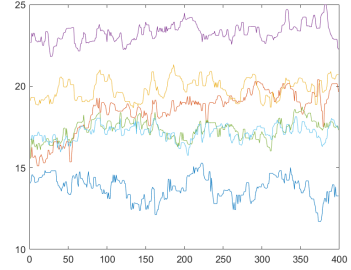
Señales originales para 5 personas



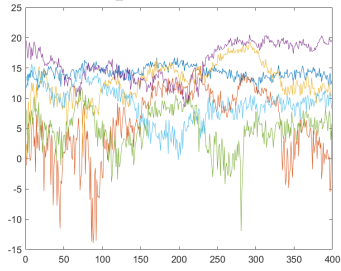
Señales para 5 personas después de filtro de Hampel



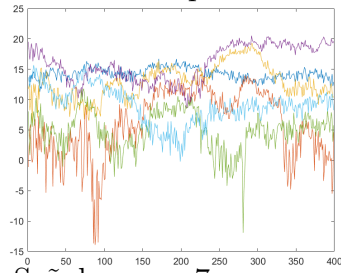
Señales para 5 personas después de filtro de mediana



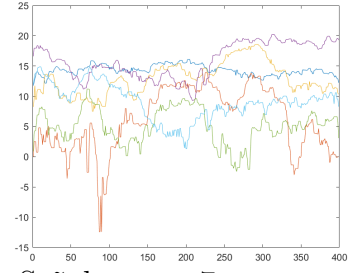
Señales originales para 6 personas



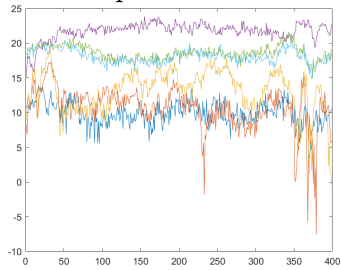
Señales para 6 personas después de filtro de Hampel



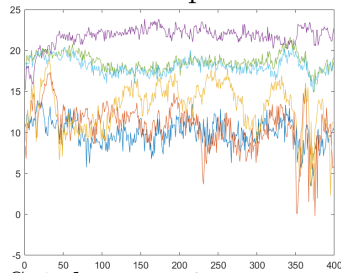
Señales para 6 personas después de filtro de mediana



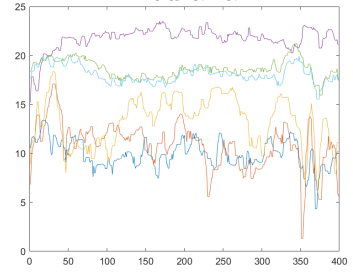
Señales originales para 7 personas



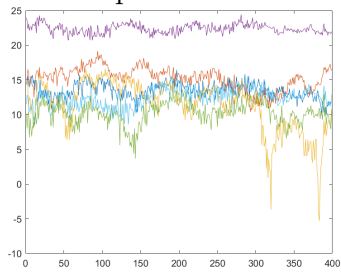
Señales para 7 personas después de filtro de Hampel



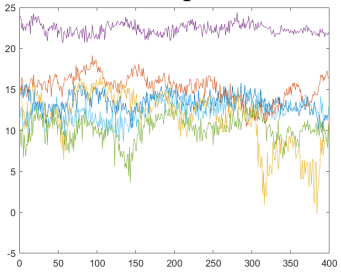
Señales para 7 personas después de filtro de mediana



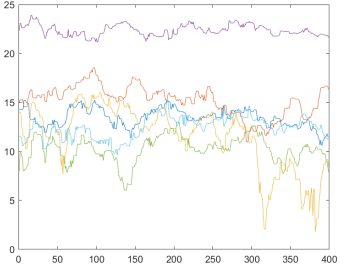
Señales originales para 8 personas



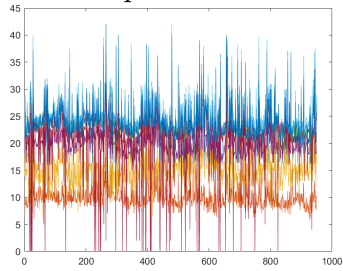
Señales para 8 personas después de filtro de Hampel



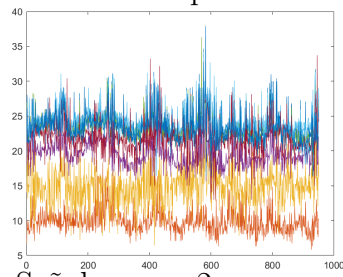
Señales para 8 personas después de filtro de mediana



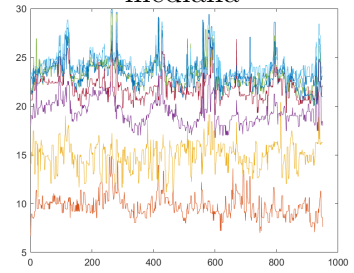
Señales originales para 1 persona



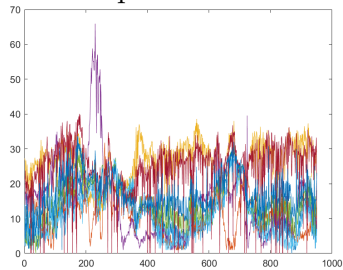
Señales para 1 persona después de filtro de Hampel



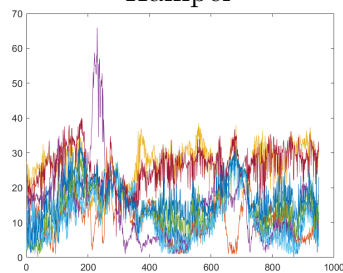
Señales para 1 persona después de filtro de mediana



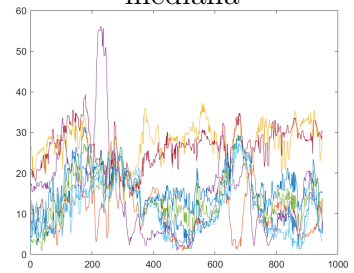
Señales originales para 2 personas



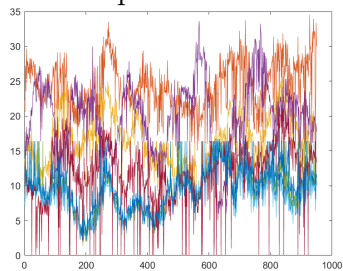
Señales para 2 personas después de filtro de Hampel



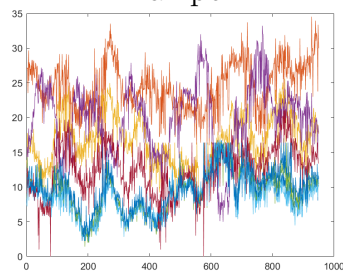
Señales para 2 personas después de filtro de mediana



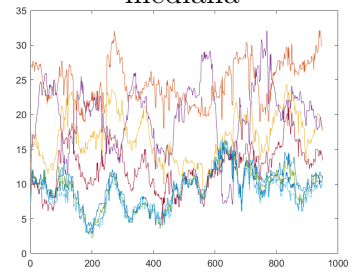
Señales originales para 3 personas



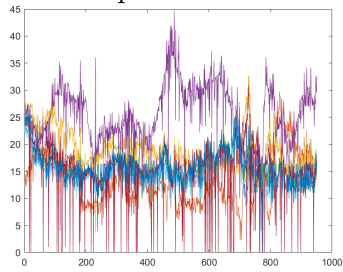
Señales para 3 personas después de filtro de Hampel



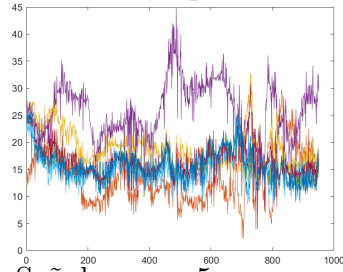
Señales para 3 personas después de filtro de mediana



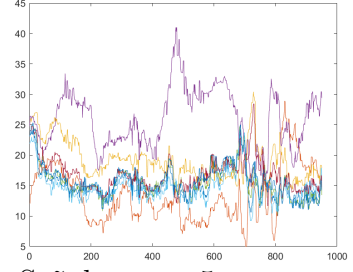
Señales originales para 4 personas



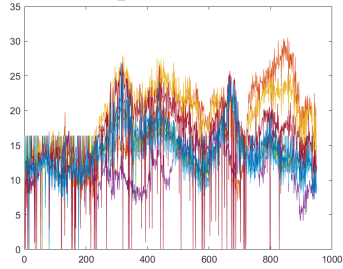
Señales para 4 personas después de filtro de Hampel



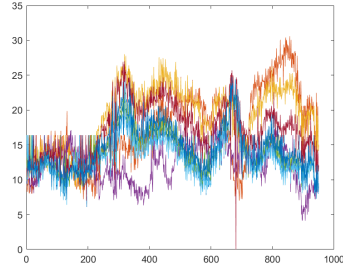
Señales para 4 personas después de filtro de mediana



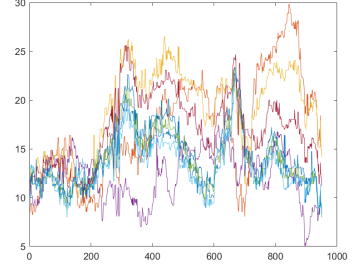
Señales originales para 5 personas



Señales para 5 personas después de filtro de Hampel



Señales para 5 personas después de filtro de mediana



8. Apéndice B

Productos generados a partir de la Tesis

Congresos

1. M. A. Torres-Cerda, F. F. Gonzalez-Navarro, J. Caro-Gutierrez and J. A. Armenta-Garcia, "CSI crowd-counting: An experimental study using Machine Learning and Deep Learning Algorithms," 2023 Mexican International Conference on Computer Science (ENC), Guanajuato, Guanajuato, Mexico, 2023, pp. 1-6, doi: 10.1109/ENC60556.2023.10508604.

Indautor

1. Registro de Obra ante INDAUTOR titulada "WI-FI CSI-PY: HERRAMIENTA DE RECEPCION Y TRANSOFMRACION DE INFORMACION DE ESTADO DEL CANAL WI-FI RECOLECTADA A PARTIR DE DISPOSITIVOS ESP-32".
2. Registro de Obra ante INDAUTOR titulada "ESP32 CSI WEB COLLECTING TOOL: PLATAFORMA WEB PARA LA CONFIGURACION DE UNA HERRAMIENTA DE RECOLECCIONDE WI-FI CSI".
3. Registro de Obra ante INDAUTOR titulada "ESP32 CSI WEB COLLECTING TOOL: HERRAMIENTA PARA LA RECOLECCION DE WI-FI CSI CON DISPOSITIVOS ESP32".
4. Registro de Obra ante INDAUTOR titulada "CIMARRON A LA CIMA". Juego que se presentó en diversos eventos de divulgación de la ciencia en Mexicali.

Divulgación de la ciencia

1. Segundo festival aeroespacial "Mexicali en las alturas".
2. Tercer festival aeroespacial "Mexicali en las alturas".

Constancias



CONAHCYT
CONSEJO NACIONAL DE HUMANIDADES
CIENCIAS Y TECNOLOGÍAS



El Centro de Investigación en Matemáticas A.C.,
el Instituto Nacional de Astrofísica, Óptica y Electrónica,
la Red Temática en Inteligencia Computacional Aplicada
y la Sociedad Mexicana de Ciencia de la Computación
otorgan la presente


CONSTANCIA

a:

Miguel A. Torres Cerda, Félix F. González Navarro,
Jesús Caro y Jesús Albany Armenta García

Por su artículo "CSI crowd-counting: An experimental study using Machine Learning and Deep Learning Algorithms" presentado en la edición conjunta del ENCUENTRO NACIONAL DE COMPUTACIÓN ENC2023 y el XI SEMINARIO Y ESCUELA NACIONAL DE APRENDIZAJE E INTELIGENCIA COMPUTACIONAL llevada a cabo del 11 al 15 de septiembre de 2023


Dra. Yasmín Hernández Pérez
Presidenta Comité Organizador
ENC 2023
CENIDET - SMCC - RedICA


Dra. Alicia Morales Reyes
Co-Presidenta Comité Organizador
XI SENAIC
INAOE - RedICA



CERTIFICADO

Registro Público del Derecho de Autor

Para los efectos de los artículos 13, 162, 163 fracción I, 164 fracción I, y demás relativos de la Ley Federal del Derecho de Autor, se hace constar que la **OBRA** cuyas especificaciones aparecen a continuación, ha quedado inscrita en el Registro Público del Derecho de Autor, con los siguientes datos:

AUTORES: ARMENTA GARCIA JESUS ALBANY
CARO GUTIERREZ JESUS
GONZALEZ NAVARRO FELIX FERNANDO
TORRES CERDA MIGUEL ANGEL

TÍTULO: WI-FI CSI-PY: HERRAMIENTA DE RECEPCION Y TRANSFORMACION DE INFORMACION DE ESTADO DEL CANAL WI-FI RECOLECTADA A PARTIR DE DISPOSITIVOS ESP-32

RAMA: PROGRAMAS DE COMPUTACION

TITULAR: UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA (CON FUNDAMENTO EN EL ARTICULO 83 DE LA LFDA)

Con fundamento en lo establecido por el artículo 168 de la Ley Federal del Derecho de Autor, las inscripciones en el registro establecen la presunción de ser ciertos los hechos y actos que en ellas consten, salvo prueba en contrario. Toda inscripción deja a salvo los derechos de terceros. Si surge controversia, los efectos de la inscripción quedarán suspendidos en tanto se pronuncie resolución firme por autoridad competente.

Con fundamento en los artículos 2, 208, 209 fracción III y 211 de la Ley Federal del Derecho de Autor; artículos 64, 103 fracción IV y 104 del Reglamento de la Ley Federal del Derecho de Autor; y artículos 1, 3 fracción I, 4, 8 fracción I y 9 del Reglamento Interior de Instituto Nacional del Derecho de Autor, se expide el presente certificado.

Número de Registro: 03-2023-110911185200-01

Ciudad de México, a 15 de noviembre de 2023

EL DIRECTOR DEL REGISTRO PÚBLICO DEL DERECHO DE AUTOR

JESÚS PARETS GÓMEZ



SECRETARÍA DE CULTURA
INSTITUTO NACIONAL DEL
DERECHO DE AUTOR
DIRECCIÓN DE REGISTRO
PÚBLICO DEL DERECHO DE
AUTOR



CULTURA
SECRETARÍA DE CULTURA



INDAUTOR
INSTITUTO NACIONAL DEL DERECHO DE AUTOR

CERTIFICADO

Registro Público del Derecho de Autor

Para los efectos de los artículos 13, 162, 163 fracción I, 164 fracción I, y demás relativos de la Ley Federal del Derecho de Autor, se hace constar que la **OBRA** cuyas especificaciones aparecen a continuación, ha quedado inscrita en el Registro Público del Derecho de Autor, con los siguientes datos:

AUTORES: ARMENTA GARCIA JESUS ALBANY
CAPO GUTIERREZ JESUS
GARCIA REYES CONRADO IVAN
GONZALEZ NAVARRO FELIX FERNANDO
IBARRA ESQUER JORGE EDUARDO
TORRES CERDA MIGUEL ANGEL
URREA PARTIDA DANIELA ISABEL

TÍTULO: ESP32 CSI WEB COLLECTING TOOL: PLATAFORMA WEB PARA LA CONFIGURACION DE UNA HERRAMIENTA DE RECOLECCION DE WI-FI CSI

RAMA: PROGRAMAS DE COMPUTACION

TITULAR: UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA (CON FUNDAMENTO EN EL ARTICULO 83 DE LA L.F.D.A.)

Con fundamento en lo establecido por el artículo 168 de la Ley Federal del Derecho de Autor, las inscripciones en el registro establecen la presunción de ser ciertos los hechos y actos que en ellas consten, salvo prueba en contrario. Toda inscripción deja a salvo los derechos de terceros. Si surge controversia, los efectos de la inscripción quedarán suspendidos en tanto se pronuncie resolución firme por autoridad competente.

Con fundamento en los artículos 2, 208, 209 fracción III y 211 de la Ley Federal del Derecho de Autor; artículos 64, 103 fracción IV y 104 del Reglamento de la Ley Federal del Derecho de Autor; y artículos 1, 3 fracción I, 4, 8 fracción I y 9 del Reglamento Interior de Instituto Nacional del Derecho de Autor, se expide el presente certificado.

Número de Registro: 03-2024-081214421300-01

Ciudad de México, a 14 de agosto de 2024

EL DIRECTOR DEL REGISTRO PÚBLICO DEL DERECHO DE AUTOR

JESÚS PARETS GÓMEZ



CERTIFICADO

Registro Público del Derecho de Autor

Para los efectos de los artículos 13, 162, 163 fracción I, 164 fracción I, y demás relativos de la Ley Federal del Derecho de Autor, se hace constar que la **OBRA** cuyas especificaciones aparecen a continuación, ha quedado inscrita en el Registro Público del Derecho de Autor, con los siguientes datos:

AUTORES: ARMENTA GARCIA JESUS ALBANY
GARCIA REYES CONRADO IVAN
GONZALEZ NAVARRO FELIX FERNANDO
GUTIERREZ JESUS CARO
IBARRA ESQUER JORGE EDUARDO
TORRES CERDA MIGUEL ANGEL
URREA PARTIDA DANIELA ISABEL

TÍTULO: ESP32 CSI WEB COLLECTING TOOL: HERRAMIENTA PARA LA RECOLECCION DE WI-FI CSI CON DISPOSITIVOS ESP32

RAMA: PROGRAMAS DE COMPUTACION

TITULAR: UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA (CON FUNDAMENTO EN EL ARTICULO 83 DE LA L.F.D.A.)

Con fundamento en el artículo 3° de la Ley Federal del Derecho de Autor el presente certificado ampara única y exclusivamente la obra original Programa de Computación.

Con fundamento en lo establecido por el artículo 168 de la Ley Federal del Derecho de Autor, las inscripciones en el registro establecen la presunción de ser ciertos los hechos y actos que en ellas consten, salvo prueba en contrario. Toda inscripción deja a salvo los derechos de terceros. Si surge controversia, los efectos de la inscripción quedarán suspendidos en tanto se pronuncie resolución firme por autoridad competente.

Con fundamento en los artículos 2, 208, 209 fracción III y 211 de la Ley Federal del Derecho de Autor; artículos 64, 103 fracción IV y 104 del Reglamento de la Ley Federal del Derecho de Autor; y artículos 1, 3 fracción I, 4, 8 fracción I y 9 del Reglamento Interior de Instituto Nacional del Derecho de Autor, se expide el presente certificado.

Número de Registro: 03-2024-062413562000-01

Ciudad de México, a 28 de junio de 2024

EL DIRECTOR DEL REGISTRO PÚBLICO DEL DERECHO DE AUTOR

JESÚS PARETS GÓMEZ



CULTURA
SECRETARÍA DE CULTURA



INDAUTOR
INSTITUTO NACIONAL DEL DERECHO DE AUTOR

CERTIFICADO

Registro Público del Derecho de Autor

Para los efectos de los artículos 13, 162, 163 fracción I, 164 fracción I, y demás relativos de la Ley Federal del Derecho de Autor, se hace constar que la **OBRA** cuyas especificaciones aparecen a continuación, ha quedado inscrita en el Registro Público del Derecho de Autor, con los siguientes datos:

AUTORES: ARMENTA GARCÍA JESÚS ALBANY
TORRES CERDA MIGUEL ANGEL

TÍTULO: CIMARRÓN A LA CIMA

RAMA: PROGRAMAS DE COMPUTACION

TITULARES: ARMENTA GARCÍA JESÚS ALBANY
TORRES CERDA MIGUEL ANGEL

Con fundamento en lo establecido por el artículo 14 fracción III de la Ley Federal del Derecho de Autor, no es objeto de protección como derecho de autor: los esquemas, planes o reglas para realizar actos mentales, juegos o negocios.

Con fundamento en lo establecido por el artículo 168 de la Ley Federal del Derecho de Autor, las inscripciones en el registro establecen la presunción de ser ciertos los hechos y actos que en ellas consten, salvo prueba en contrario. Toda inscripción deja a salvo los derechos de terceros. Si surge controversia, los efectos de la inscripción quedarán suspendidos en tanto se pronuncie resolución firme por autoridad competente.

El presente certificado se expide con fundamento en el Decreto por el que se reforman, adicionan y derogan diversas disposiciones de la Ley Orgánica de la Administración Pública Federal, así como de otras leyes para crear la Secretaría de Cultura, publicado el 17 de diciembre de 2015 en el Diario Oficial de la Federación; artículos 26 y 41 Bis, fracción XVIII de la Ley Orgánica de la Administración Pública Federal; artículos 2, 208, 209 fracción III de la Ley Federal del Derecho de Autor; artículo 69-C de la Ley Federal de Procedimiento Administrativo, de aplicación supletoria de acuerdo con lo establecido por la Ley Federal del Derecho de Autor en su artículo 10; artículo 84 de la Ley General de Mejora Regulatoria; artículos 2, apartado B, fracción IV, 26 y 27 del Reglamento Interior de la Secretaría de Cultura; artículos 103 fracción IV y 104 del Reglamento de la Ley Federal del Derecho de Autor; artículos 1, 3 fracción I, 4, 8 fracción I, 9, 16 y 17 del Reglamento Interior del Instituto Nacional del Derecho de Autor; ACUERDO por el que se establecen los Lineamientos para el uso de la Firma Electrónica Avanzada en los actos y actuaciones de los servidores públicos del Instituto Nacional del Derecho de Autor, publicado en el Diario Oficial de la Federación el 19 de mayo del año dos mil veintiuno; y Acuerdo por el que se establecen las reglas para la presentación, substanciación y resolución de las solicitudes de registro de obras, fonogramas, videogramas y edición de libros en línea ante el Instituto Nacional del Derecho de Autor, publicado el 8 de diciembre de 2021 en el Diario Oficial de la Federación.

1/2



CULTURA
SECRETARÍA DE CULTURA



INDAUTOR
INSTITUTO NACIONAL DEL DERECHO DE AUTOR



Otorga el presente

RECONOCIMIENTO

**A: MIGUEL ANGEL TORRES
CERDA**

Por haber participado en la Segunda edición del
Festival Aeroespacial "Mexicali en las alturas" el día 26 de
Noviembre del 2022 en la ciudad de Mexicali, Baja California.



Nancy Paola Minor Ibarra

NANCY PAOLA MINOR IBARRA
Directora general del Comité de
Turismo y Convenciones de Mexicali

Visit
MEXICALI
COTUCO



EL COMITÉ DE TURISMO Y CONVENCIONES DEL MUNICIPIO DE MEXICALI , BAJA CALIFORNIA
HACE ENTREGA DE LA SIGUIENTE CONSTANCIA

A: Miguel Ángel Torres Cerda

Por haber participado como tallerista en el pabellón de la Ciencia de la Tercera edición del Festival Aeroespacial "Mexicali en las alturas", dirigido al público en general para fomentar el acercamiento a la ciencia, tecnología, ingeniería, artes y matemáticas, llevado a cabo el día 25 de noviembre del 2023 en la ciudad de Mexicali, Baja California.



Nancy Paola Minor Ibarra
DIRECTORA DEL COMITÉ DE TURISMO Y CONVENCIONES
DEL MUNICIPIO DE MEXICALI

Referencias

- [1] I. A. Moosa, “The effectiveness of social distancing in containing covid-19,” *Applied Economics*, vol. 52, no. 58, pp. 6292–6305, 2020. [Online]. Available: <https://doi.org/10.1080/00036846.2020.1789061> 1, 2
- [2] C. Sun and Z. Zhai, “The efficacy of social distance and ventilation effectiveness in preventing covid-19 transmission,” *Sustainable Cities and Society*, vol. 62, p. 102390, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2210670720306119> 1, 2
- [3] S. Di Domenico, M. De Sanctis, E. Cianca, and G. Bianchi, “A trained-once crowd counting method using differential WiFi channel state information,” in *Proceedings of the 3rd International on Workshop on Physical Analytics - WPA '16*. New York, New York, USA: ACM Press, 2016. 5, 8
- [4] Y. Yang, J. Cao, X. Liu, and X. Liu, “Wi-count: Passing people counting with COTS WiFi devices,” in *2018 27th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2018. 5, 8
- [5] S. Depatla and Y. Mostofi, “Crowd counting through walls using wifi,” in *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2018, pp. 1–10. 6, 8
- [6] H. Zou, Y. Zhou, J. Yang, and C. J. Spanos, “Device-free occupancy detection and crowd counting in smart buildings with wifi-enabled iot,” *Energy and Buildings*, vol. 174, pp. 309–322, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0378778817339336> 6, 8
- [7] M. Liyanage, C. Chang, S. Srirama, and S. Loke, “Indoor people density sensing using wi-fi and channel state information,” vol. 61, no. 1, pp. 37–47, 2018. 6, 8

- [8] J. Weppner, B. Bischke, A. Reiss, R. Duerichen, and P. Lukowicz, “Sensing room occupancy levels with signal-to-noise ratio and signal phase and multiple antenna configurations,” in *2018 IEEE SENSORS*, 2018, pp. 1–4. 6, 8
- [9] O. Oshiga, H. U. Suleiman, S. Thomas, P. Nzerem, L. Farouk, and S. Adeshina, “Human detection for crowd count estimation using csi of wifi signals,” in *2019 15th International Conference on Electronics, Computer and Computation (ICECCO)*, 2019, pp. 1–6. 6, 8
- [10] M. Chen, X. Yang, Y. Jin, and M. Zhou, “Dynamic time warping based passive crowd counting using WiFi received signal strength,” in *Lecture Notes in Computer Science*. Cham: Springer International Publishing, 2021, pp. 667–677. 6, 8
- [11] R. F. Brena, E. Escudero, C. Vargas-Rosales, C. E. Galvan-Tejada, and D. Munoz, “Device-free crowd counting using multi-link wi-fi csi descriptors in doppler spectrum,” *Electronics*, vol. 10, no. 3, 2021. [Online]. Available: <https://www.mdpi.com/2079-9292/10/3/315> 6, 8
- [12] H. Choi, M. Fujimoto, T. Matsui, S. Misaki, and K. Yasumoto, “Wi-cal: Wifi sensing and machine learning based device-free crowd counting and localization,” *IEEE Access*, vol. 10, pp. 24 395–24 410, 2022. 7, 8
- [13] Z. Liu, R. Yuan, Y. Yuan, Y. Yang, and X. Guan, “A sensor-free crowd counting framework for indoor environments based on channel state information,” *IEEE Sens. J.*, vol. 22, no. 6, pp. 6062–6071, 2022. 7, 8
- [14] H. Hou, S. Bi, L. Zheng, X. Lin, Y. Wu, and Z. Quan, “DASECount: Domain-agnostic sample-efficient wireless indoor crowd counting via few-shot learning,” *IEEE Internet Things J.*, vol. 10, no. 8, pp. 7038–7050, 2023. 7, 8
- [15] M. De Sanctis, T. Rossi, S. Di Domenico, E. Cianca, G. Ligresti, and M. Ruggieri, “LTE signals for device-free crowd density estimation through CSI secant set and SVD,” *IEEE Access*, vol. 7, pp. 159 943–159 951, 2019. 7, 8
- [16] S. Kianoush, S. Savazzi, and M. Nicoli, “Device-free crowd sensing in dense wifi mimo networks: Channel features and machine learning tools,” in *2018 15th Workshop on Positioning, Navigation and Communications (WPNC)*, 2018, pp. 1–6. 8, 11
- [17] O. T. Ibrahim, W. Gomaa, and M. Youssef, “CrossCount: A deep learning system for device-free human counting using WiFi,” *IEEE Sens. J.*, vol. 19, no. 21, pp. 9921–9928, 2019. 8, 11

- [18] W. Shi, U. Tahir, H. Zhang, and J. Zhao, “Design and implementation of non-intrusive stationary occupancy count in elevator with wifi,” in *Broadband Communications, Networks, and Systems*, Q. Li, S. Song, R. Li, Y. Xu, W. Xi, and H. Gao, Eds. Cham: Springer International Publishing, 2019, pp. 3–19. [9](#), [11](#)
- [19] U. Singh, J.-F. Determe, F. Horlin, and P. D. Doncker, “Crowd forecasting based on WiFi sensors and LSTM neural networks,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 9, pp. 6121–6131, 2020. [9](#), [11](#)
- [20] R. Zhou, X. Lu, Y. Fu, and M. Tang, “Device-free crowd counting with wifi channel state information and deep neural networks.” Springer International Publishing, 2020. [9](#), [11](#)
- [21] C. Tang, W. Li, S. Vishwakarma, K. Chetty, S. Julier, and K. Woodbridge, “Occupancy detection and people counting using wifi passive radar,” in *2020 IEEE Radar Conference (RadarConf20)*, 2020, pp. 1–6. [9](#), [11](#)
- [22] W. Zhuang, Y. Shen, C. Gao, L. Li, H. Sang, and F. Qian, “Adaptive scheme for crowd counting using off-the-shelf wireless routers,” *Computer Systems Science and Engineering*, vol. 41, no. 1, pp. 255–269, 2022. [Online]. Available: <http://www.techscience.com/csse/v41n1/44797> [9](#), [11](#)
- [23] Y. Zhang, J. Li, and H. Wang, “Indoor seat occupancy classification with wi-fi channel state information and machine learning methods,” in *2021 4th International Conference on Sensors, Signal and Image Processing*, ser. SSIP 2021. New York, NY, USA: Association for Computing Machinery, 2021, p. 72–81. [Online]. Available: <https://doi.org/10.1145/3502814.3502826> [9](#), [11](#)
- [24] Z. Guo, F. Xiao, B. Sheng, L. Sun, and S. Yu, “TWCC: A robust through-the-wall crowd counting system using ambient WiFi signals,” *IEEE Trans. Veh. Technol.*, vol. 71, no. 4, pp. 4198–4211, 2022. [10](#), [11](#)
- [25] F. Wang, F. Zhang, C. Wu, B. Wang, and K. J. R. Liu, “Respiration tracking for people counting and recognition,” *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5233–5245, 2020. [10](#), [11](#)
- [26] Y. Li and G. L. Stuber, Eds., *Orthogonal frequency division multiplexing for wireless communications*, 2006th ed. Nueva York, NY, Estados Unidos de América: Springer, 2006. [12](#)
- [27] Y. Ma, G. Zhou, and S. Wang, “Wifi sensing with channel state information: A survey,” *ACM Comput. Surv.*, vol. 52, no. 3, Jun. 2019. [Online]. Available: <https://doi.org/10.1145/3310194> [13](#)

- [28] D. M. P. Murti, U. Pujiyanto, A. P. Wibawa, and M. I. Akbar, “K-nearest neighbor (k-nn) based missing data imputation,” in *2019 5th International Conference on Science in Information Technology (ICSITech)*, 2019, pp. 83–88. [13](#)
- [29] W. K. Pratt, “Digital image processing: PIKS inside.” John Wiley & Sons, 2001, pp. 271–273. [14](#)
- [30] F. R. Hampel, “The influence curve and its role in robust estimation,” *Journal of the American Statistical Association*, vol. 69, no. 346, pp. 383–393, 1974. [Online]. Available: <http://www.jstor.org/stable/2285666> [15](#)
- [31] “hampel,” <https://es.mathworks.com/help/signal/ref/hampel.html>, accessed: 2024-4-23. [15](#)
- [32] G. Darbellay and I. Vajda, “Estimation of the information by an adaptive partitioning of the observation space,” *IEEE Transactions on Information Theory*, vol. 45, no. 4, pp. 1315–1321, 1999. [15](#)
- [33] C. Ding and H. Peng, “Minimum redundancy feature selection from microarray gene expression data,” in *Computational Systems Bioinformatics. CSB2003. Proceedings of the 2003 IEEE Bioinformatics Conference. CSB2003*, 2003, pp. 523–528. [15](#)
- [34] F. González and L. Belanche, *Feature selection in proton magnetic resonance spectroscopy data of brain tumors*. Università degli Studi di Salerno, 2011, p. 1–8. [Online]. Available: <http://hdl.handle.net/2117/18003> [16](#)
- [35] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne.” *Journal of machine learning research*, vol. 9, no. 11, 2008. [19](#)
- [36] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian network classifiers,” *Mach. Learn.*, vol. 29, no. 2/3, pp. 131–163, 1997. [19](#)
- [37] S. Sun and R. Huang, “An adaptive k-nearest neighbor algorithm,” in *2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 1, 2010, pp. 91–94. [19](#)
- [38] A. Mammone, M. Turchi, and N. Cristianini, “Support vector machines: Support vector machines,” *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 1, no. 3, pp. 283–289, 2009. [20](#)
- [39] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A survey of convolutional neural networks: Analysis, applications, and prospects,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, 2022. [21](#)

- [40] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, “1d convolutional neural networks and applications: A survey,” *Mechanical Systems and Signal Processing*, vol. 151, p. 107398, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0888327020307846> 21, 54
- [41] S. J. Chapman, D. P. Hewett, and L. N. Trefethen, “Mathematics of the faraday cage,” *SIAM Rev. Soc. Ind. Appl. Math.*, vol. 57, no. 3, pp. 398–417, 2015. 21
- [42] A. Muñoz Herrerías, “Jaula de faraday,” *Vida Científica Boletín Científico de la Escuela Preparatoria No. 4*, vol. 5, no. 10, jul. 2017. [Online]. Available: <https://repository.uaeh.edu.mx/revistas/index.php/prepa4/article/view/2582> 21
- [43] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, “Tool Release: Gathering 802.11n Traces with Channel State Information,” *ACM SIGCOMM CCR*, vol. 41, no. 1, p. 53, 2011. 25
- [44] Y. Xie, Z. Li, and M. Li, “Precise power delay profiling with commodity WI-fi,” *IEEE Trans. Mob. Comput.*, vol. 18, no. 6, pp. 1342–1355, 2019. 26
- [45] E. Eggers, “Diseñando una jaula de faraday para redes Wi-Fi,” <https://es.linkedin.com/pulse/dise~n}ando-una-jaula-de-faraday-para-redes-wi-fi-eliot-eggers>, 1528, accessed: 2023-10-6. 36