

TÉCNICAS DE DEEP LEARNING ENFOCADAS A LA ESTIMACIÓN ADAPTATIVA DE CANALES EN REDES DE QUINTA GENERACIÓN



ANDRÉS FELIPE CHACÓN ALONSO
MIGUEL ÁNGEL FUENTES RAMÍREZ



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS
Acreditación Institucional de Alta Calidad

Técnicas de Deep Learning enfocadas a la estimación adaptativa de canales en redes de quinta generación

Autores

ANDRÉS FELIPE CHACÓN ALONSO

Código: 20182005007

MIGUEL ÁNGEL FUENTES RAMÍREZ

Código: 20182005007

Director:

PHD CARLOS ARTURO SUÁREZ FAJARDO

Línea de investigación:

Telecomunicaciones e Inteligencia Computacional

Grupo de Investigación:

Grupo de Radiación Electromagnética y Comunicaciones Ópticas (GRECO)

Monografía como requisito para optar por el título de:

Ingeniero Electrónico

Proyecto Curricular de Ingeniería Electrónica

Universidad Distrital Francisco José de Caldas

Bogotá DC, Colombia

2024

Resumen

En la actualidad, es innegable reconocer la importancia de las comunicaciones móviles en el día a día, ya que proveen a los usuarios de la posibilidad de tener una comunicación ubicua, desde cualquier lugar y en cualquier momento. La tendencia en el crecimiento de la cantidad de usuarios con dispositivos móviles, implica la constante búsqueda de optimizar los recursos disponibles para ofrecer un servicio de calidad, confiable y eficiente. Por tal motivo, las redes de quinta generación (5G) son uno de los principales focos de la transformación digital, promoviendo tres servicios fundamentales: comunicación ultra fiable de baja latencia, comunicación masiva de tipo máquina y banda ancha móvil mejorada.

El aprendizaje profundo es una rama de la inteligencia artificial que se ha vuelto esencial en los últimos años. La posibilidad de que las máquinas aprendan información relevante de grandes conjuntos de datos y así mismo realicen tareas complejas, da pie a impulsar tecnologías emergentes que puedan ser aplicadas en diferentes campos de estudio, como la economía, agricultura, ingeniería e incluso la medicina. Los modelos de aprendizaje profundo permiten trabajar con diferentes datos de entrada a partir del concepto de neurona artificial, con la cual se pueden desarrollar modelos por capas que puedan hallar patrones y características complejas que cumplan con tareas de regresión o clasificación.

En este documento se presenta una propuesta de aplicación de las técnicas de aprendizaje profundo en el campo de las telecomunicaciones, concretamente en un escenario de redes móviles 5G en donde la estimación de canal se hace a partir de la inserción de señales piloto y modelos convolucionales. Se realiza una comparativa de cada uno de los modelos propuestos en un entorno de simulación hecho en MATLAB, verificando la calidad de la estimación y el rendimiento general de cada modelo dentro de un sistema de comunicaciones a partir de métricas como el MSE, BER, EVM y tiempo de estimación. Las evaluaciones de cada modelo se realizan variando algunos parámetros como el perfil de retardo, la dispersión del retardo, el nivel de ruido, la velocidad del usuario y la modulación que se usa para transmitir bits aleatorios.

Dedicatoria

Andrés Felipe Chacón Alonso

A mi padre Libardo, que desde el cielo me ha guiado y me ha motivado a seguir adelante, en alcanzar y cumplir mis metas. A mi madre, Martha y a mis hermanas Carolina y Paola, quienes me brindan constantemente su apoyo incondicional, amor y confianza. Y por supuesto, a mi compañero de las mil traspasadas, Kiwi.

Miguel Ángel Fuentes Ramírez

A mis padres Helena y José Miguel, a mi hermano Esteban, quienes con tanto esfuerzo y sacrificio han dado todo por mí y han sido mi motor todos estos años, gracias por apoyarme incondicionalmente, guiarme, cuidarme, amarme y soportarme en este proceso, gracias a ellos me he convertido en quien soy.

Agradecimientos

En primer lugar nos gustaría agradecer al profesor Gustavo Adolfo Puerto Leguizamón, quien con su dedicación, orientación y valiosas sugerencias, nos guiaron a lo largo del proceso de investigación y escritura de esta memoria. También, queremos agradecer al profesor Carlos Arturo Suárez Fajardo y al grupo de investigación GRECO, por brindarnos los espacios y herramientas necesarias para completar este trabajo de grado.

Finalmente, queremos extender nuestro agradecimiento a los docentes de la Universidad Distrital Francisco José de Caldas que nos brindaron su sabiduría y conocimiento a lo largo de nuestra carrera universitaria. Sus enseñanzas han sido pilares en nuestra formación personal y profesional.

Índice general

Resumen	i
Dedicatoria	ii
Agradecimiento	iii
Acrónimos	ix
Índice de figuras	xii
Índice de cuadros	xvi
1 Descripción general del proyecto	1
1.1 Formulación del problema	1
1.2 Motivación	3
1.3 Objetivo General	4
1.4 Objetivos Específicos	4
1.5 Alcances y limitaciones	4
1.6 Estructura de la memoria	5
2 Fundamentos Teóricos de 5G	8
2.1 Generalidades del 5G	8
2.1.1 Evolución en las comunicaciones móviles	9
2.1.2 Retos actuales	11
2.1.3 Ventajas	12
2.2 Capa física	13
2.2.1 Bandas de operación	13
2.2.2 Formas de onda	14
2.2.3 Prefijo cíclico	16
2.2.4 Numerología OFDM	16
2.3 Recursos físicos	18
2.4 Canales físicos	19

2.5	Señales físicas	20
2.5.1	En enlace de bajada	21
2.5.2	En enlace de subida	22
2.6	Modulaciones soportadas por 5G	22
2.7	Estructuras de transmisión	23
2.7.1	En el dominio del tiempo	23
2.7.2	En el dominio de la frecuencia	24
3	Canal de comunicaciones	25
3.1	Generalidades	25
3.2	Ruido	27
3.3	Desvanecimiento	28
3.3.1	Distribución de Rice (LoS)	29
3.3.2	Distribución de Rayleigh (NLoS)	30
3.4	Canal variante en el tiempo	30
3.5	Características en un canal variante en el tiempo	32
3.5.1	Dispersión	32
3.5.2	Ancho de banda de coherencia	34
3.5.3	Dispersión Doppler	34
3.5.4	Tiempo de coherencia	35
3.6	Modelo de canal TDL	35
3.7	Estimación de canal	40
3.7.1	Estimador por mínimos cuadrados	41
3.7.2	Estimador MMSE	41
3.8	Ecualización de canal	42
3.9	Compensación de fase	45
3.10	Métricas de desempeño	47
3.10.1	Magnitud del Vector de Error	47
3.10.2	Tasa de Error de Bit	48
4	Aprendizaje profundo	49
4.1	Generalidades	49
4.2	Algoritmos de aprendizaje supervisado	51
4.3	El aprendizaje profundo	52

4.4	Redes neuronales artificiales	53
4.4.1	El concepto biológico de la neurona	54
4.4.2	Modelado de neuronas artificiales	55
4.4.3	Funciones de activación	57
4.4.4	Función de propagación	58
4.4.5	Función de salida	59
4.4.6	Conceptos adicionales	59
4.4.7	Ventajas de uso	66
4.4.8	Tipos de redes neuronales artificiales	66
4.5	Fase de aprendizaje	68
4.5.1	Propagación hacia adelante	68
4.5.2	Propagación hacia atrás	69
4.6	Error cuadrático medio	71
5	Arquitecturas de redes neuronales artificiales	72
5.1	Perceptrón Multicapa (MLP)	74
5.2	Redes neuronales convolucionales	75
5.2.1	Componentes	75
5.2.2	Hiper-parámetros	78
5.3	Redes neuronales recurrentes	79
5.4	Derivaciones en las arquitecturas de CNN	81
5.4.1	Autoencoder	81
5.4.2	Red <i>denoising</i> (DnCNN)	82
6	Estado del arte	84
7	Desarrollo de ingeniería	87
7.1	Generalidades	87
7.2	Configuración del sistema de comunicaciones en MATLAB	89
7.2.1	Consideraciones de valor	89
7.2.2	Funciones de interés	91
7.2.3	Transmisión	92
7.2.4	Canal de comunicaciones	93
7.2.5	Recepción	93
7.3	Generación de los datos de entrenamiento	96

7.4	Estimadores de canal con ANN	98
7.4.1	Preprocesamiento de los datos	100
7.4.2	Modelos con redes neuronales convolucionales	102
7.4.3	Autoencoder	105
7.4.4	Red neuronal tipo <i>denoising</i>	107
8	Resultados	109
8.1	Generalidades	109
8.2	Rendimiento individual de cada modelo	111
8.3	Selección de estimadores de canal ANN	114
8.3.1	Configuración del sistema de comunicaciones y simulación	115
8.3.2	Modelos de redes neuronales convolucionales	116
8.3.3	Modelos de redes Autoencoder	120
8.3.4	Modelos de redes denoising	123
8.4	Comparación entre modelos seleccionados con diferentes escenarios	125
8.4.1	Parámetros de la simulación	125
8.4.2	Canal con perfil de retardo TDL-B (NLoS)	127
8.4.3	Canal con perfil de retardo TDL-E (LoS)	133
9	Conclusiones y líneas futuras	139
9.1	Conclusiones	139
9.2	Futuras líneas de investigación	141
9.2.1	Enfocadas a Inteligencia Computacional	141
9.2.2	Enfocadas a las Telecomunicaciones	142
9.3	Aportes	142
A	Anexo : Resultados canales TDL-A, TDL-C, TDL-D	144
A.1	TDL-A (NLoS)	144
A.1.1	Modulación QPSK	144
A.1.2	Modulación 64 QAM	145
A.1.3	Modulación 256 QAM	146
A.2	TDL-C (NLoS)	147
A.2.1	Modulación QPSK	147
A.2.2	Modulación 64 QAM	148
A.2.3	Modulación 256 QAM	149

A.3	TDL-D (LoS)	150
A.3.1	Modulación QPSK	150
A.3.2	Modulación 64 QAM	151
A.3.3	Modulación 256 QAM	152
B	Anexo : Herramienta de simulación y aprendizaje 5GNR (DLS5G)	154
B.1	Arquitectura del sistema	154
B.2	Diseño del modelamiento de datos	156
B.2.1	Funcion Simulation_Configuration	158
B.2.2	Variable_Generation	160
B.2.3	Variables_initialization	161
B.2.4	Transmitter	161
B.2.5	Receiver	163
B.2.6	Función Phase_Compensation	164
B.2.7	Función Metricas_Slot	165
B.2.8	Función Matrix_metrics	166
B.2.9	Función Perfect_Estimation	167
B.2.10	Función Practical_Estimation	167
B.2.11	Funciones CNN_Estimation y CNN_2_Estimation	167
B.2.12	Función Autoencoder_Estimation	167
B.2.13	Función Lineal_Estimation	168
B.2.14	Función Denoising_Estimation	168
B.2.15	Función Denoising_Estimation_2	168
B.3	Procesos y servicios	169
B.4	GUI - Ejemplos de utilización	171
B.4.1	Configuración General	171
B.4.2	Simulación Temporal	173
B.4.3	Simulación en tiempo real	174
	Referencias	178

Acrónimos

ML Aprendizaje Automático

DL Aprendizaje Profundo

OFDM Multiplexación por División de Frecuencias Ortogonales

CP Prefijo Cíclico

UM Unidad Móvil

EB Estación Base

TDL Línea de Retardo de Pulsación

ANN Red Neuronal Artificial

MLP Perceptrón Multicapa

CNN Red Neuronal Convolucional

RNN Red Neuronal Recurrente

MSE Error Cuadrático Medio

MMSE Mínimo Error Cuadrático Medio

LS Mínimos Cuadrados

QoS Calidad del Servicio

AWGN Ruido Blanco Aditivo Gaussiano

CDL Cluster Delay Line

MIMO Múltiples Entradas Múltiples Salidas

SISO Una Entrada Una Salida

LTV Lineal Variante en el Tiempo

U/L Enlace de Subida

D/L Enlace de Bajada

BER Tasa de error de bit

EVM Magnitud del Vector Error

LSTM Red de Memoria a Corto Plazo Largo

FDMA Acceso Múltiple por División de Frecuencias

CDMA Acceso Múltiple por División de Códigos

TDMA Acceso Múltiple por División de Tiempos

SCS Espaciado entre Subportadoras

PDSCH Canal Físico Compartido en D/L

LoS Línea de Vista

NLoS Sin Línea de Vista

SNR Relación Señal a Ruido

FR1 Rango de frecuencias 1

FR2 Rango de frecuencias 2

5G Quinta Generación

4G Cuarta Generación

LTE Evolución a Largo Plazo

OFDMA FDMA Ortogonal

CS FDMA FDMA de Única Portadora

IoT Internet de las Cosas

QPSK Modulación por Desplazamiento de Fase en Cuadratura

IMT Telecomunicaciones Móviles Internacional

LTE-A LTE Avanzado

3GPP Proyecto Asociación de Tercera Generación

CP-OFDM OFDM con Prefijo Cíclico

DFT-OFDM OFDM con Transformada Discreta de Fourier

PSK Modulación por Desplazamiento de Fase

M-QAM Modulación por Amplitud de Cuadratura

DS Dispersión del Retardo

RE Elemento de Recurso

RB Bloque de Recursos

PDSCH Canal Físico Compartido de Enlace de Bajada

DM-RS Señal de Referencia de Demodulación

TDD Duplexación por División de Tiempo

TTI Intervalos de Tiempo de Transmisión

ISI Interferencia entre Símbolos

MLSE Estimador de Máxima Verosimilitud

SGD Descenso de Gradiente Estocástico

PT-RS Señal de referencia para seguimiento de fase

DnCNN Red Neuronal Convolutacional Denoising

Índice de figuras

2.1	Triángulo de los retos del 5G [10]	12
2.2	Espectro de una señal OFDM (Autoría propia)	15
2.3	Construcción de un símbolo OFDM con la inserción del prefijo cíclico [15]	16
2.4	Numerología OFDM para 5G [16]	17
2.5	Rejilla de recursos: (azul) RE, (rojo) RB. (Autoría propia)	19
2.6	Inserción de señales piloto DM-RS en la rejilla de recursos. (Autoría propia)	21
2.7	Estructura de transmisión en el dominio del tiempo. (Autoría propia)	23
3.1	Comportamiento de los tipos de desvanecimiento en una gráfica de d vs P_R [24]	28
3.2	Propagación multi-trayectoria [23]	29
3.3	Funciones de densidad de probabilidad de la envolvente [20]	30
3.4	Respuesta al impulso de un canal LTV [20]	31
3.5	Modelo estadístico de un canal LTV [20]	32
3.6	Diagrama de bloques de un canal TDL. (Autoría propia)	36
3.7	Tipos de distribución de las señales piloto en la cuadrícula de recursos. [29]	40
3.8	Inserción de señales PT-RS en la rejilla de recursos de 5G. [33]	46
3.9	Diagramas de constelaciones 16QAM. [32]	46
3.10	Representación gráfica de EVM. [34]	47
4.1	Subcategorías de la inteligencia artificial. (Autoría propia)	50
4.2	Partes de la neurona biológica. [47]	55
4.3	Diagrama de bloques del modelo de una neurona artificial. (Autoría propia)	56
4.4	Estructura por capas de una Red Neuronal Artificial (ANN). [48]	57
4.5	Funciones de activación empleadas en las ANN. (Autoría propia)	58
4.6	Variación de μ para hallar el máximo de una función. (Autoría propia)	62
4.7	Comparativa del descenso de gradiente estocástico. [55]	64
4.8	Comparativa del desempeño de los métodos de optimización. [56]	65
4.9	Taxonomía de modelos de aprendizaje profundo. [43]	67
4.10	Diagrama de bloques del aprendizaje en una ANN. (Autoría propia)	71

5.1	Diferencia entre los clasificadores. (Autoría propia)	73
5.2	Ejemplo de un MLP. (Autoría Propia)	74
5.3	Topología de una red CNN para clasificación. [60]	76
5.4	Filtros entrenables en una capa convolucional. [62]	76
5.5	Ejemplo de funcionamiento de la capa de agrupación. (Autoría propia)	77
5.6	Ejemplo de la capa de abandono. [64]	78
5.7	Modelado de una neurona recurrente. (Autoría propia)	80
5.8	Estructura de una red Autoencoder. (Autoría propia)	82
5.9	Arquitectura de una red DnCNN. [69]	82
7.1	Diagrama de bloques del sistema desarrollado. (Autoría propia)	88
7.2	Distribución del conjunto de datos. (Autoría propia)	96
7.3	Variaciones presentes en los datos de entrenamiento. (Autoría propia)	98
7.4	Estructura por bloques del estimador de canal. (Autoría propia)	100
7.5	Preprocesamiento en la entrada y estimación con modelo neuronal. (Autoría propia)	101
8.1	Comparación de rendimiento entre los modelos con los datos de evaluación	112
8.2	Muestras de referencia para la comparativa de la calidad de predicción con un SNR de 0 dB	116
8.3	Muestras de la estimación hecha por cada modelo CNN estándar con 0 dB	117
8.4	Etiquetas de las curvas de cada modelo CNN estándar	118
8.5	Métricas de desempeño de las CNN estándar	118
8.6	Muestras de la estimación hecha por cada modelo Autoencoder con 0 dB	120
8.7	Etiquetas de las curvas de cada modelo Autoencoder	121
8.8	Métricas de desempeño de los Autoencoder	121
8.9	Muestras de la estimación hecha por cada modelo Denoising con 0 dB	123
8.10	Etiquetas de las curvas de cada modelo Denoising	123
8.11	Métricas de desempeño de las DnCNN	124
8.12	Nomenclaturas simulaciones asociadas Canales TDL	126
8.13	Velocidad Usuario 5 km/h	127
8.14	Velocidad Usuario 30 km/h	128
8.15	Velocidad Usuario : 100 km/h	128
8.16	Velocidad Usuario 5 km/h	129
8.17	Velocidad Usuario 30 km/h	130

8.18 Velocidad Usuario : 100 km/h	130
8.19 Velocidad Usuario 5 km/h	131
8.20 Velocidad Usuario 30 km/h	132
8.21 Velocidad Usuario : 100 km/h	132
8.22 Velocidad Usuario 5 km/h	133
8.23 Velocidad Usuario 30 km/h	134
8.24 Velocidad Usuario : 100 km/h	134
8.25 Velocidad Usuario 5 km/h	135
8.26 Velocidad Usuario 30 km/h	135
8.27 Velocidad Usuario : 100 km/h	136
8.28 Velocidad Usuario 5 km/h	136
8.29 Velocidad Usuario 30 km/h	137
8.30 Velocidad Usuario : 100 km/h	137
A.1 Velocidad Usuario 5 km/h	144
A.2 Velocidad Usuario 30 km/h	145
A.3 Velocidad Usuario : 100 km/h	145
A.4 Velocidad Usuario 5 km/h	145
A.5 Velocidad Usuario 30 km/h	146
A.6 Velocidad Usuario : 100 km/h	146
A.7 Velocidad Usuario 5 km/h	146
A.8 Velocidad Usuario 30 km/h	147
A.9 Velocidad Usuario : 100 km/h	147
A.10 Velocidad Usuario 5 km/h	147
A.11 Velocidad Usuario 30 km/h	148
A.12 Velocidad Usuario : 100 km/h	148
A.13 Velocidad Usuario 5 km/h	148
A.14 Velocidad Usuario 30 km/h	149
A.15 Velocidad Usuario : 100 km/h	149
A.16 Velocidad Usuario 5 km/h	149
A.17 Velocidad Usuario 30 km/h	150
A.18 Velocidad Usuario : 100 km/h	150
A.19 Velocidad Usuario 5 km/h	150
A.20 Velocidad Usuario 30 km/h	151

A.21 Velocidad Usuario : 100 km/h	151
A.22 Velocidad Usuario 5 km/h	151
A.23 Velocidad Usuario 30 km/h	152
A.24 Velocidad Usuario : 100 km/h	152
A.25 Velocidad Usuario 5 km/h	152
A.26 Velocidad Usuario 30 km/h	153
A.27 Velocidad Usuario : 100 km/h	153
B.1 Estructura principal de directorios	155
B.2 Archivos resultantes simulaciones	155
B.3 Ficheros GUI	156
B.4 Ficheros Auxiliares	156
B.5 Ficheros Módulos complementarios	156
B.6 Diagrama de clases función de configuración de simulación	158
B.7 Diagrama de clases función de Generación de variables	160
B.8 Diagrama de clases función de Inicialización de variables	161
B.9 Diagrama de clases Transmisor	162
B.10 Diagrama de clases Receptor	163
B.11 Diagrama de clases función de compensación de fase	164
B.12 Diagrama de casos de uso app DLS5G	169
B.13 Diagrama de flujo Script Simulación Final	170
B.14 Interfaz configuración de parámetros	171
B.15 Interfaz simulación con visualización de métricas	173
B.16 Interfaz simulación en tiempo real	174
B.17 Interfaz para gráficas múltiples a partir de base de datos	175
B.18 Interfaz para gráfica simple a partir de base de datos	176
B.19 Interfaz resultados constelaciones	176
B.20 Interfaz resultados transmisión imágenes	177

Índice de cuadros

2.1	Evolución de las características en cada generación de las redes móviles [8]	11
2.2	Generalidades sobre los rangos de operación en 5G.	14
2.3	Características de las numerologías de 5G [8] [14]	18
3.1	Caracterización del perfil de retardo TDL-A	37
3.2	Caracterización del perfil de retardo TDL-B	37
3.3	Caracterización del perfil de retardo TDL-C	38
3.4	Caracterización del perfil de retardo TDL-D	39
3.5	Caracterización del perfil de retardo TDL-E	39
3.6	Requisitos de EVM para cada modulación soportada en 5G. [36]	48
7.1	Parámetros de entrenamiento de los estimadores de canal	101
7.2	Topología de la referencia: Modelo CNN 1. Tomado de [89]	102
7.3	Topología del modelo CNN 2	103
7.4	Topología del modelo CNN 3	103
7.5	Topología del modelo CNN 4	103
7.6	Topología del modelo CNN 5	104
7.7	Topología del modelo CNN 6	104
7.8	Topología del modelo CNN 7	104
7.9	Topología del modelo CNN 8	105
7.10	Topología del modelo Autoencoder 1	106
7.11	Topología del modelo Autoencoder 2	106
7.12	Topología del modelo Autoencoder 3	107
7.13	Topología del modelo DnCNN	108
8.1	Evaluación del desempeño de los modelos con los datos de evaluación	112
8.2	Parámetros para la simulación con condiciones fijas en el canal de comunicaciones	115
8.3	Parámetros utilizados durante las simulaciones	126

1

Descripción general del proyecto

Índice

1.1	Formulación del problema	1
1.2	Motivación	3
1.3	Objetivo General	4
1.4	Objetivos Específicos	4
1.5	Alcances y limitaciones	4
1.6	Estructura de la memoria	5

1.1 Formulación del problema

En el informe anual de internet de Cisco [1] se evalúa la transformación digital, se presenta la importancia vital del Quinta Generación (5G) y la necesidad de optimizar las redes de comunicaciones para hacerle frente al aumento significativo de la demanda. Con proyecciones que sugieren que casi dos tercios de la población mundial tendrán acceso a internet para 2023, alcanzando los 5.300 millones de usuarios y un crecimiento explosivo de dispositivos conectados que alcanzarán los 29.3 mil millones para 2023 por lo que se destaca la necesidad de adaptación a esta evolución digital. La movilidad también experimentará un auge, con más del 70% de la población mundial con conectividad móvil en 2023, y el 5G jugará un papel crucial al representar más del 10% de los dispositivos móviles globales en ese año. El informe resalta mejoras sustanciales en el rendimiento de la red, con velocidades de banda ancha fija y móvil que se duplicaran y triplicaran, respectivamente en 2023, por lo que este panorama demuestra la necesidad de adoptar medidas para fortalecer infraestructuras digitales y asegurar el éxito de la evolución tecnológica en materia de comunicaciones.

Considerando el aumento previsto en la demanda de los usuarios, el retraso de la información en las redes de comunicaciones es un gran inconveniente para la respuesta en tiempo real, esto se

aprecia en lo que se conoce como latencia, si se compara la tecnología Cuarta Generación (4G) con la 5G, esta última presenta enormes diferencias en términos de velocidad y cantidad de recursos compartidos como se menciona en el estudio realizado por OpenSignal [2]. En otro estudio de GSMA [3] se especifica que la latencia de las redes 5G es de 1 ms, esto conlleva a que se puedan tener excelentes respuestas en tiempo real.

El despliegue de redes de quinta generación supone un reto en la necesidad de reducir los tiempos de respuesta, optimizar los recursos disponibles y garantizar Calidad del Servicio (QoS), para ello se identificaron los factores mejorables, tanto en hardware como de software, donde la estimación del canal ha sido un punto vital dada su importancia en el establecimiento de las comunicaciones entre la Estación Base (EB) y la Unidad Móvil (UM), para realizar esta tarea se han desarrollado soluciones como modelos estadísticos, pero estos podrían presentar limitaciones para situaciones de mayor complejidad dado la gran cantidad de parámetros que se calculan como operaciones matriciales con un elevado número de coeficientes, por lo que se requerirán plantear soluciones más robustas, eficientes y dinámicas.

Para garantizar una estimación de canal con mayor dinamismo y adaptabilidad a los entornos 5G, se puede pensar en recurrir a enfoques más robustos y complejos, como los modelos de Aprendizaje Automático (ML) y Aprendizaje Profundo (DL), que han sido muy populares en los últimos años, como lo indica la tendencia de búsquedas en Google [4]. La popularidad de estos modelos se relaciona con las aplicaciones diarias, como las recomendaciones en línea, traducciones automáticas, reconocimiento facial e incluso en el sector de la medicina, con la identificación de enfermedades mediante el análisis de datos. Un término asociado es el de ANN, una aproximación al concepto biológico de la neurona, con la que se busca aprovechar la capacidad de aprendizaje de patrones en los datos para realizar predicciones y tomar decisiones en tareas complejas.

Con base en lo descrito anteriormente, resulta pertinente encaminar el problema de las estimaciones de canales a través del uso de modelos que presenten una mejor adaptabilidad y robustez en los entornos dinámicos bajo los cuales operan las tecnologías móviles, sobre todo la quinta generación, donde es crítico reducir los tiempos y garantizar una calidad del servicio, por lo que surgen las preguntas: ¿cómo se pueden incorporar las técnicas de aprendizaje profundo como solución adaptable a la estimación de canales de quinta generación? ¿Cómo es la respuesta que tienen estos modelos ante variaciones en las condiciones de un canal de comunicaciones?

1.2 Motivación

En la actualidad resulta indispensable mantener un mundo conectado, donde los avances tecnológicos sean soporte de las personas y les permitan enterarse de lo que acontece a su alrededor, debido el creciente aumento de la cantidad de dispositivos que se conectan a una red, se tienen problemas en el rendimiento en las conexiones, por ejemplo, empiezan a presentar problemas asociados con: la alta latencia, la baja velocidad de transmisión y el consumo energético que acarrea solventar la necesidad de un gran grupo de usuarios. Hay una alta demanda que no se cubre completamente con las tecnologías aplicadas actualmente, esta situación revela ciertos problemas en las redes de comunicaciones de 4G Evolución a Largo Plazo (LTE) y presenta oportunidades de desarrollo para las futuras tecnologías, dando pie a nuevas tecnologías donde se puedan incorporar los modelos computacionales con las telecomunicaciones, ya que juntar ambos conocimientos, supondría un avance notable en el paradigma donde se puede tener una transmisión y recepción de la información confiable, flexible, segura y adaptativa a las necesidades de cada usuario y al entorno en el que se encuentra.

La implementación de las redes 5G representa un hito destacado en las comunicaciones móviles, introduciendo desafíos técnicos particulares, especialmente en la gestión efectiva de los canales de comunicación. La naturaleza de los entornos 5G, definidos por una mayor concentración de dispositivos y una asignación espectral eficaz, ha generado la necesidad de aplicar métodos avanzados para estimar los canales de una forma más precisa, ya que es importante disponer de la información detallada de las condiciones del canal, para que los sistemas 5G ajusten la transmisión de manera dinámica para superar obstáculos, minimizar las interferencias y maximizar la eficiencia espectral. Este trabajo aborda la creciente complejidad de las redes de quinta generación, enfrentándose a desafíos en la gestión eficiente de canales de comunicación en entornos con alta demanda, esta iniciativa propone una aproximación innovadora centrada en el aprendizaje profundo para la estimación de canales.

La aplicación exitosa de técnicas de DL en la estimación de canales no solo optimizará el rendimiento de las redes 5G, sino que también tendrá un impacto directo en QoS. Al mejorar la precisión de las estimaciones de canales, se logrará una asignación espectral más eficiente, reduciendo la interferencia y aumentando la confiabilidad de las comunicaciones. Este avance no solo contribuirá a la creación de una infraestructura más robusta, sino que también se tendrán conexiones más

estables y velocidades de transmisión más rápidas para los usuarios finales.

El rol de un ingeniero electrónico contempla solventar problemáticas desde diferentes enfoques, como control, telecomunicaciones, telemática, inteligencia computacional o bioingeniería; pero cuando se integran los enfoques hacia un objetivo, se tienen como resultados tecnologías innovadoras, que se pueden encaminar hacia la búsqueda de mejoras en la calidad de vida de las personas, optimizar los sistemas de comunicaciones modernos permite ofrecer una conectividad segura, rápida y confiable, manteniendo un mundo conectado a las nuevas tecnologías, esto no solo es gratificante a nivel profesional, sino también a nivel personal y social.

1.3 Objetivo General

Evaluar técnicas de Deep Learning aplicadas a la estimación adaptativa de canales en redes de quinta generación.

1.4 Objetivos Específicos

- Reconocer los aspectos característicos del canal de comunicaciones en una red de quinta generación junto con las bases teóricas del aprendizaje profundo y las técnicas de estimación de canales.
- Diseñar distintos mecanismos de estimación para canales Línea de Retardo de Pulsación (TDL) de quinta generación basados en técnicas de aprendizaje profundo.
- Evaluar la calidad y rendimiento de los resultados obtenidos en los modelos propuestos usando estadística descriptiva.
- Contrastar el desempeño de los algoritmos desarrollados ante las variaciones de parámetros de simulación en canales TDL con respecto a un escenario ideal.

1.5 Alcances y limitaciones

Se propone evaluar el desempeño que tienen los modelos de aprendizaje profundo en los problemas asociados con la estimación de canales 5G, para ello es necesario definir algunos parámetros e

hiper-parámetros en el entrenamiento de los modelos de redes neuronales profundas, con el fin de evitar sobre-ajustes o predicciones erróneas. Un aspecto considerado está relacionado con la propuesta de solución, ya que se requiere recrear el canal de comunicaciones desde la ubicación de las señales piloto en la rejilla de recursos para luego, realizar un proceso de interpolación que permita recrear las condiciones del canal con alta precisión, y es bajo esta situación, donde se emplean los modelos de aprendizaje profundo, porque se requiere verificar si son soluciones óptimas con la simulación en MATLAB. Para comprobar la correcta generalización y comparar los rendimientos, se evalúa el desempeño de las técnicas empleadas mediante variaciones en las condiciones del canal, como la relación señal a ruido y la velocidad del terminal. Por último se obtienen métricas tales como Error Cuadrático Medio (MSE), Magnitud del Vector Error (EVM), Tasa de error de bit (BER) y el tiempo de estimación.

El proyecto asume ciertos aspectos tales como: la cantidad de datos, ya que esto influye directamente en la capacidad de generalización y eficacia de los algoritmos planteados, es por ello que se emplean únicamente los canales de tipo TDL cuyas aplicaciones en el marco de la quinta generación están más relacionados con sistemas Una Entrada Una Salida (SISO). Otra condición que se maneja tiene que ver con la selección de la numerología OFDM, ya que esta influye la rejilla de recursos, para ello se opta por emplear una de las numerologías en FR1, ya que esta banda de operación es la que se encuentra desplegada en diversas ciudades, porque tiene un equilibrio entre cobertura y la capacidad de datos en zonas urbanas. Finalmente, otra cuestión es la complejidad de los modelos, ya que pueden suponer tiempos de entrenamiento y evaluación prolongados, por lo que existe cierta limitación en la carga computacional, por ello, el trabajo está pensado en desarrollarse únicamente en simulación con el software de MATLAB.

1.6 Estructura de la memoria

Una vez comprendido el panorama global sobre la importancia del despliegue de los sistemas 5G como tecnología que destaca por la optimización de los recursos disponibles en el espectro electromagnético, las altas velocidades de transmisión y la baja latencia, se pasa a describir el contenido de cada capítulo que compone este documento.

En el *capítulo 2* se presentan los fundamentos teóricos detrás de los sistemas de 5G, haciendo

una comparativa con sus predecesoras. Además se incluyen las especificaciones dadas por Proyecto Asociación de Tercera Generación (3GPP) sobre la capa física del sistema de comunicaciones móviles NR, con términos tales como las formas de onda, la numerología y los recursos físicos que se emplean para establecer la comunicación ya sea en el enlace de subida o en el enlace de bajada.

En el *capítulo 3* se presenta una descripción de los fenómenos inherentes en los canales de comunicaciones: ruido aditivo, desvanecimiento y dispersión. Adicionalmente, se introduce el concepto del canal variante en el tiempo y los parámetros que se usan para desarrollar modelos estadísticos que sean una representación aproximada del canal real. Se hace énfasis en el modelo de canal a nivel de enlace TDL, el cual replica un entorno de ecos ya que la señal recibida se compone de la combinación lineal de los retardos de la señal transmitida. Finalmente se explican los conceptos de estimación de canal y de ecualización y las métricas: EVM y BER que permiten evaluar y cuantificar el desempeño del sistema de comunicaciones.

En el *capítulo 4* se explica la teoría del aprendizaje profundo y de su diferenciación con el aprendizaje de máquina y la inteligencia artificial. Se profundiza en el concepto de las redes neuronales artificiales y los parámetros que se tienen en cuenta en el diseño y configuración, como los hiper-parámetros, funciones de activación, función de error y optimizadores. Por último, se muestra la taxonomía del aprendizaje profundo en donde se encuentran algunas arquitecturas como el perceptrón multicapa, las redes neuronales convolucionales y las redes neuronales recurrentes.

En el *capítulo 5* se hace una descripción detallada de los modelos principales del aprendizaje profundo: MLP, CNN y RNN, explicando su funcionamiento y los hiper-parámetros que son necesarios tener en cuenta para que en el entrenamiento y posterior validación se tengan respuestas que se asemejen a las etiquetas del conjunto de datos. Por último, se describen de manera general arquitecturas derivadas de las CNN, como lo son las redes *autoencoder* y las redes *denoising*.

En el *capítulo 6* se presentan algunas literaturas relacionadas con la estimación de canales de comunicaciones. Se parte de los dos tipos de estimación: con pilotos o a ciegas y se presentan algunas soluciones que se han abordado que van desde el uso de procesos matemáticos como las transformadas geométricas o de Fourier hasta contemplar usar modelos estadísticos para inferir las condiciones del canal. Adicionalmente, se incluyen referencias sobre el uso de técnicas de aprendizaje

profundo, en la que se usan modelos como el MLP o las redes bidireccionales LSTM.

En el *capítulo 7* se detalla el procedimiento realizado para la construcción de los diferentes modelos neuronales, el desarrollo se aborda cono 3 partes: la configuración del sistema de comunicaciones 5G y los bloques de transmisor, canal TDL y receptor; enseguida, se hace la descripción de la forma en la que se generan los datos de entrenamiento para las ANN, para ello, es necesario tener en cuenta diversos escenarios que se pueden presentar en un proceso de comunicación, como el perfil del retardo, la dispersión, el ruido y la velocidad del terminal móvil; por último, se diseñan los modelos neuronales usando CNN estándar, autoencoder y red *denoising*, en esta última etapa se hacen variaciones a los modelos para comprobar la calidad de la estimación de canal variando las condiciones del medio de transmisión.

En el *capítulo 8* se evalúa el rendimiento individual de cada modelo y el desempeño general que estos tienen dentro de un entorno de sistemas de comunicaciones basados en 5G a partir de tres etapas: en la primera se realiza la comparativa de los modelos en términos de calidad de predicción y de latencia con los datos de evaluación; en la segunda etapa se evalúan los estimadores ANN dentro del sistema propuesto, manteniendo algunas condiciones fijas en el canal y variando los niveles de ruido para obtener las métricas de MSE, BER, EVM y tiempo de estimación; por último, se seleccionan los mejores modelos y se contrastan los rendimientos en diversos escenarios, en los que se varía el tipo de canal, la modulación aplicada a los bits de transmisión, la velocidad del usuario y los valores de SNR.

Finalmente, en el *capítulo 9* se redactan las conclusiones sobre los resultados obtenidos, las futuras líneas de investigación y los aportes hechos con la propuesta de solución aplicada en la estimación de canal.

Fundamentos Teóricos de 5G

Índice

2.1	Generalidades del 5G	8
2.1.1	Evolución en las comunicaciones móviles	9
2.1.2	Retos actuales	11
2.1.3	Ventajas	12
2.2	Capa física	13
2.2.1	Bandas de operación	13
2.2.2	Formas de onda	14
2.2.3	Prefijo cíclico	16
2.2.4	Numerología OFDM	16
2.3	Recursos físicos	18
2.4	Canales físicos	19
2.5	Señales físicas	20
2.5.1	En enlace de bajada	21
2.5.2	En enlace de subida	22
2.6	Modulaciones soportadas por 5G	22
2.7	Estructuras de transmisión	23
2.7.1	En el dominio del tiempo	23
2.7.2	En el dominio de la frecuencia	24

2.1 Generalidades del 5G

5G corresponde a un notable avance en las tecnologías de comunicaciones móviles, por medio de esta se pueden optimizar los recursos que están disponibles en el espectro electromagnético, así como mejorar la calidad, seguridad y confiabilidad de los servicios de telecomunicaciones. En la

actualidad, la alta demanda de usuarios ha acrecentado inconvenientes asociados al rendimiento de las redes, tener varios dispositivos conectados en una única red supone problemas de velocidad y un deterioro constante de la calidad del servicio, razón por la cual, la actual 4G se ve superada por los nuevos retos y exigencias de los usuarios. Bajo esta premisa es donde el 5G adquiere una connotación vital en el mundo moderno, ya que se busca llegar a un mundo completamente conectado, sacándole provecho a las nuevas tecnologías que incorpora esta última generación, permitiendo que la capacidad y el tráfico en las redes sean manejadas de una forma confiable para garantizar la conexión entre los usuarios.

2.1.1 Evolución en las comunicaciones móviles

La evolución de las comunicaciones móviles ha supuesto un notorio avance en la forma de conectar las personas alrededor del mundo, cada generación ha traído consigo características y aplicaciones que van cambiando según las necesidades de los usuarios, se ha pasado de transmitir voz a través de circuitos eléctricos a transmitir contenido multimedia en tiempo real digitalizando y enviando la información sobre el protocolo de Internet. Las generaciones de comunicaciones móviles son [5] [6]:

- **Primera generación (1G - AMPS):** Esta tecnología surgió en la década de los 80s, en esta los sistemas eran completamente analógicos y la aplicación principal era la transmisión de voz. Para el acceso múltiple usaba la técnica Acceso Múltiple por División de Frecuencias (FDMA) lo que implica que las frecuencias en Enlace de Bajada (D/L) y Enlace de Subida (U/L) son diferentes, adicional con esa multiplexación, se manejaba un ancho de banda de 30 kHz.
- **Segunda Generación (2G - GSM):** En la década de los 90s, ante la alta demanda de los usuarios, se necesitaron nuevas técnicas para manejar de forma óptima el espectro disponible. Bajo esta premisa, esta segunda generación se orientaba en la transmisión digital de voz y datos (mensajes de texto cortos o SMS). La técnica de acceso se basaba en la división de tiempos Acceso Múltiple por División de Tiempos (TDMA) y FDMA, por lo que había una división en el dominio del tiempo y de la frecuencia para mantener una comunicación diferenciada para los enlaces de bajada y de subida; además se empleaba un ancho de banda de 200 kHz.
- **Tercera Generación (3G - UMTS):** Surgió debido a las limitaciones de capacidad que presentaba la generación predecesora, la demanda de contenido multimedia y las bajas velocidades de transmisión de datos, fue por ello que la principal aplicación de esta generación

era la de incorporar conmutación de circuitos y de paquetes para optimizar la calidad del servicio. Las técnicas de acceso empleadas eran FDMA y Acceso Múltiple por División de Códigos (CDMA), esta última implica que existe una asignación de códigos ortogonales para controlar el acceso al canal de comunicaciones. El ancho de banda estaba asignado entre 5 MHz y 20 MHz.

- **Cuarta Generación (4G - LTE):** La tecnología LTE está basada en una arquitectura de conmutación de paquetes sobre el protocolo IP, lo cual supone un uso eficiente del espectro electromagnético y de mayor flexibilidad a la hora de asignar los recursos [7]. El acceso múltiple varía en función de si el enlace es descendente o ascendente, ya que emplea técnicas como FDMA Ortogonal (OFDMA) o FDMA de Única Portadora (CS FDMA) respectivamente; el uso de este tipo de técnicas permite mitigar los efectos generados por el canal de comunicaciones, como es la atenuación en frecuencias altas, la interferencia de banda estrecha y el desvanecimiento selectivo de las multitrayectorias, permitiendo reducir la complejidad en el proceso de ecualización. Se mantienen los valores del ancho de banda de la generación anterior, pero se incluye un ancho opcional de 40 MHz.
- **Quinta Generación (5G - NR):** Esta nueva tecnología móvil surge con el objetivo de suplir las necesidades de la sociedad moderna en el ámbito de las telecomunicaciones [8], ofreciendo velocidades de datos ultra rápidas, menores latencias y una gran capacidad de conectar varios dispositivos en simultáneo, sin afectar la calidad del servicio. Utiliza OFDMA para mejorar la eficiencia espectral, permitiendo mejores conectividades para aplicaciones tales como Internet de las Cosas (IoT) o realidad virtual.

En la siguiente tabla, se resumen algunas de las características principales de cada una de las generaciones de comunicaciones móviles.

Generación	Velocidad de Bajada (Teórica)	Latencia (ms)	Técnica de acceso múltiple (D/L)	Aplicaciones
1G - AMPS	2 kbps	N/A	FDMA	Voz analógica
2G - GSM	384 kbps	629	FDMA - TDMA	Digitalización de la voz y SMS
3G - UMTS	56 Mbps	212	FDMA - CDMA	Contenido multimedia Acceso a internet
4G - LTE	1 Gbps	60 - 98	OFDMA	Streaming
5G - NR	10 Gbps	<1	OFDMA	Internet de las cosas Ciudades inteligentes

Cuadro 2.1: Evolución de las características en cada generación de las redes móviles [8]

De acuerdo con la información resumida en la tabla 2.1 para las características de cada generación de las redes móviles, es importante, tener en cuenta que a medida que evoluciona esta tecnología se presta una mejor calidad del servicio, esto se ve reflejado en las altas tasas de velocidad y la baja latencia, que resulta crítico para entornos donde existe una masificación de usuarios conectados a una red.

2.1.2 Retos actuales

Las redes 4G fueron pensadas para que el acceso a la red fuera por medio de un *Mobile Broadband*, es decir, que se puede obtener internet desde cualquier lugar y momento siempre y cuando se disponga de una cobertura móvil, sin embargo, con el 5G, se da la libertad de ver nuevos horizontes [9]. Esto abre una gama de posibilidades y de aplicaciones que pueden mejorar la calidad de las comunicaciones. Los retos a los que se debe afrontar esta nueva tecnología van encaminados hacia un mundo conectado, donde tanto la industria como la agricultura sean pilares y grandes centros de información. Algunos de los retos que se tienen en cuenta a la hora de pensar en las soluciones de las redes 5G son [10]:

- **Banda ancha móvil mejorada:** Corresponde al manejo de la información a través de la red de forma confiable, segura y rápida, con velocidades en conexiones inalámbricas de hasta 10 Gbps entregando contenido a diferentes dispositivos con la misma velocidad, sin perder la calidad del servicio.
- **Internet de las Cosas masivo:** El IoT resulta un pilar fundamental en un mundo conectado y

moderno, lo que se logra con esto es que muchos dispositivos estén conectados al mismo tiempo realizando diferentes tareas. Con el 5G se desea que la cantidad de dispositivos conectados a una red se incremente hasta 100 veces, sin perder velocidades y ofreciendo confiabilidad en las conexiones. Un ejemplo es que se pueda conectar un millón de dispositivos entre sensores, electrodomésticos u otros aparatos electrónicos en un área de 1 km^2 .

- **Comunicación crítica de las máquinas:** Por último, se busca que con la incorporación de la tecnología 5G se puedan tener latencias inferiores a 1 ms con tasas de error inferiores a $\times 10^{-9}$ [10]. Esto, sin duda, presenta confiabilidad y seguridad en diferentes ámbitos como la automatización industrial.

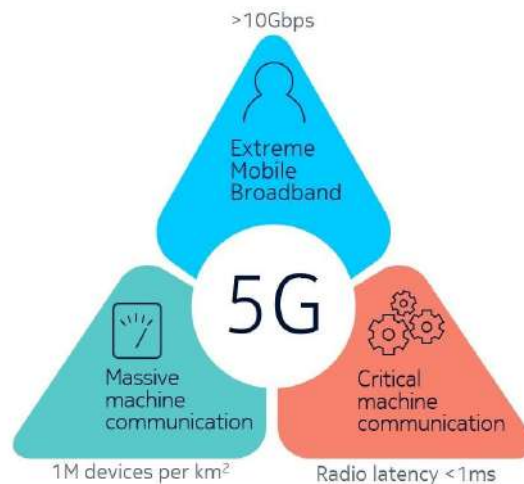


Figura 2.1: Triángulo de los retos del 5G [10]

2.1.3 Ventajas

En [11], se puede ver que el 5G presenta una notable evolución con respecto a su predecesor 4G, entre las principales novedades que trae se encuentran:

- Mejoras en el ancho de banda, lo que permite que el volumen de la información y la capacidad de la red aumenten.
- Bajas latencias, que es fundamental en la transmisión y recepción de datos en tiempo real.
- Un aumento notable en la cantidad de dispositivos que pueden estar conectados en simultáneo a la red, sin degradar la calidad de la conexión.
- Impulso a otras tecnologías, como lo puede ser el Internet de las Cosas, la computación en

la nube y la incorporación de aprendizaje profundo en las redes, para una administración inteligente.

2.2 Capa física

Para los enlaces de bajada, Telecomunicaciones Móviles Internacional (IMT), considera que se debe tener: tasas pico de bajada cercanas a los 20 Gbps, una tasa de bits experimentada por el usuario alrededor de 100 Mbps, una latencia inferior a 1 ms y fijar una movilidad del usuario máxima de 500 km/h [12]. Este tipo de requisitos, son cumplidos con los enlaces de fibra óptica por el ancho de banda y la velocidad de propagación en el medio, sin embargo, al hablar de las comunicaciones inalámbricas es necesario recurrir a eficiencias espectrales superiores a las obtenidas con LTE Avanzado (LTE-A), así mismo es imprescindible proveer soluciones con tecnologías avanzadas, para dotar al sistema de comunicaciones de cierta robustez, para que la información pueda ser transmitida de forma eficiente y confiable en entornos dinámicos.

Teniendo en cuenta los requisitos definidos para el 5G, es necesario considerar aspectos relevantes para la correcta operación y funcionamiento de esta nueva generación de comunicaciones móviles, como, por ejemplo: la banda de frecuencias de operación, las numerologías definidas, las modulaciones soportadas e incluso los modelos de canales y de señales físicas con las cuales se pueden realizar varios procesos como la estimación de canal, sincronización o ecualización. Es por ello que el 3GPP define algunos aspectos clave que son descritos a continuación.

2.2.1 Bandas de operación

La quinta generación tiene un rango de frecuencias más amplio comparada con su predecesora, estas frecuencias de operación se dividen en dos bandas: Rango de frecuencias 1 (FR1) para referirse a las frecuencias inferiores a 6 GHz y Rango de frecuencias 2 (FR2) para las frecuencias superiores a 6 GHz. Los correspondientes rangos de frecuencia para FR1 y FR2 son de (450-6000) MHz y (2425052600) MHz respectivamente [13]. Para cada banda de operación, tanto las EB como las UM, soportan cierto número de portadoras, este parámetro varía de acuerdo al ancho de banda de la portadora y a la capacidad del terminal móvil. La siguiente tabla resume algunos aspectos relevantes de cada rango de frecuencias:

Rango de frecuencias	Ancho de banda de la portadora (MHz)	Características	Aplicaciones
FR1	5, 10, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100	Ofrece un mayor alcance y mejor capacidad de penetración a través de obstáculos naturales o artificiales. Menos susceptible a la absorción debida a la lluvia y las condiciones climáticas	Está pensada para desplegarse en áreas urbanas y suburbanas.
FR2	50, 100, 200, 400	Tiene un ancho de banda alta, comparada con FR1, lo que permite manejar tasas de transferencia de datos ultrarrápidas. Tienen un alcance más limitado y está pensada en un despliegue a partir de celdas pequeñas, con mayor densidad de estaciones base.	Áreas urbanas densas y con alta demanda de capacidad.

Cuadro 2.2: Generalidades sobre los rangos de operación en 5G.

2.2.2 Formas de onda

3GPP acordó mantener OFDM con Prefijo Cíclico (CP-OFDM) tanto en U/L como en D/L (en 4G se empleaba únicamente en D/L) y también se dispone la posibilidad de usar OFDM con Transformada Discreta de Fourier (DFT-OFDM) en U/L, todo esto permite que los dispositivos mantengan una baja complejidad y, por ende, un bajo costo de hardware, sin embargo, dadas las condiciones y las funcionalidades de la quinta generación, es necesario añadir varias numerologías que se adapten a las condiciones del servicio, las frecuencias de operación y los escenarios de despliegue [14].

Multiplexación por División de Frecuencias Ortogonales (OFDM) es una técnica de transmisión multiportadora que es ampliamente usada en la comunicación digital de banda ancha, teniendo aplicaciones en redes inalámbricas, sistemas cableados de radiodifusión o comunicaciones móviles [7]. La popularidad que tiene OFDM está asociada con la eficiencia espectral que maneja al hacer la transmisión paralela de datos, además cuenta con resistencia a los problemas de desvanecimiento selectivo, inmunidad a la interferencia entre subportadoras, flexibilidad, adaptabilidad y es altamente compatible con diferentes técnicas de modulación como Modulación por Amplitud de Cuadratura (M-QAM) o Modulación por Desplazamiento de Fase (PSK).

El funcionamiento de OFDM está ligado con el mapeo de símbolos (información modulada con técnicas M-QAM o PSK) sobre un conjunto de M portadoras [15], cada portadora está separada entre sí con un espaciado Δf óptimo de manera que la duración T_s de cada símbolo es el inverso del espaciado. Esto implica, que el ancho de banda ocupado por una sola portadora pasa a ser ocupado

por un conjunto de subportadoras, como consecuencia de ello se presentan notables mejoras en la eficiencia espectral. Entre cada símbolo se suele recurrir a usar un intervalo de guarda, que es una porción de tiempo que se usa para mitigar la dispersión en el canal y los retardos de llegada en el receptor debido a las condiciones del canal.

La señal banda base que se genera con OFDM, es descrita en el dominio del tiempo con la siguiente expresión:

$$X(t) = \sum_{n=\frac{N}{2}}^{\frac{N}{2}-1} X_s(t) \exp(j 2\pi n \Delta f t) \quad (2.1)$$

En donde: $X_s(t)$ hace referencia a la información que contiene cada uno de los símbolos, Δf es el espaciado entre las subportadoras y N es el número de subportadoras. Un aspecto que se tiene en cuenta en el lado del receptor tiene que ver con el ruido $\eta(t)$ que es generado en el medio de transmisión dadas las condiciones en las que se puede encontrar el canal de comunicaciones, usualmente se emplea el modelo de Ruido Blanco Aditivo Gaussiano (AWGN), ya que asume que el ruido agregado a una señal es de naturaleza aleatoria, tiene una densidad espectral de potencia constante en todas las frecuencias y sigue una distribución normal.

El espectro de una señal que ha sido modulada con OFDM se muestra a continuación:

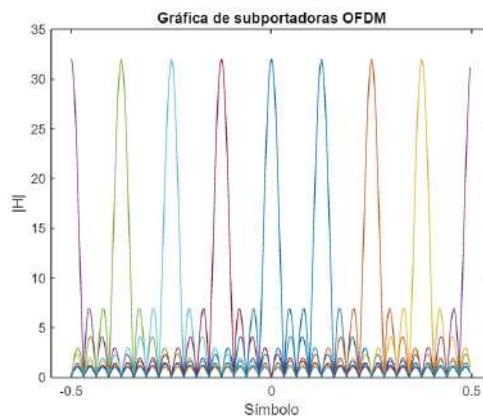


Figura 2.2: Espectro de una señal OFDM (Autoría propia)

Entre las ventajas de la técnica de transmisión OFDM está la baja complejidad, ya que en la práctica se trabaja en el dominio discreto del tiempo; en el transmisor se usa la transformada

discreta inversa de Fourier y en el receptor la transformada discreta. La complejidad del sistema se puede reducir aún más, cuando se emplean los algoritmos de la transformadas rápidas de Fourier, tanto para transmisión y como recepción [15]. Así mismo, con esta técnica se maneja el ancho de banda de una forma más eficiente.

2.2.3 Prefijo cíclico

Otro aspecto fundamental dentro de OFDM corresponde al uso del Prefijo Cíclico (CP) en cambio del intervalo de guarda, este consiste en realizar una copia de las últimas muestras de un símbolo OFDM, la cual es agregada en el inicio del siguiente símbolo antes de la transmisión. La técnica facilita la recepción de la señal y su tratamiento para recuperar la información enviada, ya que el receptor R_x tiene la posibilidad de mitigar los efectos asociados a las interferencias intersimbólicas y entre las portadoras, para recuperar la señal original de forma fiable, superando los problemas de reflexión y retardos en el canal de comunicaciones [15]. Se considera también, que si la duración del CP es superior a la Dispersión del Retardo (DS), la forma de onda no se ve afectada por la interferencia intersimbólica [16].

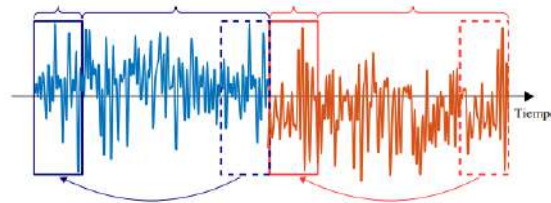


Figura 2.3: Construcción de un símbolo OFDM con la inserción del prefijo cíclico [15]

Un aspecto que se maneja dentro de la tecnología enmarcada en 5G esta relacionada con el manejo de una extensión del prefijo cíclico, el cual varía en función del Espaciado entre Subportadoras (SCS), asociado con la numerología μ , lo cual dota de una mayor capacidad de adaptabilidad y flexibilidad a los entornos de comunicaciones en esta generación.

2.2.4 Numerología OFDM

La numerología corresponde a la configuración de los parámetros de la portadora, logrando mayor flexibilidad comparada con LTE, porque se definen aspectos como: separación de portadoras o el tipo de prefijo cíclico (normal o extendido). En [13] se determina que el factor clave a considerar

para la selección del SCS es el efecto Doppler, ligado con la velocidad del terminal móvil, que puede llegar a ser de 500 km/h. Este fenómeno ocasionado por el desplazamiento de la UM genera cambios en la frecuencia de la portadora f_0 .

Considerando lo mencionado anteriormente y el amplio rango de frecuencias de operación que maneja 5G, es necesario añadir más espaciados Δf entre las subportadoras, que logren dotar a los sistemas de quinta generación de una mayor robustez y flexibilidad. Entre mayor sea el SCS, menor es el tiempo de duración de cada símbolo, lo que resulta útil para la transmisión con altas tasas de velocidad, siendo importante en aplicaciones como control remoto o aquellas en las que sea crítico una transmisión en tiempo real.

La numerología OFDM se denota con μ y de acuerdo con el estándar propuesto por 3GPP se manejan 5 valores de SCS, los cuales se generan a partir de la expresión:

$$SCS = 2^\mu \cdot 15 \text{ kHz} \quad \text{con} \quad \mu = 0, 1, \dots, 4 \quad (2.2)$$

Además se propone dar uso al CP extendido cuando se maneja $\mu = 2$ o en otros términos, con un SCS de 60 kHz. Esta configuración implica un aumento en la duración del tiempo t_{CP} en la formación de las subtramas de la transmisión, lo cual permite mejorar la calidad en la recepción de la información en entornos donde se presenten mayores dispersiones de retardo, es decir, en los que hayan un mayor número de obstáculos que retarden la propagación de la señal enviada.

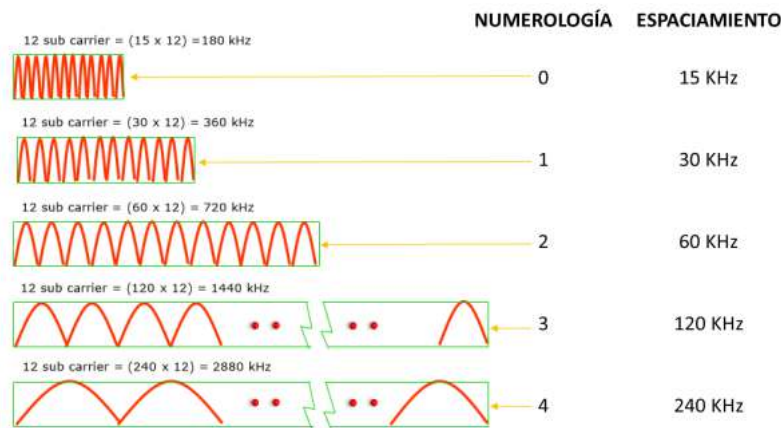


Figura 2.4: Numerología OFDM para 5G [16]

Como se puede evidenciar en la figura (2.4), la variación en el SCS afecta el ancho de banda,

ya que en 5G se define que los bloques de recursos se componen de 12 subportadoras, por lo que existe una relación $BW = RB \times SCS$. El aumento en el orden de la numerología permite disponer de una serie de ventajas entre las cuales destacan: mayor velocidad en la transferencia de datos, mejoras en la calidad de la transmisión con el uso de esquemas de modulación como 64QAM o 256QAM y la reducción de la latencia.

En la siguiente tabla se listan algunas características de la numerología en 5G:

Numerología	SCS (kHz)	T_s (μs)	CP	T_{CP} (μs)	Número de símbolos
0	15	66,666	Normal	4,69	14
1	30	33,333	Normal	2,34	14
2	60	16,666	Normal	1,17	14
			Extendido	4,16	12
3	120	8,333	Normal	0,58	14
4	240	4,166	Normal	0,29	14

Cuadro 2.3: Características de las numerologías de 5G [8] [14]

2.3 Recursos físicos

Se refiere a los elementos fundamentales de la red que se asignan para la transmisión de información a través del medio, estos recursos están organizados y gestionados de manera tal que se pueda hacer una comunicación confiable y eficiente entre el transmisor y el receptor. De acuerdo con 3GPP [17] se listan:

- **Puertos de antena:** Son las interfaces que conectan a las antenas con el equipo transmisor o receptor. En los entornos Múltiples Entradas Múltiples Salidas (MIMO), se suele encontrar un amplio número de puertos dada las altas capacidades que soportan este tipo de antenas.
- **Elemento de Recurso (RE):** Es una subportadora que esta asignada a un símbolo OFDM, es cada elemento que se encuentra en la rejilla de recursos, que tiene asociado un puerto de antena p y una numerología μ . Cada RE tiene una coordenada dentro de la cuadrícula de recursos de la forma $(k, l)_{p, \mu}$, en donde k es el índice de la subportadora y l es el símbolo OFDM asociado.

- **Bloque de Recursos (RB):** El bloque de recursos corresponde al conjunto de 12 subportadoras en el dominio de la frecuencia.
- **Cuadrícula de recursos:** La rejilla de recursos es una estructura fundamental que organiza los recursos físicos, teniendo el índice de las portadoras en el dominio de la frecuencia y el símbolo OFDM en el dominio temporal. Este elemento es fundamental para la asignación de recursos en el sistema de comunicaciones, ya que a partir de una estimación correcta del canal es posible que el usuario final sea capaz de recibir correctamente la información que le fue enviada.

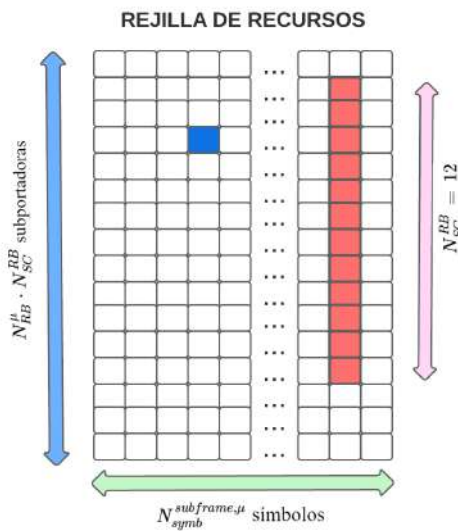


Figura 2.5: Rejilla de recursos: (azul) RE, (rojo) RB. (Autoría propia)

2.4 Canales físicos

La información que es transmitida en los enlaces de bajada y de subida se hacen en diferentes tipos de canales físicos. Los canales físicos se pueden definir como los recursos usados en la transmisión de datos generados en las capas superiores para D/L y para U/L, cada uno tiene diferencias según el sentido de transmisión y las características de la información enviada. Algunos tipos de canales físicos son [7]:

- **PBCH (Physical Broadcast Channel):** Este canal se usa principalmente para proporcionarle toda la información de la estación base a la unidad móvil como por ejemplo la identidad de la celda, la configuración de la red y otros parámetros importantes. Además, este canal cuenta

con el soporte de establecer la sincronización inicial de los dispositivos tanto en el dominio de frecuencia como el de tiempo.

- **Canal Físico Compartido de Enlace de Bajada (PDSCH):** Se usa para la transmisión de datos desde la estación base hasta el terminal móvil. Contiene los datos del usuario (mensajes de texto, imágenes o vídeos), mensajes de control que provienen de las capas superiores, bloques de información del sistema y mensajes de paginación que alertan al terminal móvil para realizar una acción en concreto. La importancia de este canal físico se relaciona con la flexibilidad en la asignación de recursos, tanto de frecuencia como de tiempo, ya que esto varía según la condición actual del canal de comunicaciones, por eso es crítica emplear técnicas de estimación de canal precisas, para que se puedan hacer las asignaciones de recursos de manera eficiente y rápida con los dispositivos de la red. Otro aspecto importante es que soporta varios esquemas de modulación y de codificación.
- **PDCCH (Physical Downlink Control Channel):** Se usa para la planificación en la asignación de los recursos en enlace de bajada dependiendo de las condiciones del enlace, el esquema de modulación empleado es Modulación por Desplazamiento de Fase en Cuadratura (QPSK), dado que este esquema es capaz de soportar condiciones adversas como los altos niveles de ruido en el canal inalámbrico y efectos asociados con la multitrayectoria y el desvanecimiento.
- **PUSCH (Physical Uplink Shared Channel):** Este canal físico funciona en el enlace de subida, por lo cual es el encargado de enviar los datos desde el terminal móvil hasta la estación base.
- **PUCCH (Physical Uplink Control Channel):** Tiene la función de informar a la estación base sobre peticiones de asignación dinámica de recursos, la confirmación de la recepción de datos y el control en el enlace de subida.

2.5 Señales físicas

Las señales físicas corresponden al tipo de señal que se usa como referencia o piloto y que cumple un rol fundamental en procesos para demodulación, estimación de canal o sincronización tanto en el dominio de la frecuencia como en el dominio del tiempo. Este tipo de señales se tienen tanto en U/L como para D/L [7].

2.5.1 En enlace de bajada

- **Señal de Referencia de Demodulación (DM-RS)** Este tipo de señal tiene relevancia en procesos asociados con la estimación de canales, ya que le proporciona al receptor información sobre cómo debe interpretar y decodificar la señal $r(t)$ que recibe, para recuperar la información transmitida a través del canal de comunicaciones. Con el DM-RS es posible obtener información sobre el canal, para tener una asignación de recursos eficiente que garantice una óptima comunicación entre el transmisor y el receptor.

Existen dos tipos de señales DM-RS, tipo 1 y tipo 2 y tienen que ver con la densidad de la señal en el dominio de la frecuencia. El tipo 1 admite 6 señales ortogonales para un símbolo y 12 señales ortogonales con una longitud de dos símbolos. Por otro lado, el tipo 2 admite 4 señales ortogonales (dos grupos de dos subportadoras consecutivas) y 8 señales ortogonales para una longitud de dos símbolos [18].

El aspecto importante en las señales DM-RS es la configuración del mapeo tanto en el canal PDCCH como en el PDSCH, ya que se usan los tipos A y B. Para el primero se ubican las señales piloto en el segundo o tercer símbolo, mientras que el segundo mapeo se usa para inicializar los canales físicos en cualquier otro lugar, por lo que las señales de referencia se alojan en el comienzo de la duración de la transmisión [13].

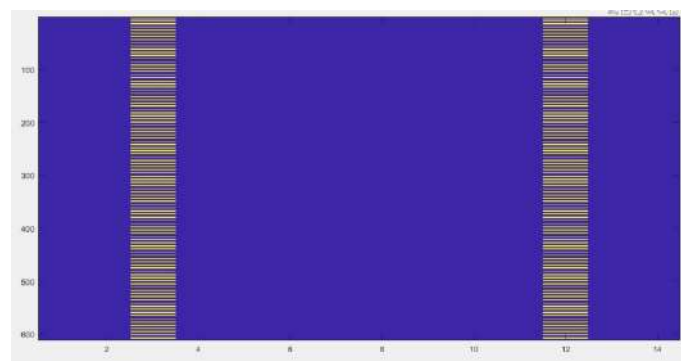


Figura 2.6: Inserción de señales piloto DM-RS en la rejilla de recursos. (Autoría propia)

- **Señales de referencia para el seguimiento de fase (PT RS):** Permite mitigar los efectos de la desviación de frecuencia. Es una señal útil para efectuar procesos de compensación de fase de las señales que fueron transmitidas a través del canal de comunicaciones. con el fin de

disminuir las distorsiones generadas por el mismo.

- **Señal de referencia sobre la información del estado del canal (CSI-RS):** Como su nombre lo indica, permite extraer características acerca del canal, lo cual resulta útil para obtener una estimación del mismo.
- **Señales de sincronización primaria y secundaria (PSS y SSS):** Ambas funcionan en conjunto para establecer la correcta sincronización entre la estación base y la unidad móvil.

2.5.2 En enlace de subida

Para los U/L existen señales iguales a las descritas en el enlace de bajada, tal es el caso de las señales de referencia DM-RS y la señal PT-RS. Sin embargo, en 5G se adiciona otra señal completamente independiente para este enlace, dicha señal física es la señal de sondeo de referencia (SRS) estas se usan para la estimación del canal en enlace de subida y para la sincronización. De acuerdo con [13] la principal función que cumple este tipo de señal física es la de obtener la información de las condiciones del canal para sistemas que utilizan Duplexación por División de Tiempo (TDD) y para obtener un rendimiento excepcional de las antenas MIMO.

2.6 Modulaciones soportadas por 5G

De acuerdo con 3GPP, la tecnología móvil de 5G es capaz de soportar los siguientes esquemas de modulación [18]:

- Modulación por desplazamiento de fase en cuadratura (QPSK).
- Modulación de amplitud en cuadratura de 16 estados (16QAM).
- Modulación de amplitud en cuadratura de 64 estados (64QAM).
- Modulación de amplitud en cuadratura de 256 estados (256QAM).
- Para la transmisión de datos con bajas velocidades en el enlace de subida, 5G es capaz de soportar el esquema de modulación por desplazamiento de fase binaria $\pi/2$ ($\pi/2$ -BPSK)[16].

2.7 Estructuras de transmisión

Las estructuras de transmisión se refiere a la infraestructura física y tecnológica que se usa para el despliegue de los sistemas 5G. A diferencia de su predecesora y debido a la inclusión de nuevas numerologías μ , se dotan a los sistemas de una mayor adaptabilidad a las necesidades de cada usuario final, esto se hace con las variaciones de recursos tanto en el dominio del tiempo como en el dominio de la frecuencia [16].

2.7.1 En el dominio del tiempo

En el dominio del tiempo, la estructura de transmisión se compone de tramas con una duración de 10 ms, cada trama esta dividida en 10 subtramas, cada una presenta una duración de 1 ms. Las subtramas se componen de los Intervalos de Tiempo de Transmisión (TTI) que son la cantidad de símbolos OFDM asociados con cada una de las numerologías (ver tabla 2.3), justamente esta variación en μ es la que permite que en la transmisión se puedan ajustar diferentes TTI, lo cual permite una mayor flexibilidad y adaptabilidad. Así mismo, la variación del TTI permite insertar una mayor cantidad de información en anchos de banda superiores, dando cabida a entornos con baja latencia y velocidades de transmisión superiores a las presentes en 4G.

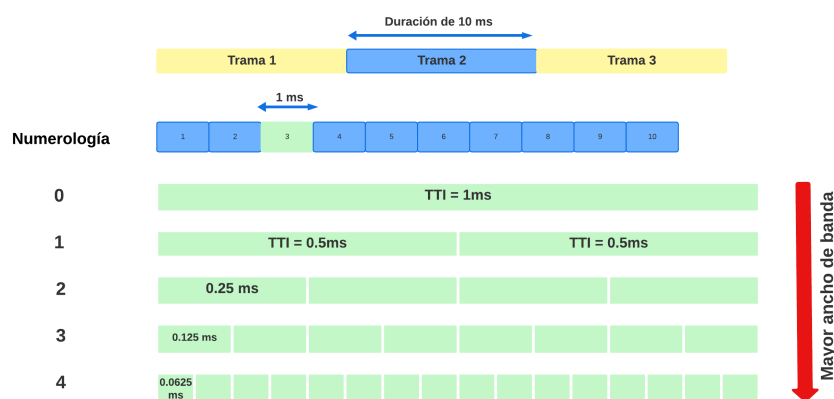


Figura 2.7: Estructura de transmisión en el dominio del tiempo. (Autoría propia)

En 5G es posible tener TTI inferiores a los 14 símbolos que se manejan por defecto, dichas estructuras se conocen como mini-TTI con duraciones de 2, 4 y 7 símbolos OFDM [16].

2.7.2 En el dominio de la frecuencia

En el dominio de la frecuencia, para 5G se maneja el concepto de los bloques de recurso RB, formados por las 12 subportadoras, sin embargo, para añadir flexibilidad en los entornos, se presenta la posibilidad de agrupar los RB en los denominados: Partes de Anchos de Banda (BWP) [16]. Adicional a ello, se realiza un proceso de asignación dinámica de las frecuencias de acuerdo con los requerimientos de velocidad necesarios para cubrir la demanda de las UM conectadas en la red, de manera tal, que se pueda disponer de una mayor capacidad acompañada de una buena calidad del servicio.

Al unir los dominios de la frecuencia y del tiempo, se maneja una rejilla de recursos (figura 2.5), con la cual se asigna un RE con las coordenadas del símbolo OFDM y el índice de la subportadora, esto se traduce, en que dependiendo de la numerología μ se puede hacer una variación en la cantidad de recursos, implicando diferentes velocidades de transmisión y latencias que se acomodan con los requisitos de cada usuario.

3

Canal de comunicaciones

Índice

3.1	Generalidades	25
3.2	Ruido	27
3.3	Desvanecimiento	28
3.3.1	Distribución de Rice (LoS)	29
3.3.2	Distribución de Rayleigh (NLoS)	30
3.4	Canal variante en el tiempo	30
3.5	Características en un canal variante en el tiempo	32
3.5.1	Dispersión	32
3.5.2	Ancho de banda de coherencia	34
3.5.3	Dispersión Doppler	34
3.5.4	Tiempo de coherencia	35
3.6	Modelo de canal TDL	35
3.7	Estimación de canal	40
3.7.1	Estimador por mínimos cuadrados	41
3.7.2	Estimador MMSE	41
3.8	Ecualización de canal	42
3.9	Compensación de fase	45
3.10	Métricas de desempeño	47
3.10.1	Magnitud del Vector de Error	47
3.10.2	Tasa de Error de Bit	48

3.1 Generalidades

El canal de comunicaciones es una parte fundamental para entender la operación, el diseño y el análisis de cualquier sistema de comunicaciones, puesto que es a través de este donde se hace el

envío de la información desde un transmisor T_x a un receptor R_x . La distancia entre el transmisor y el receptor puede ser con Línea de Vista (LoS) e incluso Sin Línea de Vista (NLoS), debido a la aparición de obstáculos naturales (montañas, superficies irregulares) o artificiales (edificios). Así mismo, se presentan otros efectos inherentes al tipo de comunicación: la atenuación, el desvanecimiento y la aparición de ruido aditivo. A diferencia de los canales alámbricos, el tipo de tecnología como la móvil, implica la aparición de comportamientos aleatorios que impiden un análisis predecible de la propagación de la señal transmitida $s(t)$ [19].

De acuerdo con lo descrito anteriormente, el concepto del canal de comunicaciones ideal es aquel en el que la señal transmitida $s(t)$ llega sin ningún cambio al receptor y satisface la expresión:

$$r(t) = \alpha s(t - \tau_0) + \eta(t) \quad (3.1)$$

En donde α es la atenuación en el canal, τ_0 es el retardo y $\eta(t)$ es el ruido aditivo [20]. Dadas las características variables del canal tanto en el dominio del tiempo como de la frecuencia, es necesario definir ciertas parámetros que se ajusten a modelos estadísticos, teniendo en cuenta los fenómenos inherente y las derivaciones ocasionadas por el movimiento del transmisor o del receptor.

El canal puede ser representado a partir del dominio de la frecuencia o el dominio del tiempo, usando la transformada de Fourier y la convolución respectivamente. La respuesta al impulso $h(t)$ representa el modo en el que la energía de una señal transmitida es dispersada en el canal, mientras que la respuesta en frecuencia $H(f)$ permite determinar las frecuencias f en las que la señal es atenuada [20]. Conocer ambas componentes, permite abordar el modelo de los canales de una forma más precisa, tal como se describe en la ecuación 3.1.

Dada la naturaleza aleatoria que tienen los canales en las comunicaciones móviles, resulta erróneo abordar los modelos como si se comportaran de manera estática, por el contrario, los modelos se piensan como un sistema Lineal Variante en el Tiempo (LTV), por lo cual las condiciones y características que este tiene varían en función del tiempo. Con este premisa, es posible tener representaciones altamente complejas que se basan en la geometría del problema y la evolución de los elementos presentes en dicho problema, así como también representaciones usando la estadística para obtener las realizaciones del canal que son equivalentes a los escenarios reales, justamente esta última es la empleada en los entornos de simulación dada su baja complejidad.

De acuerdo con lo descrito anteriormente, a continuación se describen algunos fenómenos presentes en los canales de comunicaciones, como lo son el tipo de ruido y el desvanecimiento. Tras ello se presentan de manera general algunos conceptos detrás de los canales variantes en el tiempo y que repercuten altamente en la calidad de la recepción de las señales $r(t)$ si no se tratan adecuadamente.

3.2 Ruido

El ruido $\eta(t)$ responde a una naturaleza aleatoria, inevitable y generalmente incontrolable que esta presente en los sistemas de comunicaciones, mitigar sus efectos permiten recibir señales de buena calidad, que se asemejen en gran medida a la señal transmitida $s(t)$ [21]. Las fuentes de ruido se pueden considerar como modelos completamente aleatorios, ya que no responden a un comportamiento en particular tanto en amplitud como en fase, sin embargo, dada su relevancia en la calidad y desempeño de los sistemas, en el desarrollo de simulaciones se suele emplear AWGN, que responde a una distribución normal.

El canal AWGN implica la presencia de $\eta(t)$ que posee una densidad espectral de potencia constante en todas las frecuencias de $N_0/2$, tiene una media igual a 0 y una total independencia de la señal de entrada al canal $s(t)$ [22]. La presencia de AWGN implica que la fuente de ruido primaria esta en el receptor o que es una radiación que impacta al receptor sin importar el camino sobre el cual la señal $s(t)$ es enviada [23], esto resulta un gran acercamiento a escenarios reales, donde existen multitrayectorias que se pueden ver afectadas en diferente medida al ruido del canal.

Una forma de cuantificar el impacto que tiene el ruido sobre la señal es a través de la Relación Señal a Ruido (SNR), esta corresponde al cociente entre la potencia de la señal enviada P_s y la potencia de ruido P_η en cualquier punto del sistema de comunicaciones. La expresión es:

$$SNR = \frac{P_s}{P_\eta} \quad \text{expresada en dB como:} \quad SNR[\text{dB}] = 10 \log_{10} \left(\frac{P_s}{P_\eta} \right) \quad (3.2)$$

La SNR proporciona una medida de la calidad de una señal en cualquier punto del sistema de comunicaciones. En el diseño de los sistemas se espera manejar una SNR elevada, lo cual se deriva

de una mayor potencia en la señal enviada, sin embargo, en escenarios reales, mantener una SNR elevada implica mayores costos de implementación puesto que es necesario manejar amplificadores y antenas con una mayor ganancia [21]. Como consecuencia de ello, el ruido debe ser un factor a tener en cuenta en cualquier diseño, ya que su impacto afecta directamente la calidad de la señal recibida.

3.3 Desvanecimiento

El canal en las comunicaciones móviles opera de manera inalámbrica, por lo que las ondas electromagnéticas se propagan a través del espacio libre. La interacción entre las ondas de radio y el medio generan fluctuaciones que afectan la amplitud y la fase de las señales transmitidas, dicho fenómeno se conoce como desvanecimiento y de manera general se pueden agrupar desde dos enfoques: a gran escala (pérdidas de trayectoria y *shadowing*) y a pequeña escala (pérdidas de multi-trayectoria). La siguiente figura permite ilustrar el comportamiento que tiene la potencia recibida teniendo en cuenta el desvanecimiento del canal en función de la distancia del enlace:

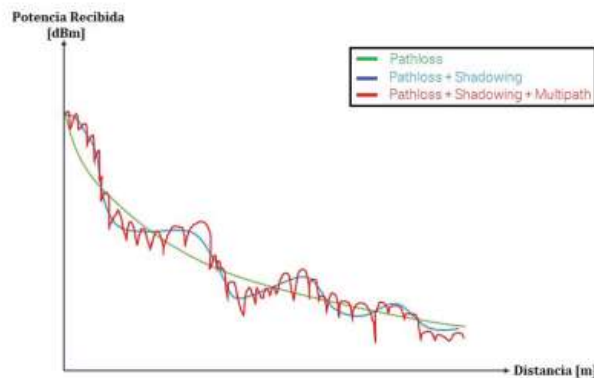


Figura 3.1: Comportamiento de los tipos de desvanecimiento en una gráfica de d vs P_R [24]

El desvanecimiento a gran escala corresponde a cambios del valor medio cuando la distancia d varía significativamente, llegando a ser superiores a varias decenas de longitudes de onda λ [24]. Los fenómenos que se asocian con este desvanecimiento son la reflexión, la difracción y la dispersión. Las pérdidas de trayecto o *path-loss* se refiere a la atenuación o disminución de la potencia de la señal a medida que esta se propaga a través del espacio libre¹ y se puede deber a: la distancia

¹Medio homogéneo, lineal e isótropo que esta libre de cargas y de corrientes eléctricas, por lo cual no existen pérdidas por el efecto Joule. En el espacio libre las ondas electromagnéticas viajan en línea recta y sin atenuación, las únicas pérdidas se deben a la dispersión del frente de onda.

entre la EB y la UM, interferencia electromagnética y la absorción atmosférica. Por otro lado, el *shadowing* surge por la obstrucción en el enlace debido a obstáculos artificiales o naturales durante la propagación de la onda.

El desvanecimiento a pequeña escala ocurre por cambios rápidos en distancias cortas (inferiores a 10λ) [24], esto ocasiona réplicas a lo largo del canal de comunicaciones de la señal $s(t)$ emitida por el transmisor y que llegan a un solo destino en fases y amplitudes diferentes, haciendo que en el receptor se capte una señal formada por la combinación lineal de las señales multi-trayecto.

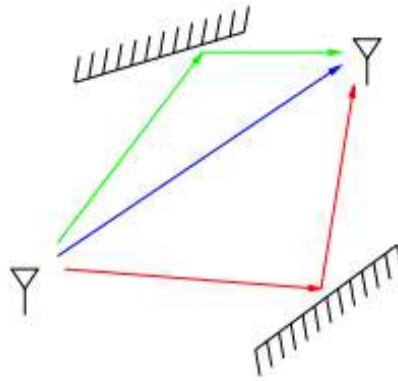


Figura 3.2: Propagación multi-trayectoria [23]

En los canales de comunicaciones en las que se presenten fluctuaciones que varían la amplitud y la fase de la señal transmitida, es común asociarla con algún modelo estadístico que describa el comportamiento de la señal recibida, para ello se emplean las distribuciones de Rice y de Rayleigh, las cuales se presentan en entornos LoS y NLoS respectivamente.

3.3.1 Distribución de Rice (LoS)

En un escenario de línea de vista se emplea la distribución de Rice, en donde el *fading* gaussiano y complejo del canal tiene una media no nula, por lo que tiene una componente dominante y varias componentes difusas o de multitrayecto. Dentro de esta distribución se tiene en cuenta un factor K o factor de Rice, que se expresa como la relación entre la potencia de la componente dominante

sobre la potencia de las componentes difusas [20] y que sigue la expresión:

$$K = \frac{P_{\text{dom}}}{P_{\text{difusa}}} = \frac{V^2}{2\sigma^2} \quad (3.3)$$

En donde V^2 es la envolvente de la componente de media no nula de la gaussiana compleja (componente LoS) y σ^2 es la varianza de las componentes difusas. Cuando el valor de K tiende a cero, quiere decir que las componentes multitrayecto tienen una mayor potencia comparada con la dominante, resultando en un comportamiento similar al de una distribución Rayleigh.

3.3.2 Distribución de Rayleigh (NLoS)

Este tipo de distribución ocurre cuando el factor K es igual a 0, por lo cual, no existe una componente dominante de la onda que es transmitida a través del canal de comunicaciones, esto implica que no hay una línea de vista entre el transmisor y el receptor, por lo cual, existen señales que llegan al receptor con diferentes amplitudes y fases. El desvanecimiento gaussiano y complejo del canal tiene media cero y una varianza idéntica para las componentes real e imaginaria [20].

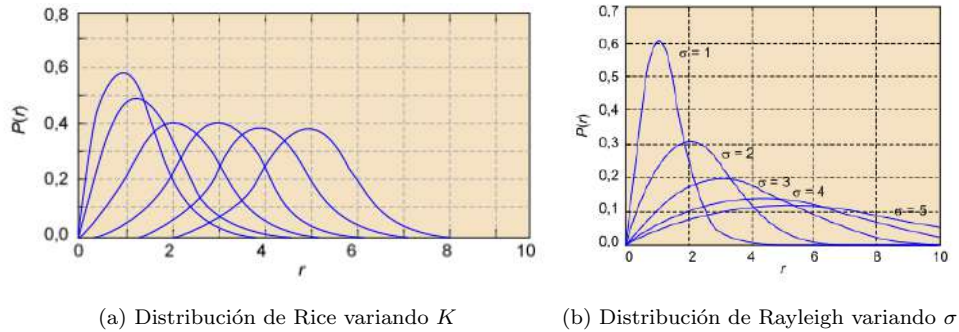


Figura 3.3: Funciones de densidad de probabilidad de la envolvente [20]

3.4 Canal variante en el tiempo

Se define a un canal variante en el tiempo a todo canal en el que las características no permanecen constantes, fenómenos como las fluctuaciones en las atenuaciones, el ruido y la interferencia se modifican en función del tiempo y el espacio. Este tipo de canales son estudiados en las comunicaciones móviles, ya que es común encontrarse con escenarios en donde la unidad móvil se desplaza

por diferentes entornos constantemente y así mismo ocurren afectaciones por los obstáculos móviles y las condiciones atmosféricas y ambientales.

Al incluir las variaciones en el tiempo, la expresión de la ecuación 3.1 se ve modificada de la siguiente forma:

$$r(t) = \alpha(t) s[t - \tau(t)] + \eta(t) \quad (3.4)$$

Esto representa que tanto la atenuación α como el retardo τ se convierten en funciones variables en el tiempo. La respuesta al impulso de estos canales queda caracterizada por la función del canal $h(t, \tau)$, en donde t es la variación temporal y τ la dispersión temporal como consecuencia del multicamino [20]. La figura (3.4) permite ilustrar de una mejor forma el comportamiento que tiene la respuesta al impulso del canal en donde se verifica que t es inherente a la variabilidad del canal y τ esta relacionado con la distorsión de la señal.

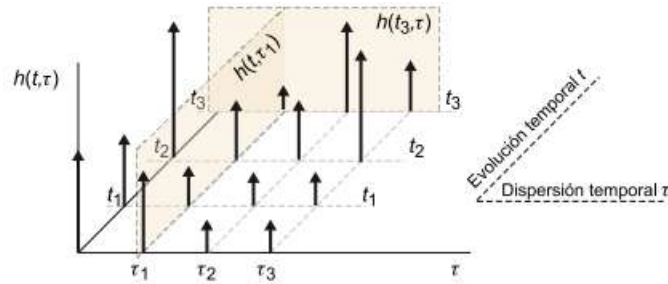


Figura 3.4: Respuesta al impulso de un canal LTV [20]

De acuerdo a lo descrito anteriormente, se aplica la convolución para relacionar la señal de salida del canal $r(t)$ con la señal de entrada $s(t)$ y se obtiene la ecuación (3.5):

$$r(t) = \int_{-\infty}^{\infty} h(t, \tau) s(t - \tau) d\tau + \eta(t) \quad (3.5)$$

Para un escenario en el que hay pérdidas de multitrayecto, es posible describir la respuesta al impulso $h(t, \tau)$ como la combinación lineal de las señales difusas que se propagan a través del espacio libre, tal como se describe en la siguiente expresión [23]:

$$h(t, \tau) = \sum_i \alpha_i(t) \delta(t - \tau_i(t)) \quad (3.6)$$

La figura (3.5) presenta el comportamiento de una señal transmitida a través de un modelo estadístico que representa un canal LTV:

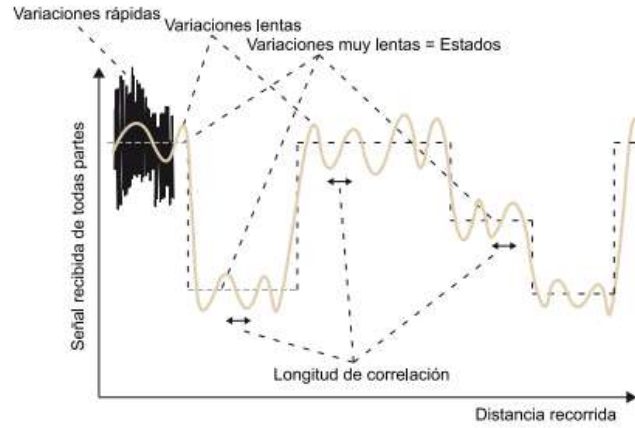


Figura 3.5: Modelo estadístico de un canal LTV [20]

3.5 Características en un canal variante en el tiempo

Teniendo en cuenta los componentes de un canal lineal variante en el tiempo, en esta sección se presentan los parámetros que permiten caracterizar adecuadamente la variabilidad en el tiempo y la dispersión temporal, estas características permiten obtener modelos estadísticos que se asemejen en gran medida a los escenarios reales. Los parámetros principales son: la dispersión del retardo, ancho de banda de coherencia, dispersión Doppler y tiempo de coherencia [20].

3.5.1 Dispersión

La dispersión es el fenómeno mediante el cual la onda propagada a través del medio interactúa con entornos que cuentan con diferentes condiciones, resultando en la generación de señales multitrayectoria que afectan la recepción de la señal $r(t)$. Este parámetro se usa para describir el comportamiento de la variación temporal en términos de τ o del índice k , por lo que tiene una alta incidencia tanto en los entornos LoS como en los NLoS. Este parámetro puede ser estimado a partir de una representación continua $h(t, \tau)$ así como también en espacio discreto de la forma $h[t, k]$, sin embargo, dada la simplicidad de notación en un canal discreto se emplea esta última [20].

Perfil del retardo de la potencia

El perfil del retardo de la potencia o en inglés conocido como *power-delay profile* es una medida que permite conocer la distribución de potencia media de una señal transmitida a través del canal de comunicaciones multitrayecto, para calcularla se emplea la esperanza matemática² con respecto a la variable t de la potencia para cada coeficiente del canal (índice K), se calcula a partir de la siguiente expresión [20]:

$$PDP[k] = E\{|h[t, k]|^2\} \quad (3.7)$$

3GPP define diferentes perfiles de acuerdo al tipo de canal que se emplea para hacer la evaluación en el enlace, en [25] se encuentra que para los canales Cluster Delay Line (CDL) y TDL hay 5 perfiles diferentes, para escenarios donde hay línea de vista o no. Esta información se encuentra en el apartado (3.6) “*Modelo de canal TDL*”.

Dispersión del retardo

La dispersión del retardo o en inglés *delay spread* es la diferencia entre los tiempos de llegada de las componentes dominantes y de las componentes difusas de una señal transmitida a través de un canal multi-camino, así pues, se puede definir como la raíz cuadrada media (desviación estándar) de las reflexiones, ponderadas por su amplitud (perfil de retardo de la potencia), se expresa de la siguiente forma [20]:

$$DS = \sqrt{\frac{\sum_k PDP[k](kT)^2}{\sum_k PDP[k]} - \left(\frac{\sum_k PDP[k](kT)}{\sum_k PDP[k]}\right)^2} \quad (3.8)$$

El *delay spread* afecta la interferencia entre símbolos, ya que al aumentar los tiempos en los que los símbolos llegan al receptor, se puede generar un solapamiento con los símbolos posteriores, haciendo que estos últimos se comporten de manera similar al ruido. De acuerdo con ello, es necesario disponer de modelos que contemplen la dispersión del retardo, así como también, contar con técnicas que mitiguen el impacto de esta interferencia como lo es el prefijo cíclico al inicio de las señales OFDM.

²Es la suma ponderada de todos los posibles valores que puede tomar una variable aleatoria, tomando la probabilidad de ocurrencia. Se representa con $E[X]$

3.5.2 Ancho de banda de coherencia

El ancho de banda de coherencia BW_c corresponde al rango de frecuencias en el cual se puede considerar que el canal se comporta de manera constante, este valor está ligado con la dispersión del retardo, ya que es el inverso proporcional, tal como se muestra en la ecuación (3.9):

$$BW_c = \frac{1}{\beta DS} \quad (3.9)$$

El valor de la constante β se elige de acuerdo con el grado de correlación entre las dos frecuencias extremo del intervalo, por lo cual, varía dependiendo de los escenarios y las condiciones del canal. En algunas literaturas, tal como se describe en [20] si se tiene que una correlación de 0.9 definen un $\beta = 50$, mientras que si la correlación entre las frecuencias es de 0.5 se puede tomar $\beta = 5$, por ello, es necesario estimar este valor a partir de un criterio en concreto, que cumpla con la condición mediante la cual el canal permanece invariante en el tiempo.

3.5.3 Dispersión Doppler

Este tipo de dispersión está asociado con el efecto Doppler, que se define en [26] como el cambio de frecuencia f_0 de una onda propagada debido al movimiento relativo de la fuente respecto al observador. Debido a ello, se considera a la dispersión Doppler como una variación en el dominio de la frecuencia de las componentes fuertes y difusas que se generan en la propagación de la onda mientras la fuente de transmisión se encuentra en movimiento, este escenario se asemeja a un entorno real de comunicaciones móviles, ya que es común que la unidad móvil está desplazándose constantemente sin perder la conexión.

Debido a que la señal original al propagarse por el canal genera un múltiples rayos, cada uno va a llegar al mismo instante de tiempo al receptor, sin embargo, van a tener ángulos de llegada diferentes, puesto que cada uno experimenta variaciones aleatorias en el canal. Lo descrito anteriormente, implica que la señal recibida va a tener una frecuencia que se compone de la dispersión frecuencial de cada rayo, dado por la expresión:

$$f_r = \sum_k f_0 + \frac{v}{c} f_0 \cos(\theta_k) \quad (3.10)$$

En donde v es la velocidad relativa de la fuente respecto al observador, c es la constante de la

velocidad de la luz, f_0 es la frecuencia fundamental de la señal $s(t)$ y θ_k es el ángulo de llegada de cada componente multi-camino generada en el canal de comunicaciones. Así pues, es un escenario en el cual, se generan 3 componentes difusas, la frecuencia f_r va a ser la suma de los 3 tonos de cada rayo: $f_r = f_1 + f_2 + f_3$ [20].

Se define una frecuencia Doppler máxima cuando el ángulo de llegada cumple con $\theta_k = 0$, por lo que: $f_{D_{\max}} = \frac{v}{c} f_0$. Esta relación permite identificar que en escenarios estáticos, $v = 0$ no hay dispersión Doppler, sin embargo, a medida que la velocidad de la fuente aumenta, lo hace también la frecuencia máxima, dificultando la detección y decodificación de la información transmitida en la señal $s(t)$.

3.5.4 Tiempo de coherencia

Este parámetro es similar al descrito en el *ancho de banda de coherencia*, sin embargo, esta enfocado en el dominio temporal, por lo cual se define como la duración de tiempo en la que la respuesta al impulso del canal tiene un comportamiento constante. Es inversamente proporcional a $f_{D_{\max}}$ y se ajusta a la siguiente expresión:

$$t_c = \frac{1}{\gamma f_{D_{\max}}} \quad (3.11)$$

Algunos autores como [20] sugieren emplear la siguiente expresión para calcular el tiempo de coherencia del canal de comunicaciones:

$$t_c = \sqrt{\frac{9}{16\pi}} \frac{1}{f_{D_{\max}}} \quad (3.12)$$

3.6 Modelo de canal TDL

Un canal TDL o *tapped delay line* corresponde a un modelo estadístico con desvanecimiento para evaluar los enlaces de subida o de bajada en la comunicación móvil. En el canal se aplica una técnica de procesamiento de señales mediante la cual se generan retardos variables de la señal transmitida $s(t)$ (cada señal $s(t - \tau_k)$ tiene su propia amplitud y fase) y se extraen en cualquier punto de la línea, por lo cual, la señal recibida se compone de la combinación lineal de la señal transmitida y sus réplicas con retardo τ_k . Los canales TDL permiten simular adecuadamente escenarios con

múltiples ecos (reverberación artificial) [27].

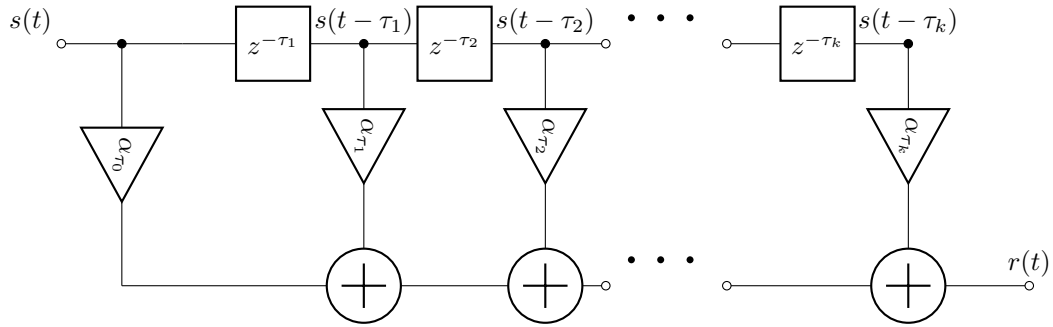


Figura 3.6: Diagrama de bloques de un canal TDL. (Autoría propia)

De acuerdo con la figura (3.6) se puede expresar la señal recibida en el receptor a partir de la siguiente expresión:

$$r(t) = \sum_k \alpha_{\tau_k} s(t - \tau_k) + \eta(t) \quad \text{en donde} \quad k = 0, 1, 2, \dots, n \quad (3.13)$$

El canal TDL según [25] se usa para evaluar el desempeño a nivel de enlace en entornos no MIMO y se especifica que funciona en las frecuencias de 0.5 GHz hasta 100 GHz. Tres perfiles de retardo se usan para caracterizar escenarios sin línea de vista: TDL-A, TDL-B y TDL-C; mientras que se usan los modelos TDL-D y TDL-E cuando hay línea de vista. Todos los perfiles de retardo están modelados a partir de una desvanecimiento con distribución de Rice y de Rayleigh. 3GPP presenta unas tablas que caracterizan cada perfil de retardo [25]:

Tap	Retardo (τ_k) normalizado	Potencia en dB	Distribución del desvanecimiento
1	0.0000	-13.4	Rayleigh
2	0.3819	0	Rayleigh
3	0.4025	-2.2	Rayleigh
4	0.5868	-4	Rayleigh
5	0.4610	-6	Rayleigh
6	0.5375	-8.2	Rayleigh
7	0.6708	-9.9	Rayleigh
8	0.5750	-10.5	Rayleigh
9	0.7618	-7.5	Rayleigh
10	1.5375	-15.9	Rayleigh
11	1.8978	-6.6	Rayleigh
12	2.2242	-16.7	Rayleigh
13	2.1718	-12.4	Rayleigh

Tap	Retardo (τ_k) normalizado	Potencia en dB	Distribución del desvanecimiento
14	2.4942	-15.2	Rayleigh
15	2.5119	-10.8	Rayleigh
16	3.0582	-11.3	Rayleigh
17	4.0810	-12.7	Rayleigh
18	4.4579	-16.2	Rayleigh
19	4.5695	-18.3	Rayleigh
20	4.7966	-18.9	Rayleigh
21	5.0066	-16.6	Rayleigh
22	5.3043	-19.9	Rayleigh
23	9.6586	-29.7	Rayleigh

Cuadro 3.1: Caracterización del perfil de retardo TDL-A

Tap	Retardo (τ_k) normalizado	Potencia en dB	Distribución del desvanecimiento
1	0.0000	0	Rayleigh
2	0.1072	-2.2	Rayleigh
3	0.2155	-4	Rayleigh
4	0.2095	-3.2	Rayleigh
5	0.2870	-9.8	Rayleigh
6	0.2986	-1.2	Rayleigh
7	0.3752	-3.4	Rayleigh
8	0.5055	-5.2	Rayleigh
9	0.3681	-7.6	Rayleigh
10	0.3697	-3	Rayleigh
11	0.5700	-8.9	Rayleigh
12	0.5283	-9	Rayleigh
13	1.1021	-4.8	Rayleigh
14	1.2756	-5.7	Rayleigh
15	1.5474	-7.5	Rayleigh
16	1.7842	-1.9	Rayleigh
17	2.0169	-7.6	Rayleigh
18	2.8294	-12.2	Rayleigh
19	3.0219	-9.8	Rayleigh
20	3.6187	-11.4	Rayleigh
21	4.1067	-14.9	Rayleigh
22	4.2790	-9.2	Rayleigh
23	4.7834	-11.3	Rayleigh

Cuadro 3.2: Caracterización del perfil de retardo TDL-B

Tap	Retardo (τ_k) normalizado	Potencia en dB	Distribución del desvanecimiento
1	0	-4.4	Rayleigh
2	0.2099	-1.2	Rayleigh
3	0.2219	-3.5	Rayleigh
4	0.2329	-5.2	Rayleigh
5	0.2176	-2.5	Rayleigh
6	0.6366	0	Rayleigh
7	0.6448	-2.2	Rayleigh
8	0.6560	-3.9	Rayleigh
9	0.6584	-7.4	Rayleigh
10	0.7935	-7.1	Rayleigh
11	0.8213	-10.7	Rayleigh
12	0.9336	-11.1	Rayleigh
13	1.2285	-5.1	Rayleigh
14	1.3083	-6.8	Rayleigh
15	2.1704	-8.7	Rayleigh
16	2.7105	-13.2	Rayleigh
17	4.2589	-13.9	Rayleigh
18	4.6003	-13.9	Rayleigh
19	5.4902	-15.8	Rayleigh
20	5.6077	-17.1	Rayleigh
21	6.3065	-16	Rayleigh
22	6.6374	-15.7	Rayleigh
23	7.0427	-21.6	Rayleigh
24	8.6523	-22.8	Rayleigh

Cuadro 3.3: Caracterización del perfil de retardo TDL-C

Los perfiles de retardo TDL-D y TDL-E tienen una componente de línea de vista en el primer *tap*, como consecuencia de ello, son modeladas a partir de un desvanecimiento con la distribución de Rice, razón por la cual tienen valores diferentes en el índice K y en las potencias medias, donde para cada perfil se tiene:

- Para el perfil TDL-D el primer pulso de retardo de la señal transmitida $s(t - \tau_1)$ tiene un índice $K_1 = 13.3$ dB con una potencia media de 0 dB.
- Para el perfil TDL-E el primer pulso de retardo de la señal transmitida $s(t - \tau_1)$ tiene un índice $K_1 = 22$ dB con una potencia media de 0 dB.

Tap	Retardo (τ_k) normalizado	Potencia en dB	Distribución del desvanecimiento
1	0	-0.2	Trayecto LoS
	0	-13.5	Rayleigh
2	35	-18.8	Rayleigh
3	612	-21	Rayleigh
4	1363	-22.8	Rayleigh
5	1405	-17.9	Rayleigh
6	1804	-20.1	Rayleigh
7	2596	-21.9	Rayleigh
8	1775	-22.9	Rayleigh
9	4042	-27.8	Rayleigh
10	7937	-23.6	Rayleigh
11	9424	-24.8	Rayleigh
12	9708	-30.0	Rayleigh
13	12525	-27.7	Rayleigh

Cuadro 3.4: Caracterización del perfil de retardo TDL-D

Tap	Retardo (τ_k) normalizado	Potencia en dB	Distribución del desvanecimiento
1	0	-0.03	LoS
	0	-22.03	Rayleigh
2	0.5133	-15.8	Rayleigh
3	0.5440	-18.1	Rayleigh
4	0.5630	-19.8	Rayleigh
5	0.5440	-22.9	Rayleigh
6	0.7112	-22.4	Rayleigh
7	1.9092	-18.6	Rayleigh
8	1.9293	-20.8	Rayleigh
9	1.9589	-22.6	Rayleigh
10	2.6426	-22.3	Rayleigh
11	3.7136	-25.6	Rayleigh
12	5.4524	-20.2	Rayleigh
13	12.0034	-29.8	Rayleigh
14	20.6519	-29.2	Rayleigh

Cuadro 3.5: Caracterización del perfil de retardo TDL-E

3.7 Estimación de canal

La estimación del canal corresponde al proceso mediante el cual se obtiene la información relacionada con las condiciones del canal de comunicaciones entre el transmisor y el receptor. Dentro de las características que se pueden extraer de la estimación de un canal se encuentran la información sobre atenuación, el retardo de propagación y la respuesta al impulso $h(t, \tau)$.

En las estimaciones de canal se deben considerar dos aspectos clave como la adaptabilidad y la complejidad, existe un dilema entre el uso de modelos de estimación, ya que un modelo complejo como el del filtro Wiener³ resuelve completamente este proceso, pero a la práctica, por el alto nivel de complejidad la implementación requiere de un número elevado de recursos, por eso las técnicas empleadas en las estimaciones de canal buscan mantener baja complejidad y alta eficiencia, que permitan que el canal estimado tenga un alto grado de similitud con el canal real.

De acuerdo con [28] se puede dividir la estimación de canal en dos tipos: esquemas asistidos por datos, son las técnicas con las cuales se extraen las propiedades y características del canal con base en los cambios obtenidos en señales de referencia al pasar a través del canal de comunicaciones, cuya ubicación en la rejilla de recursos es conocida tanto por el transmisor como por el receptor; el otro tipo son los estimadores ciegos, los cuales recurren al procesamiento digital de señales para extraer estadísticos que permitan determinar las condiciones del canal.

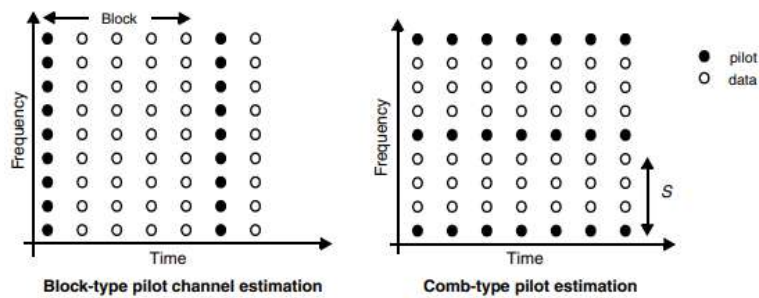


Figura 3.7: Tipos de distribución de las señales piloto en la cuadrícula de recursos. [29]

La estimación asistida por datos usa señales de referencia como es el caso de DM-RS, cuya ubicación en la rejilla de recursos es establecida de acuerdo al mapeo del PDSCH. En estas estima-

³Es un filtro que busca reducir el error cuadrático medio a partir del cálculo de la correlación entre la señal original y la señal con ruido, asumiendo que son procesos estocásticos estacionarios.

ciones, las señales piloto se envían periódicamente y se puede usar todas las subportadoras o todos los símbolos como pilotos, como se aprecia en la figura 3.7. La idea detrás del uso de las señales piloto es la de encontrar las condiciones del canal \hat{H} , con base en las variaciones de las señales piloto X y las señales recibidas Y . En [29] se encuentra que existen dos tipos de estimadores: Mínimos Cuadrados (LS) o Mínimo Error Cuadrático Medio (MMSE), ciertas limitaciones respecto a su rendimiento y complejidad ante variaciones presentes en el canal de comunicaciones.

3.7.1 Estimador por mínimos cuadrados

Este modelo parte de la premisa de que los cambios que afectan las señales de referencia tienden a ser lineales, por lo cual, se recurre al uso de la interpolación lineal para resolver el conjunto de ecuaciones lineales que permiten calcular \hat{H} . Se parte de la premisa en la que la señal captada por el receptor Y cumple con ser el producto entre la respuesta al impulso del canal y las señales de entrada (señales piloto X), tal como se observa en la siguiente expresión:

$$Y = H \cdot X \quad (3.14)$$

Despejando H se obtiene una aproximación \hat{H}_{LS} :

$$\hat{H}_{LS} = X^{-1} Y = \left(\frac{X_P}{Y_P} \right)^T, \text{ tomando los valores de } p = 0, 1, 2, 3, \dots, N-1 \quad (3.15)$$

Como se observa, la estimación basada en mínimos cuadrados resulta de baja complejidad, ya que todo el proceso consiste en resolver ecuaciones lineales, pero en entornos donde el ruido afecta considerablemente la calidad de la señal, el MSE tiende a ser mayor, por lo que, la implementación de esta técnica es limitada, ya que no puede adaptarse completamente a entornos dinámicos como los propuestos en 5G.

3.7.2 Estimador MMSE

La técnica de MMSE usa estadística de segundo orden, requiriendo el cálculo de matrices de auto correlación o la matriz de covarianza cruzada para minimizar el error cuadrático entre las señales estimadas y recibidas.

La primera parte consiste en calcular R_{HH} y R_{YY} como las matrices de auto correlación del

canal y de la señal recibida respectivamente, tras ello se calcula la covarianza cruzada entre las condiciones del canal y el conjunto de señales recibidas R_{HY} . También se debe asumir que σ_N^2 es la varianza del ruido presente en el canal de comunicaciones. Todas estas condiciones permiten determinar que:

$$\hat{H}_{MMSE} = R_{HH}[R_{HH} + \sigma_N^2(X \cdot X^H)^{-1}]^{-1} \hat{H}_{LS} \quad (3.16)$$

En donde el operador $(*)^H$ es la operación transpuesta conjugada y \hat{H}_{LS} es la estimación hecha con la interpolación lineal. La ecuación (3.16) busca reducir aún más el MSE a partir de la obtención de la estadística de segundo orden en conjunto con los valores interpolados linealmente del estimador LS. Esta técnica permite mejorar la precisión de la estimación ya que toma en cuenta los efectos que ocasiona el ruido en el canal.

Dentro de las principales limitaciones del estimador MMSE se encuentran:

- Si los parámetros como el modelo del canal y las estadísticas del ruido no se ajustan correctamente a la realidad, el MMSE no puede hacer una estimación óptima. Es por ello, que ambos parámetros deben ser determinados adecuadamente por el transmisor y el receptor.
- Requiere completamente de las estadísticas del ruido.
- No tiene una alta sensibilidad en entornos donde ocurren variaciones en el canal instantáneas o frecuentes.
- Requiere de una mayor capacidad computacional si se compara con la técnica de mínimos cuadrados, por lo que puede resultar ineficientes en aplicaciones que requieren tiempo de transmisión cortos y de bajas latencias.

Otras técnicas que basan su funcionamiento con pilotos son los algoritmos de Deep Learning, los cuales ofrecen una mayor adaptabilidad en los entornos dinámicos, ya que tiene la posibilidad de aprender patrones complejos y no lineales a partir de un conjunto de datos.

3.8 Ecualización de canal

Como se ha visto a lo largo de este capítulo, el canal de comunicaciones no presenta un comportamiento ideal, sino que se ve constantemente afectado por los fenómenos de dispersión, desva-

necimiento y el ruido. Uno de las consecuencias de no tratar adecuadamente la dispersión temporal es la Interferencia entre Símbolos (ISI), la cual degrada notablemente la calidad de la información recibida, es bajo esta problemática donde toma una connotación importante la técnica de ecualización, la cual se puede abordar tanto en comunicaciones digitales (compensando la ISI) como analógicas (compensando la distorsión en el dominio de la frecuencia) [20].

Para diseñar un ecualizador es necesario conocer la respuesta al impulso que tiene el canal $h(t, \tau)$, es por ello, que para desarrollar este proceso es necesario realizar la estimación de canal, la cual, como se mencionó en el apartado anterior (ver sección 3.7), se puede hacer a partir del envío de señales piloto y revisión de las variaciones que tuvieron al ser enviadas por el canal de comunicaciones. Bajo este proceso de envío de información piloto, se manejan los escenarios: por toma de decisiones o por la asistencia de datos, siendo el último materia de estudio de este documento, ya que la estimación se hace a partir del envío de las señales de 5G DM-RS.

Los ecualizadores se pueden agrupar en dos tipos: **lineales** y **no lineales**. El primer tipo de ecualizador como su nombre lo indica sigue un esquema completamente lineal e invariante en el tiempo, por lo tanto su caracterización se puede desarrollar a partir de la respuesta al impulso y la respuesta en frecuencia del canal. Por otro lado, los no lineales siguen un esquema no lineal [30], que los hace requerir de ajustes más complejos pero con una alta adaptabilidad a los canales que son variantes en el tiempo.

Algunos tipos de ecualizadores lineales son [30]:

- **Forzador de ceros (ZF):** Este ecualizador es lineal, por lo cual para el diseño es necesario encontrar el número de coeficientes que tiene el ecualizador L_{eq} y el valor de los coeficientes q_0, q_1, \dots, q_k . El objetivo es que los coeficientes q_k fueren la nulidad de la interferencia entre símbolos a la salida del ecualizador. Para cumplir con ello, se calcula la matriz inversa del canal H para “mitigar” los efectos generados en la transmisión de las señales piloto a través canal. Para anular completamente la ISI es necesario cumplir con $L_{eq} \rightarrow \infty$, además no contempla el ruido aditivo.
- **Ecualizador MMSE:** A diferencia del ecualizador ZF, en este si se contempla la presencia del ruido en el canal de comunicaciones. El objetivo es el de reducir el error cuadrático medio, de tal manera que la señal a la salida del ecualizador z_{eq} se asemeje a las señales piloto

transmitidas. Para diseñar el ecualizador es necesario calcular la matriz de autocorrelación de los datos de salida del canal R_{ZZ} y la correlación cruzada entre la secuencia de salida del canal y los símbolos deseados R_{ZA} , para de esta forma calcular $\vec{q}_k = R_{ZZ}^{-1} R_{ZA}$. De manera similar al ZF es necesario disponer de un alto número de coeficientes q_k que representan un alto coste computacional.

- **Ecualizador LMS:** En este tipo de ecualizador lineal se busca encontrar los coeficientes \vec{q}_k que minimicen el MSE, para ello se emplea el algoritmo del gradiente para encontrar el valor mínimo en la función de error. El proceso se hace de manera iterativa evaluando la función de error hasta alcanzar el mínimo, se usa una tasa de variación μ que se conoce como *paso de adaptación*, este parámetro es fundamental para evitar que el algoritmo diverja.

De manera similar, en [30] se encuentra la descripción de algunos ecualizadores no lineales, entre los cuales destacan el MLSE y el MLSE con la incorporación del algoritmo de Viterbi para reducir el coste computacional.

- **Estimador de Máxima Verosimilitud (MLSE)** En el receptor no se conoce la información que fue transmitida por la fuente, sin embargo, se conoce el valor finito M que se ajusta a cada esquema de modulación (por ejemplo $M = 2$ en BSPK y $M = 4$ en un QAM). También como punto de partida se conoce la respuesta al impulso que tiene el canal tras haber hecho una estimación. Conociendo la respuesta al impulso y el valor de M se calculan las posibles secuencias de símbolos transmitidos y se calcula la probabilidad de que la secuencia observada minimice la función de error con la secuencia recibida. Un inconveniente que se presenta es que en esquemas de dimensión M con una longitud de secuencia K alta, resulta computacionalmente complejo estimar la señal ecualizada, dada la expresión M^K .
- **Estimador MLSE con algoritmo de Viterbi:** El ecualizador MLSE presente un alto coste computacional con longitudes de secuencia altas, por ello se emplea el algoritmo de Viterbi que corresponde a una representación de un grafo de estados, mediante el cual se pueden reducir las dimensiones de las posibles secuencias de datos transmitidas, tomando como punto de partida el concepto del *camino más corto*. En [31] se encuentra una explicación al detalle de este algoritmo y como funciona para reducir la ISI y el ruido en la señal de salida del ecualizador.

3.9 Compensación de fase

En los sistemas de comunicaciones, existen diversos factores que influyen en la calidad de la señal captada por el receptor $r(t)$, tales como el desvanecimiento, el ruido aditivo y la dispersión (capítulo 3). Cuando se emplean técnicas de modulación como OFDM se presenta el ruido de fase, que es un término que se usa para referirse a variaciones cortas en el tiempo que ocurren en el oscilador del receptor [32], afectando la modulación en fase que se usa para transmitir los símbolos de la información.

En [33] se parte de que se tiene un sistema OFDM en el cual el ruido de fase en la entrada del receptor, por lo cual se puede expresar la señal recibida $r(t)$, en términos de la señal enviada $s(t)$, la respuesta al impulso del canal de comunicaciones $h(t)$, el ruido de fase $\Phi(t)$ y el ruido aditivo $\eta(t)$, de la siguiente forma:

$$r(t) = [x(t) \odot h(t)] e^{j\phi(t)} + \eta(t) \quad (3.17)$$

En el dominio de la frecuencia, se puede expresar la ecuación (3.17) como:

$$R_k = X_k H_k \underbrace{\Phi_0}_{CPE} + \underbrace{\sum_{i=0, i \neq k}^{N-1} X_i H_i \Phi_{(k-i)_N}}_{ICI} + \eta_k \quad (3.18)$$

Tomando a Φ_k como la transformada discreta del ruido de fase, expresada como:

$$\Phi_k = \frac{1}{N} \sum_{n=0}^{N-1} \phi[n] e^{-j \frac{2\pi}{N} nk} \quad (3.19)$$

Como se puede apreciar en la ecuación (3.18), existen efectos ocasionados por el ruido de fase, los cuales son:

- *CPE* es el error común de fase, el cual genera rotaciones en la constelación de los símbolos recibidos.
- *ICI* es la interferencia entre los símbolos recibidos, la cual rompe con la ortogonalidad de la forma de onda OFDM.

Así pues, en los sistemas de comunicaciones, se busca reducir el error de fase, para recuperar lo

símbolos transmitidos, para ello en 5G se propone usar la Señal de referencia para seguimiento de fase (PT-RS), la cual tiene baja densidad en la frecuencia y una alta densidad temporal, ya que se inserta a lo largo del eje de los símbolos OFDM en una sola coordenada de frecuencia.

3

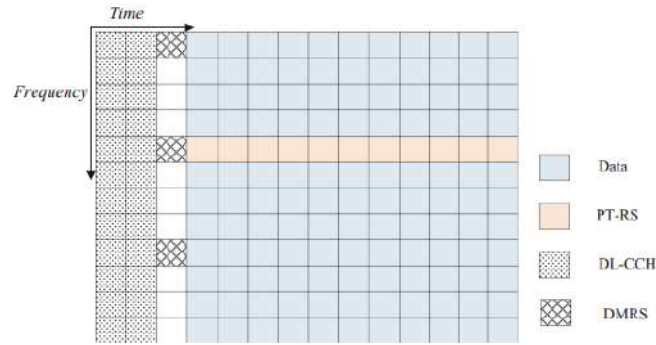


Figura 3.8: Inserción de señales PT-RS en la rejilla de recursos de 5G. [33]

En 5G se toman las señales de referencia PT-RS enviadas y de acuerdo con la estimación de canal, se hace una interpolación para recrear el efecto que tuvieron las condiciones sobre la señal, tras ello se hace una comparación con una rejilla de referencia para calcular el ángulo de rotación de la constelación θ_τ y de esta forma realizar el ajuste. En [33] el ángulo se calcula como:

$$\theta_\tau = \text{ángulo} \left\{ \sum_{k=1}^M Y_{k,3}^* Y_{k,l} \right\} \quad (3.20)$$

Reduciendo el efecto del CPE, se obtiene:

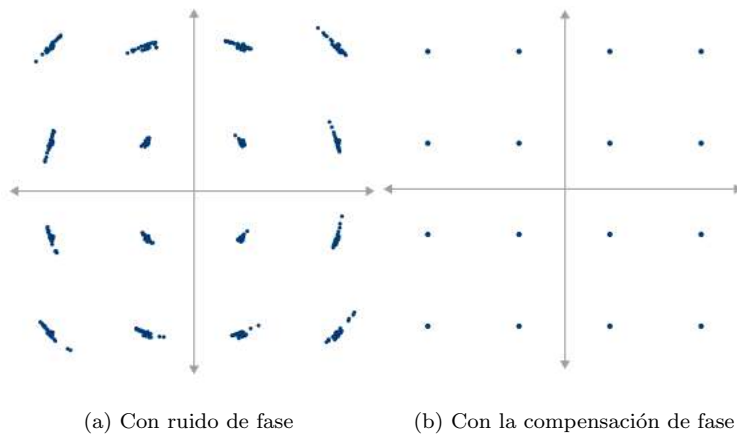


Figura 3.9: Diagramas de constelaciones 16QAM. [32]

3.10 Métricas de desempeño

Las métricas de desempeño son medidas que se usan para evaluar y cuantificar la calidad y eficacia del canal de comunicaciones. Con las métricas, se puede medir el impacto que tienen los procesos de estimación de canal y de ecualización en la recuperación de la información transmitida a través del canal de comunicaciones variante en el tiempo.

3.10.1 Magnitud del Vector de Error

EVM es una figura de mérito que se usa para evaluar la calidad de una señal modulada. Esta métrica de desempeño es un escalar que expresa la diferencia de valor de voltaje complejo esperado de un símbolo demodulado y el valor del símbolo recibido [34]. En la figura (3.10) se representa la EVM:

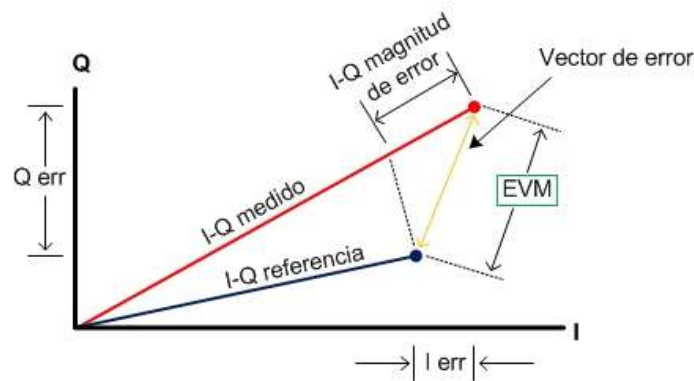


Figura 3.10: Representación gráfica de EVM. [34]

En [35] se encuentra que para calcular la EVM se usa la expresión (3.21):

$$EVM^2 = \frac{\frac{1}{N} \sum_{k=1}^N |X_{i,k} - X_{m,k}|^2}{\frac{1}{N} \sum_{k=1}^N |X_{i,k}|^2} \quad (3.21)$$

En donde: $X_{i,k}$ es el k -ésimo símbolo ideal de la constelación, $X_{m,k}$ es el k -ésimo símbolo recibido y N es el número de símbolos únicos de la constelación.

De acuerdo con el apartado de *Modulaciones soportadas en 5G*, 3GPP definió los valores tolerables de EVM para cada una de las modulaciones, tal como se aprecia en la tabla (3.6):

Esquema de modulación	Unidad	Nivel EVM promedio
$\pi/2$ -BPSK	%	30
QPSK	%	17.5
16QAM	%	12.5
64QAM	%	8
256QAM	%	3.5

Cuadro 3.6: Requisitos de EVM para cada modulación soportada en 5G. [36]

3.10.2 Tasa de Error de Bit

El BER es la probabilidad de recibir un bit de error durante un intervalo de tiempo especificado y representa una de las métricas de desempeño más importantes, puesto que permite medir la integridad de la señal transmitida.

$$\text{BER} = \frac{\# \text{ de bits de error}}{\text{Total de bits transmitidos}} \quad (3.22)$$

Cuando se emplea una modulación coherente, se pueden usar aproximaciones de las funciones de probabilidad de error de bit en función de $\gamma_b = E_b/N_0$ (SNR por bit), para las modulaciones M-PSK y M-QAM, en [37] se encuentra:

- **Para M-PSK:**

$$P_b(\gamma_b) \approx \frac{2}{\log_2 M} Q\left(\sqrt{2\gamma_b \log_2 M} \sin\left(\frac{\pi}{M}\right)\right) \quad (3.23)$$

- **Para M-QAM:**

$$P_b(\gamma_b) \approx \frac{4}{\log_2 M} Q\left(\sqrt{\frac{3\bar{\gamma}_b \log_2 M}{M-1}}\right) \quad (3.24)$$

En donde M es la dimensión de la modulación, γ_b es la SNR por bit, $\bar{\gamma}_b$ es la media del SNR por bit y $Q(*)$ es la función de distribución normal estándar con media $\mu = 0$ y varianza $\sigma = 1$, que corresponde a la expresión 3.25:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{u^2}{2}} du \quad (3.25)$$

Aprendizaje profundo

Índice

4.1	Generalidades	49
4.2	Algoritmos de aprendizaje supervisado	51
4.3	El aprendizaje profundo	52
4.4	Redes neuronales artificiales	53
4.4.1	El concepto biológico de la neurona	54
4.4.2	Modelado de neuronas artificiales	55
4.4.3	Funciones de activación	57
4.4.4	Función de propagación	58
4.4.5	Función de salida	59
4.4.6	Conceptos adicionales	59
4.4.7	Ventajas de uso	66
4.4.8	Tipos de redes neuronales artificiales	66
4.5	Fase de aprendizaje	68
4.5.1	Propagación hacia adelante	68
4.5.2	Propagación hacia atrás	69
4.6	Error cuadrático medio	71

4.1 Generalidades

En los últimos años se ha incrementado la popularidad del uso de la inteligencia artificial para facilitar algunas tareas del ser humano, siendo muy utilizado en diferentes campos, desde la medicina hasta la automatización de procesos. La premisa de que las computadoras puedan hacer tareas que suelen requerir de la inteligencia humana implica una mayor eficiencia y simplificación en la realización de tareas cotidianas, esto se logra con algoritmos y modelos para procesar una

gran cantidad de datos, aprender patrones complejos y tomar decisiones sin requerir intervención humana. La inteligencia artificial aborda varios campos entre los que se pueden destacar el *Machine Learning* (Aprendizaje Automático) y el *Deep Learning* (Aprendizaje Profundo).

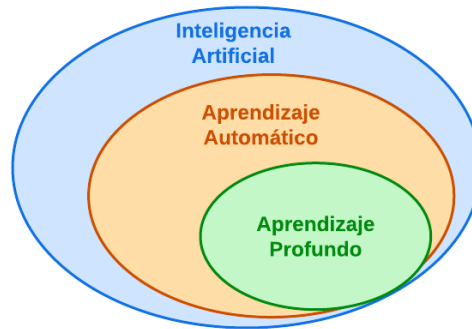


Figura 4.1: Subcategorías de la inteligencia artificial. (Autoría propia)

El aprendizaje automático es la rama de la inteligencia artificial que se centra en desarrollar algoritmos y modelos matemáticos para que las máquinas aprendan patrones y realicen tareas sin programadas explícitamente. De forma concreta, se puede entender como la generalización de comportamientos o patrones a partir de un conjunto de datos que son usados como “experiencias” en el aprendizaje de la máquina. Es, por lo tanto, un proceso de inducción al conocimiento [38] ya que, a partir de casos particulares, el sistema es capaz de generalizar y adquirir nuevo conocimiento que puede ser usado en nuevas situaciones, esto resulta fundamental puesto que permite a la máquina adaptarse a la variabilidad y a la complejidad de los datos.

De acuerdo con [38] los problemas que se pueden resolver a través de aprendizaje automático son:

- **Aprendizaje supervisado:** Se refiere al tipo de aprendizaje en el que se entrena el modelo a partir de un conjunto de datos en el que cada ejemplo tiene una salida deseada o etiqueta. El objetivo de este tipo de aprendizaje es hallar las relaciones entre la entrada y la salida, para que el modelo pueda predecir con precisión nuevas salidas con datos no vistos. Entre los problemas que se pueden abordar con estos modelos están: la regresión, referido a la predicción de respuestas continuas como la eliminación del ruido en señales de un electrocardiograma; y la clasificación, usada para predecir resultados discretos, un ejemplo es tomar las señales del electrocardiograma para definir si el paciente padece o no una afección cardíaca.

- **Aprendizaje no supervisado:** Es un enfoque en el que un modelo es entrenado sin conocer la salida deseada, por lo cual recurre a la identificación de patrones y estructuras no intrínsecas en los ejemplos. Dentro de las técnicas que se abordan con este tipo de aprendizaje se encuentran: el *clustering* donde se agrupan los datos de entrada de acuerdo a un criterio específico y la reducción de dimensionalidad, en el cual se busca conservar la mayor cantidad de información relevante reduciendo la complejidad del modelo.

4.2 Algoritmos de aprendizaje supervisado

Para solucionar problemas asociados a la regresión y a la clasificación, existen algoritmos de aprendizaje automático que son elegidos según el problema, la naturaleza de los datos o la búsqueda de una alta eficiencia computacional. Algunos algoritmos empleados listados en [39] son:

- **Regresión Lineal:** Como su nombre lo indica, encuentra una relación lineal entre los datos de entrada y las etiquetas.
- **Regresión Logística:** Se usa comúnmente en problemas de clasificación, dado que su objetivo se centra en predecir la probabilidad de ocurrencia de diferentes resultados posibles. En concreto, se busca generar un hiperplano que separe linealmente las clases.
- **K-vecinos más cercanos:** Con este algoritmo se busca que objetos con propiedades similares estén cerca en el espacio de características, de manera tal que cuando se presente una nueva instancia, el algoritmo sea capaz de hallar los k vecinos más cercanos en función de la distancia (euclidiana, Manhattan, Minkowski o de Hamming), después se busca que el algoritmo tome decisiones de acuerdo con las características de dichos vecinos.
- **Árboles de decisión:** Este algoritmo se basa en el concepto de “divide y vencerás” ya que se hace una búsqueda exhaustiva para identificar los puntos de división óptimos dentro de una arquitectura de árbol [40]. Cada nodo representa una pregunta sobre una característica específica, cada rama es una posible respuesta a dicha pregunta y las hojas son las predicciones.
- **Clasificador de Naive-Bayes:** Su funcionamiento está basado en el teorema de Bayes. Se utiliza para asignar una clase a un conjunto de ejemplos según la probabilidad condicional de que la instancia pertenezca a la clase de acuerdo con las características observadas. Este clasificador, parte de la suposición de que todas las variables son independientes y que contribuyen en igual medida al resultado final.

- **Máquinas de soporte vectorial:** El objetivo de este algoritmo es el de encontrar un hiperplano de separación óptima que maximice el margen entre clases, para ello se usan las funciones denominadas como kernel. Estas funciones se usan para mapear los datos de entrada a un espacio de características de mayor dimensión que permitan la separación lineal entre clases.
- **Algoritmos de DL:** Dado que el enfoque del aprendizaje profundo son las redes neuronales artificiales, existen diferentes tipos de redes, las cuales pueden aprender patrones complejos de los datos de entrada para generar una predicción de la salida. Con este tipo de algoritmos es posible tener el procesamiento de datos no estructurados como imágenes, vídeos, audio o texto.

4.3 El aprendizaje profundo

El aprendizaje profundo es un subcampo del aprendizaje automático (figura 4.1), en el cual se recurre al uso del concepto de redes neuronales artificiales para aprender patrones y estructuras complejas de grandes conjuntos de datos etiquetados. La técnica es útil para abordar el análisis de datos no estructurados en los que no hay una forma predefinida, sino que vienen de fuentes como texto, imágenes, voz y vídeos, lo que dota al aprendizaje profundo de una mayor funcionalidad y flexibilidad [41].

En el aprendizaje profundo se trabajan modelos definidos por el número de capas, en cada capa hay un número de neuronas artificiales, que operan para extraer información del conjunto de datos para pasar a la siguiente capa hasta llegar a una salida esperada \hat{y} , que se asemeje en mayor medida a la salida real y . Como es posible apreciar, a partir del uso de este modelo, es posible abordar soluciones a problemas asociados con la regresión y la clasificación, funcionando como un “depurador” de información [42].

El concepto de profundo se asocia con el número de capas sucesivas empleadas para que el modelo pueda aprender a partir de ejemplos. Actualmente, se pueden encontrar modelos que llegan a tener cientos de capas, lo que aumenta la complejidad del modelo, pero que permite predicciones con mayor precisión. Aunque el aprendizaje profundo requiere de mayor tiempo para entrenar el modelo por la gran cantidad de parámetros y datos, al evaluarlo con nuevos datos, tarda menos comparado con otros algoritmos del aprendizaje automático [43].

Pese a que el concepto de aprendizaje profundo se ha trabajado desde la década de los 80, en los últimos años ha adquirido una mayor connotación en la tecnología, dadas las ventajas que ofrece a nivel de escalabilidad, funcionalidad y flexibilidad. En [38] se menciona que la popularidad se debe a dos factores: la gran cantidad de datos etiquetados que pueden ser estructurados y no estructurados; y el uso de las GPU (*Graphics Processing Unit*) como herramienta que combina los clústers y la computación en la nube, para acelerar los tiempos en el entrenamiento.

Algunas de las aplicaciones del aprendizaje profundo que se listan en [41] son:

- Procesamiento del lenguaje natural.
- Cuidado de la salud.
- Asistentes virtuales como Alexa o Siri.
- Reconocimiento de objetos.
- Tratamiento de señales de sonido aplicadas al mundo del entretenimiento audiovisual.
- Análisis del estado de ánimo de las personas.
- Adición de color a imágenes en blanco y negro.
- Traducción automática.

4.4 Redes neuronales artificiales

Las ANN son modelos basados en conceptos biológicos de procesamiento de información y buscan satisfacer la necesidad de emular el pensamiento y la toma de decisiones humanas, dichas redes se basan en las redes biológicas conformadas por las neuronas del sistema nervioso. Una de las particularidades de estas redes es la capacidad de agruparse mediante capas, esto les da la facultad de abstraer información de una entrada y procesarla con el fin de obtener algún resultado en específico a la salida.

El modelo biológico ha permitido establecer modelos computacionales necesarios para que una red artificial se asemejase a lo que pensaría un ser humano, hay muchas aplicaciones, pero destacan las memorias holográficas o su versión computacional, las memorias asociativas de acceso por contenidos, la representación distribuida, el procesamiento paralelo, el aprendizaje, la inferencia, la construcción y clasificación de patrones con aprendizaje supervisado [44], y por auto-organización [45].

4.4.1 El concepto biológico de la neurona

El razonamiento y la capacidad de análisis del humano se debe a la interconexión de lo que se conoce como neuronas, este tipo de célula recibe y envía información como impulsos eléctricos a la parte central de procesamiento del sistema nervioso, el encéfalo.

Una buena definición de lo que es una neurona es [46]: *las neuronas son células muy especializadas con una morfología característica y unas propiedades funcionales que les permiten la recepción, generación y propagación de impulsos nerviosos. Poseen, además dispositivos específicos de contacto intercelular, para la transferencia inter-neuronal de señales nerviosas en los circuitos neuronales.*

El sistema nervioso consiste en un gran número de neuronas vinculadas entre sí para formar vías de conducción funcionales. El sitio en el que dos neuronas entran en estrecha proximidad y producen una comunicación ínter-neuronal funcional se denomina sinapsis.

La ciencia se ha empeñado en entender cómo funciona el procesamiento neuronal, cómo la interacción de las neuronas por medio de la sinapsis está relacionada con la toma de decisiones y para ello ha tenido que comprender la estructura de la neurona. Identificar los elementos de la neurona es útil para tener un modelo que se asemeje a la concepción biológica de una neurona, emulando su comportamiento e interacción entre neuronas de una misma capa. En la figura (4.2) se representan las partes que componen una neurona, las cuales son:

- **Dendritas:** Son extensiones especializadas que se asemejan a la rama de un árbol. Las dendritas ayudan a recibir señales entrantes de ellas.

- **Cuerpo celular o soma:** Es el núcleo de la neurona, similar a una célula que contiene el núcleo y todos los demás orgánulos celulares. Se adhieren a todas las dendritas y así integran todas las señales.
- **Axones:** Son extensiones finas y alargadas parecidas a fibras de la membrana de la célula nerviosa. Los axones van desde el cuerpo celular de una neurona hasta la terminal de la siguiente neurona.

Parts of a Neuron with Functions

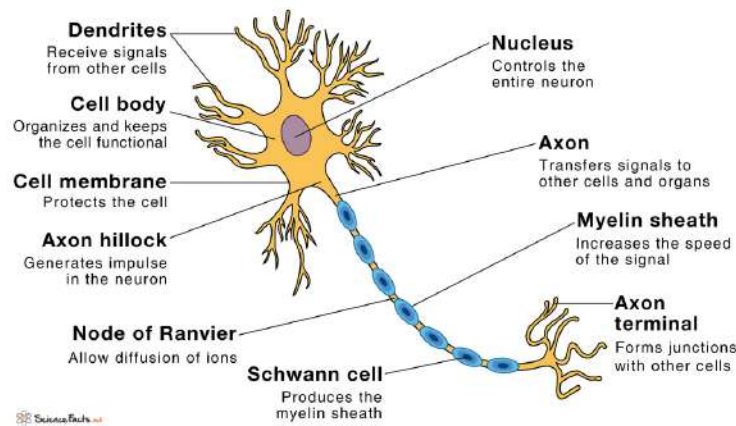


Figura 4.2: Partes de la neurona biológica. [47]

4.4.2 Modelado de neuronas artificiales

De acuerdo con lo descrito anteriormente sobre el concepto de neurona, sus partes y la conexión mediante sinapsis para formar una red similar a una telaraña, es posible aplicar una analogía en el ámbito computacional, ya que modelado de una red neuronal artificial se basa en el desarrollo de algoritmos matemáticos para emular el comportamiento del cerebro en una tarea o función, tomando la experiencia como aprendizaje.

Lo que se busca con el entrenamiento de una red neuronal artificial es calcular los pesos (sinapsis) \vec{w}_n para producir una salida deseada \hat{y} de acuerdo con los datos de entrada \vec{x} y las etiquetas y . El modelo obtenido es una función que representa un mapeo entre entrada y salida, cuyos parámetros ajustables son los pesos de la neurona, que se asocian con la sinapsis porque determinan la fuerza y dirección de la conexión entre las neuronas.

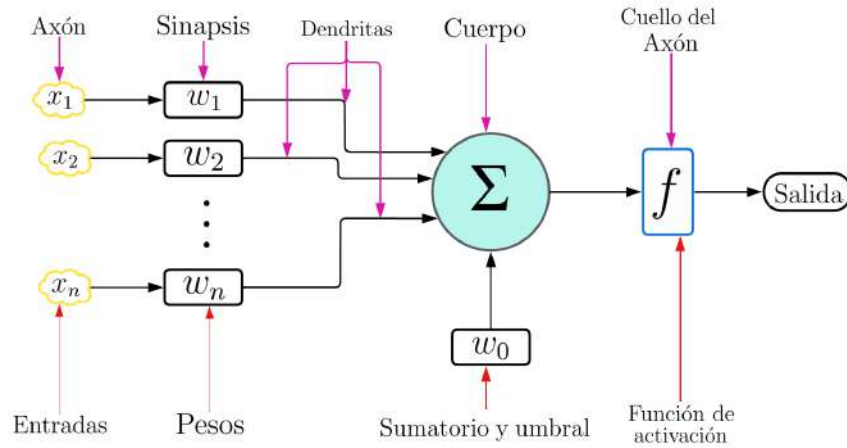


Figura 4.3: Diagrama de bloques del modelo de una neurona artificial. (Autoría propia)

De acuerdo con la figura (4.3) se identifican los elementos que intervienen en una neurona artificial, los cuales son:

- Conjunto de datos de entrada $X = (x_1, x_2, x_3, \dots, x_n)^T$ en donde n es el número de entradas a la neurona.
- Los pesos \vec{w}_n son los enlaces sinápticos y sus valores se hallan iterativamente para minimizar alguna función de pérdida, que mide la diferencia entre las salidas reales y las predichas por el modelo.
- w_0 es el umbral o sesgo.
- Regla de propagación que determina el potencial resultante en la interacción de cada entrada de la neurona con sus respectivos pesos. Resulta en la ponderación de las entradas X con los pesos \vec{w}_n .
- La función de activación determina la salida de la neurona \hat{y} de acuerdo con la regla de propagación y el umbral, introduciendo no linealidad.

De manera general, en una red neuronal se tiene un conjunto de neuronas que se agrupan en capas sucesivas, tal como se enseña en la figura (4.4):

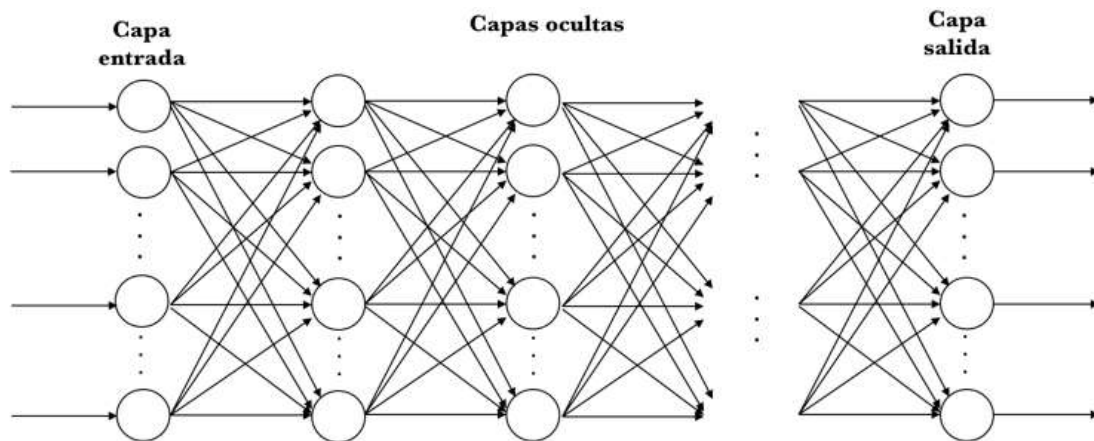


Figura 4.4: Estructura por capas de una ANN. [48]

La distribución de neuronas dentro de la red se realiza formando niveles o capas, con un número determinado de dichas neuronas en cada una de ellas. A partir de su situación dentro de la red, se pueden distinguir tres tipos de capas [49]:

- **Capa de entrada:** Es la capa que recibe directamente la información proveniente de las fuentes externas de la red.
- **Capas intermedias:** Son internas a la red y no tienen contacto directo con el entorno exterior. El número de niveles ocultos puede estar entre cero y un número elevado. Las neuronas de las capas ocultas pueden estar interconectadas de distintas maneras, lo que determina, junto con su número, las distintas topologías de redes neuronales.
- **Capa de salida:** Transfieren información de la red hacia el exterior.

En los siguientes apartados se describen algunos parámetros fundamentales en las configuraciones de las ANN, como es el caso del tipo de función de activación, las funciones de error y el optimizador para minimizar las funciones de pérdida y error en el entrenamiento.

4.4.3 Funciones de activación

Una neurona biológica puede estar activa (excitada) o inactiva (no excitada). La función de activación $f(*)$ calcula el estado de actividad de una neurona, transformando la entrada global en un valor (estado) de activación, cuyo rango normalmente va de $(0, 1)$ o de $(-1, 1)$. Esto es así, porque una neurona puede estar totalmente inactiva o activa [49].

Algunas funciones de activación con su respectiva derivada se presentan en la figura (4.5):

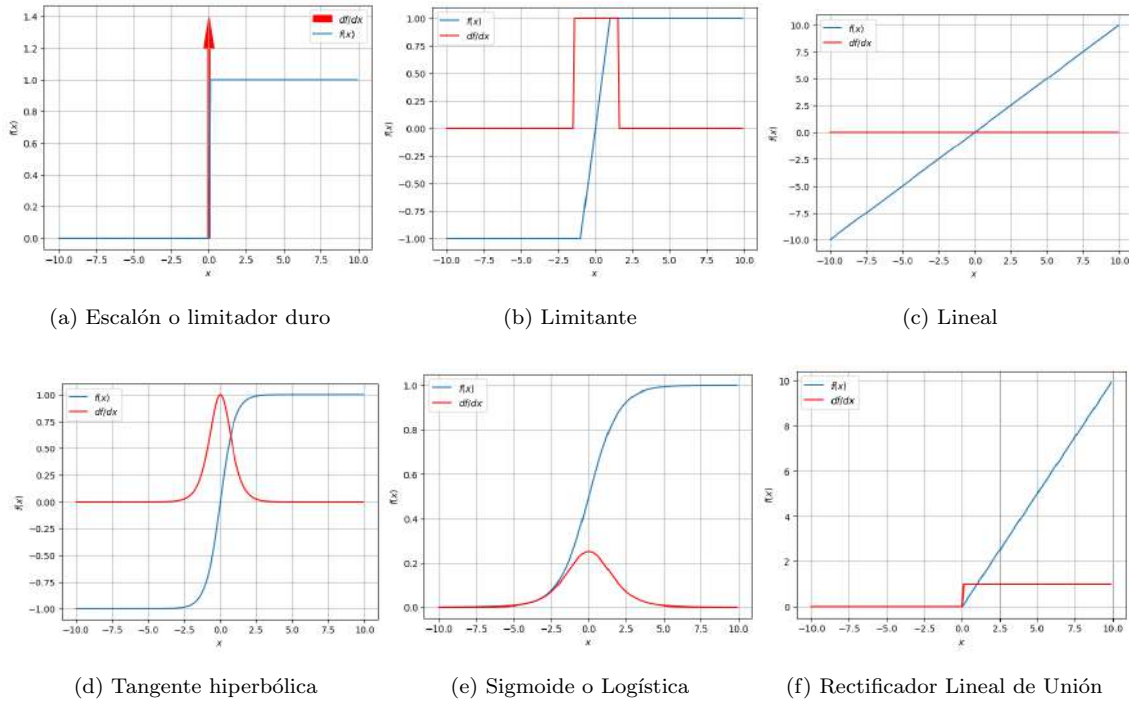


Figura 4.5: Funciones de activación empleadas en las ANN. (Autoría propia)

4.4.4 Función de propagación

La función de propagación indica el procedimiento que se debe seguir para combinar los valores de entrada y los pesos de las conexiones que llegan a una neurona. En un escenario con una capa oculta se usa la notación w_{ij} en donde i es el número de la neurona en la capa y j es el índice de la conexión sináptica en la entrada de la neurona. Con esta notación, es posible agrupar todos los pesos en una matriz W con dimensión $i \times j$ que se expresa de la siguiente forma:

$$W = \begin{pmatrix} w_{00} & w_{01} & w_{02} & \cdots & w_{0j} \\ w_{10} & w_{11} & w_{12} & \cdots & w_{1j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{i0} & w_{i1} & w_{i2} & \cdots & w_{ij} \end{pmatrix}_{i \times j} \quad (4.1)$$

En [50] se encuentra que los pesos pueden ser nulos, positivos o negativos, cada uno representa un estado de excitación que se describe como:

- Cuando es positivo w_{ij} , quiere decir que la interacción entre las neuronas i y j es excitadora.

- Cuando es nulo $w_{ij} = 0$ se considera que no hay una sinapsis entre las neuronas.
- Cuando es negativo $w_{ij} < 0$ la conexión es inhibitoria.

Así pues la función de propagación permite obtener el valor de potencial post sináptico net_i de una neurona en un instante t . De acuerdo con una función de probabilidad σ , el valor de net_i se calcula con base en los valores de entrada y pesos recibidos. La función más utilizada es de tipo lineal y consiste como en la suma ponderada de las entradas con los pesos sinápticos asociados a ellas [50]:

$$net_j = \sum_{k=1}^i w_{kj} x_k(t) = W_j^T X \quad (4.2)$$

4.4.5 Función de salida

El último componente que una neurona necesita es la función de salida. El valor resultante de esta función es la salida de la neurona, por ende, la función de salida determina que valor se transfiere a las neuronas vinculadas. Si la función de activación está por debajo de un umbral determinado, ninguna salida se pasa a la neurona subsiguiente. Normalmente, no cualquier valor es permitido como una entrada para una neurona ya que se suele trabajar con normalización de datos, por lo tanto, los valores de salida están comprendidos en el rango $[0, 1]$ o $[-1, 1]$ [49].

La función de salida para el modelo de una sola capa, con múltiples salidas, se expresa de la forma:

$$y_j = f(net_j) = f\left(\sum_i w_{ij} x_i(t)\right) \quad (4.3)$$

4.4.6 Conceptos adicionales

En este apartado del documento se hace una breve presentación de algunos conceptos que son tenidos en cuenta en el desarrollo de modelos de redes neuronales artificiales, ya sea en el desarrollo del modelo o en el entrenamiento del mismo para conseguir que la salida estimada \hat{y} tenga una gran precisión si se compara con la salida real y . Algunas configuraciones a tener en cuenta son:

Función objetivo

Es una función usada para cuantificar la diferencia entre las predicciones hechas por el modelo y las etiquetas que se usan en el entrenamiento y posterior validación [51]. En el aprendizaje profundo es común encontrarse con dos escenarios: función de error o función de “ganancia”, en donde se busca minimizar o maximizar respectivamente. En las funciones de error $E(w)$ se busca conseguir que $E \rightarrow 0$, para ello, en el entrenamiento se ajustan los pesos iterativamente con optimizadores hasta obtener la convergencia del algoritmo (mínimo de la función de error).

Existen diferentes funciones $E(w)$ y se eligen de acuerdo al tipo de etiquetas, ya que pueden ser continuas, binarias o multiclase. En [51] se definen dos funciones de error fundamentales:

- **Suma de errores cuadráticos:** De manera generalizada se expresa como:

$$E_{SSE} = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^L (\hat{y}_{ij} - y_{ij})^2 \quad (4.4)$$

Siendo n el tamaño del conjunto de datos, L es la cantidad de salidas en el modelo de red neuronal artificial, \hat{y}_{ij} son las salidas estimadas y y_{ij} son las salidas observables. La división de $\frac{1}{2}$ se hace por conveniencia [52] al momento de calcular los gradientes en el algoritmo de aprendizaje *backpropagation*. Este tipo de función se usa principalmente en escenarios en los cuales las etiquetas son continuas. Existen otras opciones de funciones $E(w)$ como el porcentaje absoluto de error y la suma de los logaritmos al cuadrado.

- **Entropía cruzada:** Esta función de pérdida se usa principalmente en casos en los que se desean obtener las probabilidades de que una predicción sea cierta o no y se expresa como:

$$E_{CE} = - \sum_{i=1}^n \sum_{j=1}^L [\hat{y}_{ij} \log(\hat{y}_{ij}) + (1 - \hat{y}_{ij}) \log(1 - \hat{y}_{ij})] \quad (4.5)$$

Con esta función de error se calcula la diferencia entre dos diferencias de probabilidad, la asociada con las etiquetas y las predicciones del modelo ANN. En el caso de las salidas multiclase se emplea la entropía-cruzada categórica.

La función de error tiene un alto impacto en el desempeño que tiene el modelo neuronal para un determinado problema [51], es por ello que debe elegirse adecuadamente, para que en el entrenamiento el algoritmo converja y en la validación el modelo generalice con los nuevos datos.

Hiper-parámetros

Los hiper-parámetros son aquellas características en los modelos que no se configuran en el entrenamiento sino antes de este, el objetivo de estos es permitir que el modelo de ANN sea capaz de generalizar adecuadamente los datos, sin tener un efecto de subajuste¹ o de sobreajuste², es por ello que son fundamentales para tener un buen desempeño de predicción. En [52] se listan los siguientes hiper-parámetros:

- **Topología de la red:** La topología de la red corresponde principalmente a la selección de un modelo en concreto de acuerdo con el tipo de datos X y etiquetas Y que se están usando. Dado el alcance que tiene cada modelo, no hay un criterio en concreto para elegirlo, sin embargo, en el procesamiento de imágenes se recomienda emplear las redes convolucionales.
- **Funciones de activación:** Se aplican en las capas de cada red y la selección va acorde con el tipo de etiquetas que se están empleando, por ejemplo, si se maneja etiquetas binarias es recomendable usar la función sigmoide (figura 4.5.e), para problemas multiclase se emplea la función *softmax* y para regresiones la función lineal (figura 4.5.c). En las capas ocultas, [52] recomienda usar la función ReLU o su variación leaky ReLU.
- **Funciones objetivo:** Es la función que se busca maximizar (función de precisión) o minimizar (función de error) para garantizar la correcta generalización del modelo con nuevos datos. En [52] se sugieren seleccionarlas de acuerdo al tipo de salida Y teniendo así: (a) para salidas continuas usar la función de pérdida MSE; (b) con etiquetas binarias la función logística; (c) para múltiples clases usar la función categórica de entropía cruzada.
- **Número de capas ocultas:** Determina la profundidad de la red neuronal, por ello, es necesario conocer si la solución se puede abordar linealmente (una sola capa) o si es necesario usar múltiples capas dada la relación no lineal entre los datos de entrada y las etiquetas. Este hiper-parámetro está ligado al dilema de la estabilidad-plasticidad³.
- **Cantidad de neuronas en cada capa:** La densidad de las capas de entrada y de salida se ajustan de acuerdo a la dimensión de X y de Y . Para el número de neuronas en las capas ocultas se recomienda seguir las siguientes reglas: (a) el valor debe estar entre la cantidad de

¹El modelo no es capaz de encontrar una relación entre los datos de entrada y de salida, generando errores altos en el entrenamiento

²El modelo es capaz de hacer predicciones precisas para los datos de entrenamiento pero no para los datos nuevos

³Un modelo de RNA debe aprender nuevos patrones sin olvidar los patrones aprendidos

neuronas de la capa de entrada y la capa de salida; (b) usar la expresión (4.6):

$$\text{neuronas por capa} = \frac{2}{3} \text{ tamaño de } X + \text{tamaño de } Y \quad (4.6)$$

- **Tipo de regularización:** Se usan para prevenir el sobre-ajuste del modelo, penalizando los pesos o desactivando neuronas aleatoriamente.
- **Tasa de aprendizaje:** Es una variable μ que controla la modificación de los pesos w_{ij} y el umbral w_0 en el entrenamiento de la red. Un valor muy bajo hace que el algoritmo converja en un mínimo local de la función de error; por otra parte, una tasa de aprendizaje elevada impide que el algoritmo converja, un ejemplo de ello se presenta en la siguiente figura:

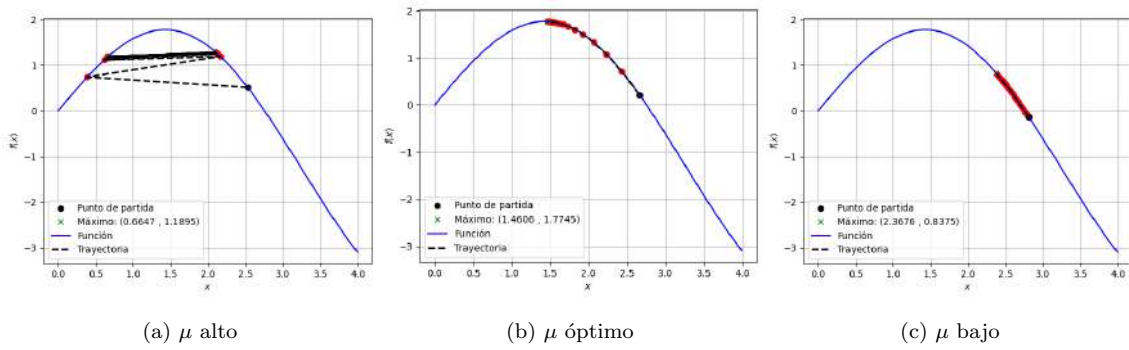


Figura 4.6: Variación de μ para hallar el máximo de una función. (Autoría propia)

Dadas las variaciones en la tasa de aprendizaje, autores como [52] recomiendan usar tasas de aprendizaje inferiores a 1×10^{-3} .

- **Número de épocas:** Una época representa la iteración completa a través de los datos de entrenamiento, así pues el número de épocas determina cuántas veces pasa el modelo recorre los datos de entrenamiento hasta alcanzar el cumplimiento de la función objetivo.
- **Tamaño del lote:** Es el número de submuestras que se procesan en simultáneo para actualizar los pesos y el valor en la función objetivo hasta alcanzar el mínimo.
- **Normalización de datos:** Esta etapa es un pre-procesamiento de los datos en la entrada X de cada modelo y se usa para escalar los datos a un intervalo $[0, 1]$ o $[-1, 1]$, de esta forma, la actualización de los pesos se puede dar de una forma más suavizada, sin tener cambios abruptos en cada lote.

Optimizadores

El optimizador es un componente clave en el entrenamiento de la red neuronal artificial [52]. Es un algoritmo matemático que se usa para minimizar la función de error $E(w)$, para ello, basa su funcionamiento en el uso de las derivadas y las derivadas parciales. La elección del optimizador determina la rapidez en obtener la convergencia en el entrenamiento y el desempeño final que tenga el modelo [53].

En [53] se listan algunos de los optimizadores más importantes, ya que funcionan con la variación de la tasa de aprendizaje μ o con la incorporación del *momentum* como mecanismo mediante el cual se evalúa el histórico de pesos para suavizar la curva de error. Algunos optimizadores son:

- **Descenso de Gradiente Estocástico (SGD)** Es un algoritmo que se usa para actualizar los pesos w_{ij}^{k+1} en función del peso actual w_{ij}^k y la dirección del gradiente de la función de error $\nabla E(w_{ij}^k)$. Lo primero que se hace es iniciar los pesos aleatoriamente con $k = 0$, tras ello se calcula el gradiente del error evaluado en los pesos actuales y se actualizan los pesos con:

$$w_{ij}^{k+1} = w_{ij}^k - \mu \nabla E(w_{ij}^k) \quad (4.7)$$

Este proceso se repite iterativamente hasta converger, lo que quiere decir que el gradiente de la función de error tiende a 0. Como se puede observar en la ecuación (4.7), es importante considerar un valor adecuado de la tasa de aprendizaje tal como se representa en la figura (4.6). Algunas limitaciones son la alta sensibilidad al valor de μ , las fluctuaciones en la curva de error y la dificultad de hallar el mínimo exacto [54].

- **Descenso de Gradiente estocástico con momento:** Uno de los principales inconvenientes del SGD son las fluctuaciones que dificultan la obtención del mínimo exacto de la función. Es bajo esta premisa donde se recurre al concepto del momento o *momentum*, el cual acelera la convergencia del algoritmo y optimiza la capacidad de alcanzar el mínimo global de la función $E(w)$. Las expresiones que se usan en este método son:

$$v_k = \beta v_{k-1} + \mu \nabla E(w_{ij}^k) \quad (4.8)$$

$$w_{ij}^{k+1} = w_{ij}^k - v_k \quad (4.9)$$

Siendo v_k la información del gradiente anterior, β es el momentum o factor de decaimiento

del gradiente anterior (se suele usar $\beta = 0.9$) y v_{k-1} el gradiente en la iteración anterior. El término de v_k aumenta cuando las componentes del gradiente tienen la misma dirección y se reduce cuando son opuestas [54], permitiendo un tiempo de convergencia menor. Comparando el SGD con y sin momentum se presenta la figura (4.7)

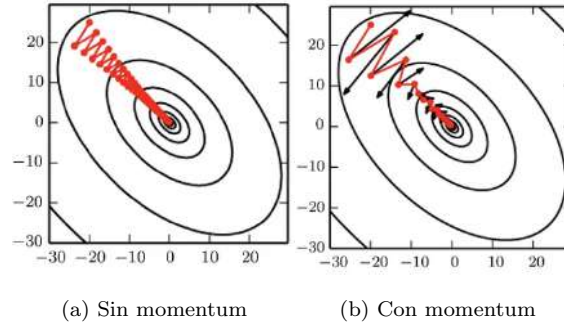


Figura 4.7: Comparativa del descenso de gradiente estocástico. [55]

- **Propagación media cuadrática (RMSProp):** Es un modelo donde la tasa de aprendizaje varía de acuerdo con la media móvil de los cuadrados de los gradientes anteriores. La media móvil es la actualización ponderada de los cuadrados de los gradientes anteriores, tal como se muestra a continuación:

$$M[v^2]_k = \beta M[v^2]_{k-1} + (1 - \beta)M[v^2]_k \quad \text{donde:} \quad v_k = \nabla E(w_{ij}^k) \quad (4.10)$$

Con β siendo el factor de decaimiento de los gradientes. Como se puede ver en la ecuación, es necesario contar con el histórico del gradiente y darle una mayor prioridad a la componente de $k - 1$ por sobre el gradiente actual, así se reducen las fluctuaciones en la curva de error. Con ello, la actualización de los pesos se hace con la ecuación (4.11):

$$w_{ij}^{k+1} = w_{ij}^k - \frac{\mu}{\sqrt{M[v^2]_k + \epsilon}} v_k \quad (4.11)$$

El término ϵ se usa para evitar la división con 0. De acuerdo con la expresión anterior, se identifica que el método de RMSProp divide la tasa de aprendizaje exponencialmente con un decaimiento promedio de los cuadrados del gradiente. El valor de β se recomienda fijarlo en $\beta = 0.9$ y la tasa de aprendizaje $\mu = 0.001$ [54].

- **ADAM:** También es conocido como *Adaptive Moment Estimation* es un método de optimización estocástica que requiere únicamente de gradientes de primer orden, con alta eficiencia computacional y que requiere de menos tiempo para conseguir la convergencia [56]. Este

método resulta en una variación del RMSProp (que maneja una tasa variable) junto con el *momentum*, por lo cual tiene en cuenta decaimiento promedio de la suma de los cuadrados de los gradientes pasados v_k en conjunto la información del gradiente anterior m_k . Así pues, se toma la notación de g_k como el gradiente de la función de error en el peso w_{ij}^k , por lo que se puede expresar v_k y m_k como:

$$m_k = \beta_1 m_{k-1} + (1 - \beta_1) g_k \quad (4.12)$$

$$v_k = \beta_2 v_{k-1} + (1 - \beta_2) g_k^2 \quad (4.13)$$

Siendo m_k y v_k el primer y segundo momento respectivamente. Tanto β_1 como β_2 son tasas de decaimiento exponencial para los dos momentos, en [56] se fijan en el intervalo $[0, 1)$. Además de los dos momentos, se usan factores de corrección del umbral: \hat{m}_k y \hat{v}_k :

$$\hat{m}_k = \frac{m_k}{1 - \beta_1^k} \quad y \quad \hat{v}_k = \frac{v_k}{1 - \beta_2^k} \quad (4.14)$$

Así pues, la actualización de pesos se hace de acuerdo con la ecuación (4.15):

$$w_{ij}^{k+1} = w_{ij}^k - \frac{\mu}{\sqrt{\hat{v}_k} - \epsilon} \hat{m}_k \quad (4.15)$$

Siendo ϵ una constante que se emplea para evitar la división con 0. Se sugiere que los valores para las constantes de decaimiento sean de $\beta_1 = 0.9$, $\beta_2 = 0.999$ y $\epsilon = 10^{-8}$ [54].

En [56] se realiza una comparativa entre los métodos de optimización listados previamente, para ello se hace una evaluación del desempeño de los algoritmos variando las topologías de red, tal como se presenta en la siguiente imagen:

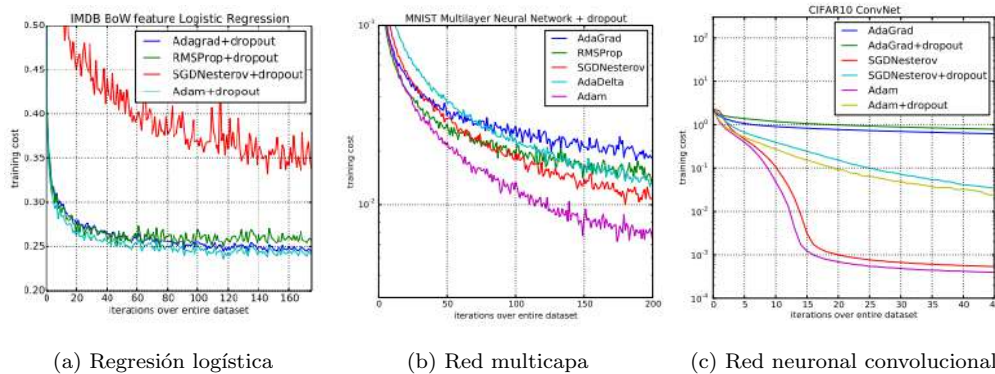


Figura 4.8: Comparativa del desempeño de los métodos de optimización. [56]

Varios autores como [53], [54] y [56] sugieren elegir el optimizador ADAM, debido al bajo costo computacional en cada iteración y el poco tiempo que tiene para conseguir la minimización de la función de error. Además, funciona con las diversas topologías de ANN, lo cual brinda un mayor capacidad de adaptabilidad en cualquier entorno y conjunto de datos.

4.4.7 Ventajas de uso

Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante, etc. Esto hace que ofrezcan numerosas ventajas y que este tipo de tecnología se esté aplicando en múltiples áreas [49]. Entre las ventajas se incluyen:

- **Aprendizaje adaptativo:** Capacidad de aprender a realizar tareas basadas en un entrenamiento o en una experiencia inicial.
- **Auto-organización:** Una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.
- **Tolerancia a fallos:** La destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, incluso sufriendo un gran daño.
- **Operación en tiempo real:** Los cálculos neuronales pueden ser realizados en paralelo; para esto se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.

4.4.8 Tipos de redes neuronales artificiales

La diferenciación de las redes neuronales artificiales se basa en la arquitectura, en la forma en que las unidades de procesamiento se organizan y se conectan y en cómo se realiza el aprendizaje. Algunos de los tipos de redes artificiales son [43]:

- **Perceptrón Multicapa (MLP):** Es un modelo con acercamiento al aprendizaje supervisado, que se caracteriza por ser un modelo de red neuronal *feedforward*. La arquitectura de un MLP es de una conexión completa entre las neuronas de todas las capas, en cada capa se pueden

manejar funciones de activación diferentes que permiten hallar relaciones complejas en los datos de entrenamiento. Durante el entrenamiento se emplea el algoritmo de propagación hacia atrás (en inglés, *backpropagation*).

- **Red Neuronal Convolutacional (CNN):** Este tipo de redes imita el comportamiento del córtex visual, ya que se encarga de decodificar la percepción y convertirla en visión. Estos modelos se usan comúnmente en el análisis de imagen, ya que con cada capa es posible extraer características de los píxeles. Usualmente, en este tipo de modelo se emplea la técnica de abandono para evitar el sobre-ajuste.
- **Red Neuronal Recurrente (RNN):** Son modelos que se usan para el procesamiento de secuencias de datos. Este tipo de redes artificiales se caracterizan por tener conexiones retroactivas que les permiten tener información del estado anterior como entradas en el estado actual, teniendo un comportamiento similar a una memoria de corto plazo.

Otros modelos empleados en el aprendizaje profundo son las Máquinas Restringidas de Boltzmann [42], Redes de Creencias Profundas [42], Redes Regenerativas Adversarias[43], Autocodificadores [43] y modelos híbridos [43].

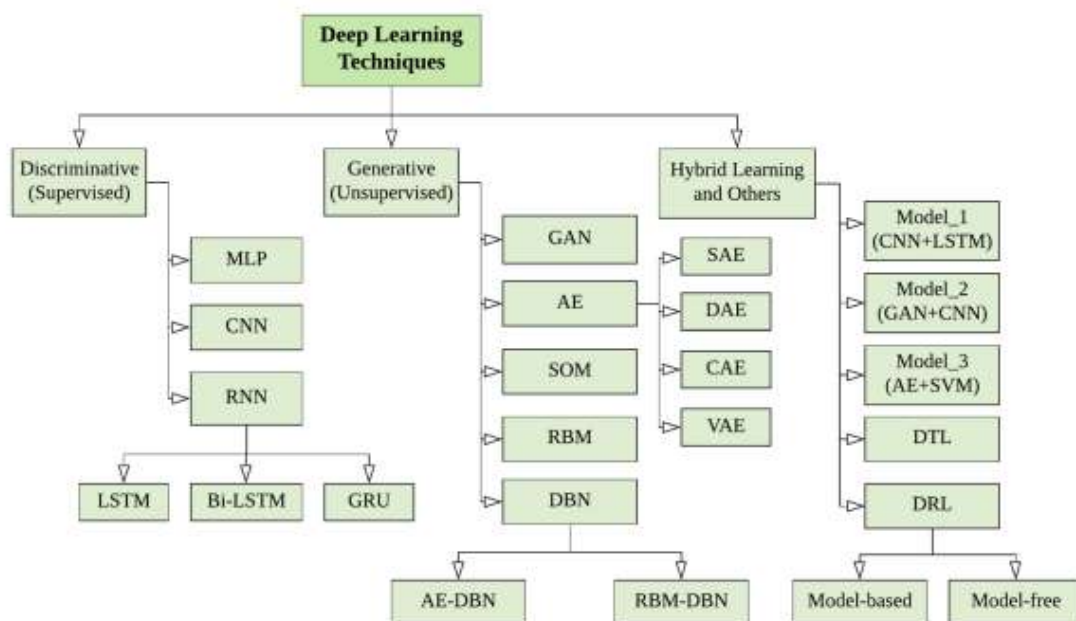


Figura 4.9: Taxonomía de modelos de aprendizaje profundo. [43]

4.5 Fase de aprendizaje

4.5.1 Propagación hacia adelante

La propagación hacia adelante corresponde al proceso mediante el cual se hace la combinación lineal de las entradas con las conexiones sinápticas de cada neurona, en la salida se emplea una función de activación para tener un comportamiento no lineal, dicha salida se propaga hacia la siguiente capa. Este proceso se realiza iterativamente entre cada capa hasta llegar a la salida estimada \hat{y} .

En un perceptrón multicapa con L capas y $L - 2$ capas ocultas, con m siendo la cantidad de neuronas en cada capa oculta, se tiene una matriz de pesos $W^l = w_{ij}^{(l)}$, que se expresan como la conexión sináptica j sobre la neurona i en las capas $l = 2, 3, \dots, L$. Adicional en cada neurona se define un estado interno $a_i^{(l)}$ como la combinación de las entradas ponderadas con los pesos y una salida de neurona z_i^l , que corresponde a la aplicación de la función de activación $f(*)$ en el estado interno de cada neurona. Con ello el proceso de propagación hacia adelante se calcula de la siguiente forma [38]:

- La salida de las neuronas en la capa de entrada corresponde a la propagación de las entradas x_i , por lo que se puede expresar como:

$$z_i^{(l)} = x_i \quad (4.16)$$

- En las capas ocultas l , cada neurona procesa la información recibida de la capa anterior aplicando una función de activación f a la suma de los productos de las activaciones por los pesos $w_{ij}^{(l)}$, es decir:

$$z_i^{(l)} = f \left(\sum_{j=1}^{m-1} w_{ij}^{l-1} z_j^{l-1} + w_{0j}^l \right) \longrightarrow Z_i^l = f \left(W^{(l-1)} \cdot Z^{(l-1)} \right) \quad (4.17)$$

Para $i = 1, 2, \dots, m$ y $l = 2, 3, \dots, L - 1$.

- Por último en la capa de salida $L - 1$ se expresa la salida y_i en términos de los pesos y las

salidas de cada neurona de la forma:

$$y_i = z_i^L = f \left(\sum_{j=1}^{m-1} w_{ij}^{L-1} z_i^{L-1} + w_{0j}^L \right) \longrightarrow Y = Z^L = f(W^{L-1} \cdot Z^{L-1}) \quad (4.18)$$

4.5.2 Propagación hacia atrás

El entrenamiento en las redes neuronales artificiales corresponde al proceso mediante el cual se hace el ajuste de los pesos para minimizar la diferencia entre la salida estimada con la etiqueta del conjunto de datos de entrenamiento. El método con el cual las ANN aprenden patrones se conoce como propagación hacia atrás o *backpropagation* y consiste en la propagación del error desde la salida hacia las neuronas de las capas anteriores que influyen en la salida [38], de manera tal que cada neurona calcula la contribución relativa de error que aporta en la salida \hat{y} .

Este proceso se hace iterativamente capa por capa hasta que cada neurona tenga el cálculo aproximado de la fracción de error relativo, lo que conlleva a una actualización de las conexiones y de los umbrales hasta conseguir la minimización de la función de error. Algunas características del algoritmo se listan en [52]:

- Es capaz de predecir salidas tanto continuas como multiclase.
- Tiene la capacidad de permitir al modelo aprender de patrones más complejos entre los datos o características de entrada y las etiquetas o salidas observables.
- Pese a ser un algoritmo altamente usado en modelos de ANN, presenta algunas desventajas como el alto costo computacional cuando el modelo se hace más complejo, la susceptibilidad al sobre-ajuste con los datos de entrenamiento y que los resultados son difíciles de interpretar, ya que no existe una forma de determinar el aporte de cada neurona.

Para estimar la actualización de los pesos, es necesario calcular el gradiente del error ∇E con el fin de minimizarlo, este proceso se conoce como regla delta. Como punto de partida, se tiene la función de error $E(w)$ expresada en la ecuación (4.4) y la composición de funciones del estado interno de la neurona a^l y de la salida de cada neurona z^l .

Dado que la salida \hat{y}_k esta formada por la combinación de las salidas de cada neurona de las capas anteriores, al momento de estimar el error $E(w)$, se encuentra que este a su vez se compone

de z^l y de a^l , de tal forma que se puede expresar como: $E[z^l(a^l)]$. Con el algoritmo de propagación hacía atrás se desea calcular la variación en el error con respecto a los parámetros de la red neuronal artificial, es decir, los pesos y los sesgos.

4

Teniendo en mente ello, se tiene que para las capas de la red se calcula el ∇E de la siguiente forma:

- Para la capa de salida:

$$\frac{\partial E}{\partial w^L} = \frac{\partial E}{\partial z^L} \frac{\partial z^L}{\partial a^L} \frac{\partial a^L}{\partial w^L} \rightarrow \frac{\partial E}{\partial w^L} = \delta^L z^{L-1} \quad (4.19)$$

- Para las capas ocultas, es necesario retropropagar el error desde la salida del modelo de red neuronal:

$$\frac{\partial E}{\partial w^{l-1}} = \delta^{l-1} z^{l-2} \quad \text{con} \quad l = 2, 3, \dots, L-1 \quad (4.20)$$

- Finalmente, en la capa de entrada se tiene:

$$\frac{\partial E}{\partial w^{(1)}} = \delta^{(1)} x_i \quad (4.21)$$

Siendo δ^* una medida de la contribución de cada peso en la función de error, de manera tal que con cada iteración se ajusta el error imputado de cada neurona hasta conseguir la minimización de la función $E(w)$ con optimizadores como el descenso del gradiente, ADAM o RMSProp.

Así pues, se presenta un diagrama de bloques que representa la forma en la que las redes neuronales aprenden los patrones que permiten relacionar la entrada X con las etiquetas Y , haciendo iterativamente un proceso de ajuste de pesos para reducir el error y aumentar la precisión en la regresión o en la clasificación:

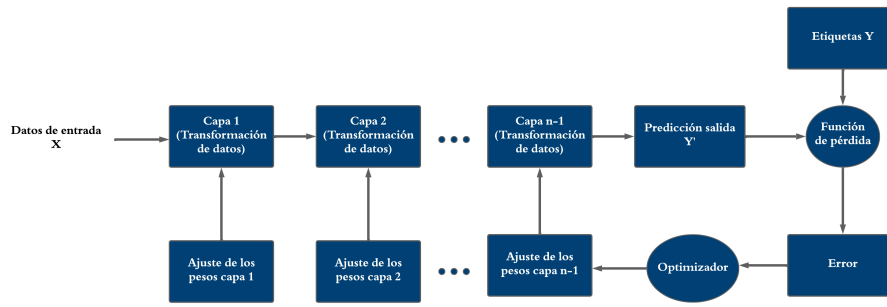


Figura 4.10: Diagrama de bloques del aprendizaje en una ANN. (Autoría propia)

4.6 Error cuadrático medio

Dentro de las métricas empleadas para evaluar el desempeño de los modelos de aprendizaje automático y aprendizaje profundo se encuentra el error cuadrático medio, el cual permite calcular la proximidad entre los resultados predichos por el modelo y las etiquetas. La expresión (4.22) permite calcular este error:

$$\text{MSE} = \frac{1}{n} \sum_{i=0}^n (y_i - \hat{y}_i)^2 \quad (4.22)$$

Donde n es la cantidad de datos, y_i es el valor real y \hat{y}_i es el valor estimado en la salida del modelo de la red neuronal.

Arquitecturas de redes neuronales artificiales

Índice

5.1	Perceptrón Multicapa (MLP)	74
5.2	Redes neuronales convolucionales	75
5.2.1	Componentes	75
5.2.2	Hiper-parámetros	78
5.3	Redes neuronales recurrentes	79
5.4	Derivaciones en las arquitecturas de CNN	81
5.4.1	Autoencoder	81
5.4.2	Red <i>denoising</i> (DnCNN)	82

Las redes neuronales artificiales han llegado para solventar diferentes problemáticas asociadas con el tratamiento y análisis de grandes conjuntos de datos, así como también ser usadas en reconocimientos de imágenes, sonidos, clasificación de objetos o regresiones de series de tiempo. El perceptrón simple fue el primer modelo de neurona artificial, este modelo fue desarrollado por el psicólogo estadounidense Frank Rosenblatt en 1958 y su estructura estaba representada con dos capas: la de entrada, en la cual se hacía la combinación lineal de los pesos w_i con las entradas x_i , esta capa se encarga de propagar la información sin procesarla; y la capa de salida, donde se obtenía una clasificación lineal binaria de acuerdo con una condición de umbral o sesgo, que se iba ajustando iterativamente [57].

Un esquema que representa el concepto del perceptrón simple es el de la figura (4.3), así pues,

con una función de activación como el limitador duro, se tenían dos clases, las cuales se hallaban a partir de la siguiente expresión:

$$y = \begin{cases} 0 & \text{si } \sum_i w_i x_i < \theta \\ 1 & \text{si } \sum_i w_i x_i \geq \theta \end{cases} \quad (5.1)$$

Con w_i los pesos de la neurona, x_i las entradas del modelo y θ el umbral o sesgo que se usa para realizar la separación lineal. El algoritmo requería de la inicialización aleatoria de los pesos y de algún punto con coordenadas x_i, y_i , tras ello se evaluaba la condición de error entre la predicción y las etiquetas con $\hat{y}_i \neq y_i$, cuando se cumplía se hacia la actualización de pesos con el descenso de gradiente, tal como se presenta en la ecuación (5.2):

$$w_i^{k+1} = w_i^k - \mu \frac{dE}{dw_i} \longrightarrow w_i^{k+1} = w_i^k + \mu (y_i - \hat{y}_i) \cdot x_i \quad (5.2)$$

En donde μ es la tasa de aprendizaje. El algoritmo se detenía cuando no había una actualización de pesos, lo que indica que el error es mínimo. Así que en la salida del \hat{y}_i se tenían dos estados clases separadas linealmente. De este algoritmo surgieron algunas derivaciones como el perceptrón con bolsillo¹ o el LMS Adaline².

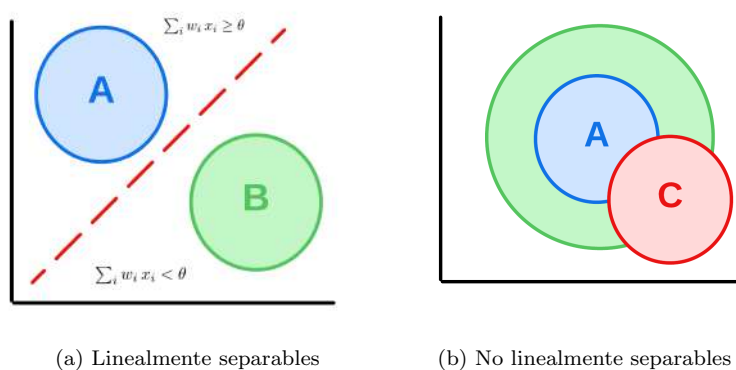


Figura 5.1: Diferencia entre los clasificadores. (Autoría propia)

Dadas las limitaciones presentes en el perceptrón simple, era necesario plantear nuevas arquitecturas que permitieran ajustarse con diversas estructuras de datos como imágenes, señales o sonido,

¹Algoritmo basado en el modelo del perceptrón simple mediante el cual se almacenaban las mejores soluciones w_i , con el fin de converger en menos tiempo.

²El LMS Adaline es un algoritmo que busca reducir el error cuadrático, permitiendo la minimización del ruido en la salida estimada.

adicionalmente, surgió la necesidad de que contaran con una mayor robustez para extraer patrones complejos de los conjuntos de datos. Fue por este motivo que se desarrollaron algoritmos de aprendizaje profundo como el perceptrón multicapa (1986), las redes neuronales convolucionales (1987) o las redes recurrentes (década de los 90), los cuales incluían reglas de aprendizaje para ajustar los parámetros de las ANN que garantizaban un óptimo desempeño con problemas no lineales o con datos no estructurados.

5.1 Perceptrón Multicapa (MLP)

El MLP es un modelo de red neuronal artificial que se compone de varias neuronas que son agrupadas por capas, las conexiones son hacia adelante (*feedforward*) y el aprendizaje es supervisado [38]. Esta arquitectura es considerada como un aproximador universal, debido a su capacidad de aproximar cualquier función continua en la salida \hat{y} .

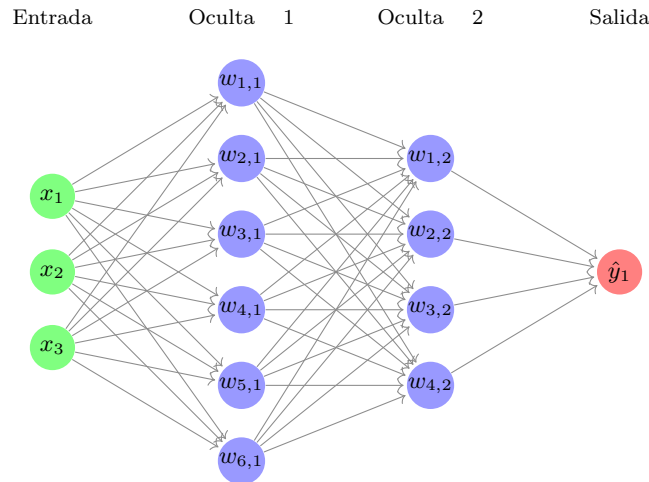


Figura 5.2: Ejemplo de un MLP. (Autoría Propia)

Un MLP es un algoritmo útil para hallar relaciones no lineales o patrones complejos entre los datos de entrada x_i y las salidas y_i . Se considera que entre mayor es el número de neuronas o de capas, mayor profundidad van a tener las redes neuronales, sin embargo, si el modelo se hace más complejo puede presentar sobre ajuste, lo que dificulta la capacidad de generalización con un nuevo conjunto de datos.

En [58] se listan algunas características del perceptrón multicapa, las cuales son:

- Se trata de una estructura altamente lineal.

- Presenta tolerancia a fallos.
- El sistema es capaz de establecer una relación entre dos conjuntos de datos.
- Posibilidad de hacer una implementación en hardware.

Adicionalmente, para la determinación de los parámetros w_{ij} se tienen las fases: propagación hacia adelante, en donde se hace la combinación de las entradas con los pesos y la propagación hacia atrás, mediante el cual se hace la actualización de los pesos con el descenso del gradiente de la función de error.

5.2 Redes neuronales convolucionales

El aprendizaje profundo proporciona un marco muy poderoso para el aprendizaje supervisado. Las arquitecturas de redes profundas han hecho cambios importantes en el desarrollo de la inteligencia artificial, al proveer estructuras y nuevos algoritmos para el aprendizaje que permiten aumentar el número de capas y facilitar más flexibilidad a los modelos para el reconocimiento de patrones en los conjuntos de datos [59].

Para el procesamiento de imágenes se usan las CNN, que son un modelo donde las neuronas corresponden a campos receptivos de forma similar a las neuronas de la corteza visual primaria de un cerebro biológico. La red se compone de múltiples capas. En el principio se encuentra la fase de extracción de características compuesta de neuronas convolucionales y de reducción, a medida que se avanza en la red se disminuyen las dimensiones activando características cada vez más complejas. Al final se encuentran neuronas simples para realizar la clasificación [59].

5.2.1 Componentes

Una CNN está compuesta por una capa de entrada, una capa de salida y muchas capas intermedias ocultas, estas capas realizan operaciones que alteran los datos con el objetivo de aprender características específicas de dichos datos, las 3 capas más frecuentes son: convolución, ReLU, y pooling. Después de aprender las características en varias capas, la arquitectura de una CNN pasa a la clasificación o a la regresión [60].

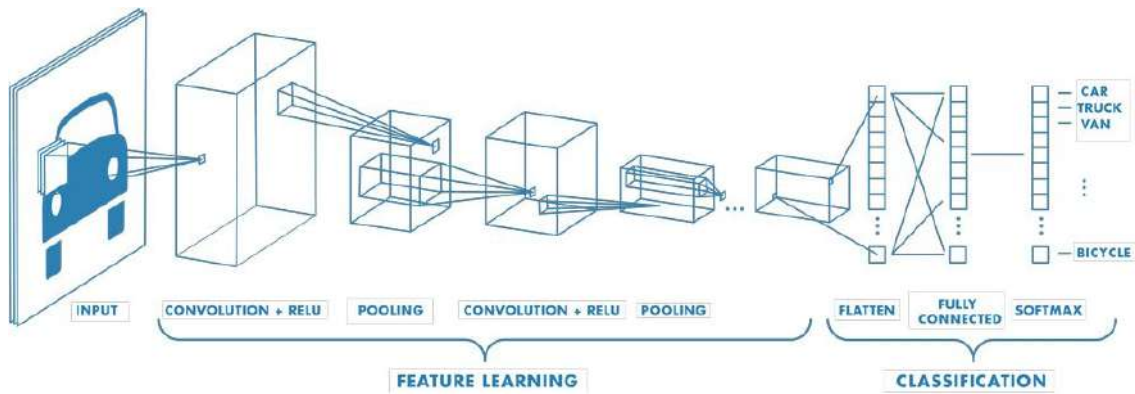


Figura 5.3: Topología de una red CNN para clasificación. [60]

5

Las capas de una CNN se listan enseguida:

- **Capa convolucional:** Su principal propósito es extraer características de una imagen. Consiste de un conjunto de filtros entrenables que realizan producto punto con los valores de la capa precedente. En la práctica, los valores de los filtros son aprendidos para su activación al encontrar ciertas características. Al colocarse en cascada se obtienen diferentes niveles de abstracción [61] que pueden interpretarse de forma gráfica por el nivel de complejidad, tamaño o número de filtros. En la figura (5.4) se muestra un ejemplo de cómo se ven gráficamente los filtros convolucionales.

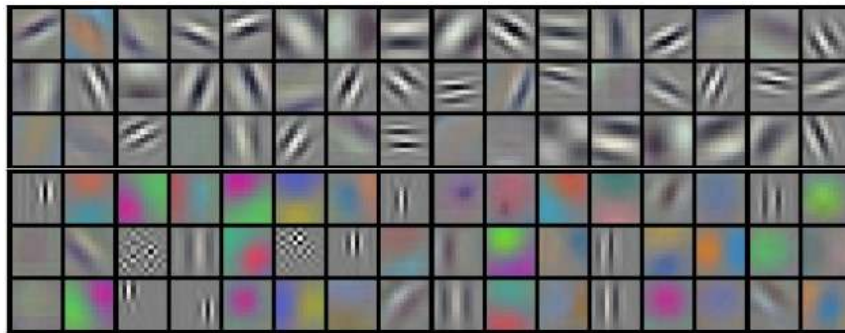


Figura 5.4: Filtros entrenables en una capa convolucional. [62]

- **Capa de agrupación:** Por su nombre en inglés *pooling*, es un algoritmo utilizado para reducir las dimensiones de las capas anteriores, especialmente después de las capas de convolución, con el objetivo de disminuir los tiempos de procesamiento reteniendo la información más importante. Es común insertar periódicamente una capa de agrupación entre capas convolucionales sucesivas. Su función es reducir progresivamente el tamaño espacial de la representación como se aprecia en la figura (5.5) para disminuir la cantidad de parámetros y el cálculo en la red

y, por lo tanto, también controlar el sobreajuste. La capa de agrupación opera independientemente y redimensiona la entrada espacialmente, utilizando la operación máxima [62].

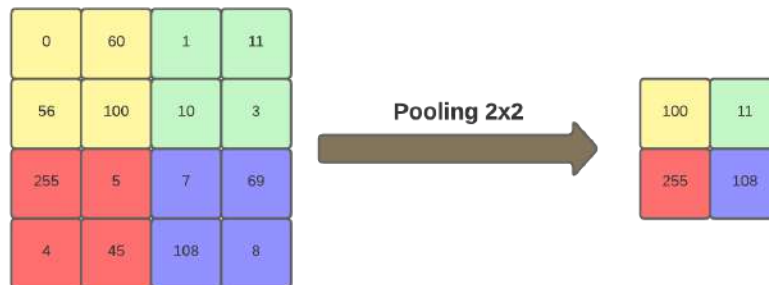


Figura 5.5: Ejemplo de funcionamiento de la capa de agrupación. (Autoría propia)

- **Capa ReLU:** Son utilizados después de cada convolución, en esta capa se realiza una operación que reemplaza los valores negativos por cero y su propósito es agregar no linealidad al modelo, eliminando la relación proporcional entre la entrada y salida. Es la función de activación comúnmente usada en modelos de aprendizaje profundo, esta retorna cero si recibe una entrada negativa, para cualquier valor positivo a la entrada retorna el valor original [61].
- **Capa de normalización:** La capa de normalización de la respuesta local realiza una especie de inhibición de los valores de la capa anterior, al normalizar las regiones de entrada locales. Esta capa es útil cuando se trata con neuronas ReLU, debido a que estas tienen activaciones ilimitadas y es necesario normalizarlas, esto con el fin de detectar características de alta frecuencia con una gran respuesta. A la vez, amortigua las respuestas uniformemente grandes en cualquier vecindario local. Si todos los valores son grandes, la normalización de esos valores disminuirá todos ellos. Básicamente, se fomenta algún tipo de inhibición y se estimulan las neuronas con activaciones relativamente más grandes [63].
- **Capa totalmente conectada:** Generalmente se encuentra en las últimas capas y realiza la clasificación basada en las características extraídas por las capas de convolución y las de pooling. En esta capa todos los nodos están conectados con la capa precedente y trata la entrada como un vector simple lo que produce una salida en forma de un solo vector de acuerdo a la función de salida y al tamaño establecido.
- **Capa de abandono o dropout:** Esta técnica como lo menciona [64] busca reducir el sobreentrenamiento de la red, el cual es el principal problema en las redes neuronales artificiales y en especial en las CNN, por medio del cálculo de un promedio se omiten o desactivan un porcentaje de las neuronas (visibles u ocultas) como se aprecia en la figura (5.6), dicha técnica

ha permitido optimizar los tiempos de entrenamiento de las redes y aumentar su fiabilidad en el porcentaje de acierto.

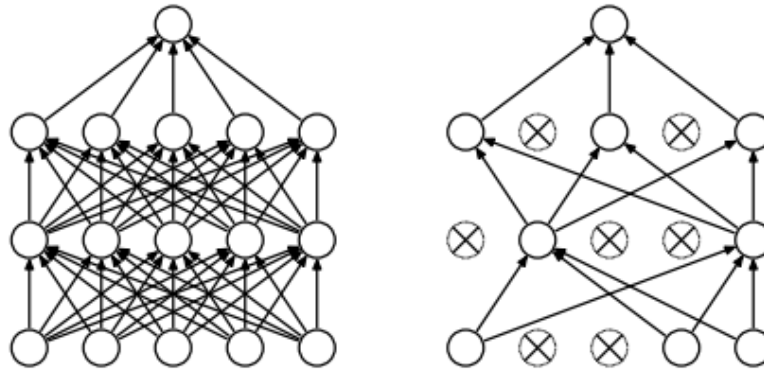


Figura 5.6: Ejemplo de la capa de abandono. [64]

Las redes neuronales profundas contienen múltiples capas ocultas no lineales y esto las convierte en modelos muy expresivos que pueden aprender relaciones muy complicadas entre sus entradas y salidas. Sin embargo, con datos de entrenamiento limitados, muchas de estas complicadas relaciones serán el resultado del ruido de muestreo, por lo que existirán en el conjunto de entrenamiento, pero no en los datos de prueba reales, incluso si se extraen de la misma distribución. Esto provoca un exceso de ajuste y se han desarrollado muchos métodos para reducirlo [64].

- **Capa final:** Esta capa se selecciona de acuerdo a la aplicación de la CNN, ya que se puede emplear para la regresión o la clasificación, esta última suele ir acompañada de la función softmax³ la cual arroja la probabilidad de ocurrencia de cada una de las clases.

5.2.2 Hiper-parámetros

En el apartado anterior, se presentaron algunas de las principales capas que se usan en el diseño de un modelo de CNN, sin embargo, de igual forma como pasan con otras arquitecturas, es necesario tener en cuenta algunos parámetros adicionales previos al inicio del entrenamiento, para su selección no hay un criterio en concreto. En [42] Y [65] se listan:

- **Tamaño del filtro:** Corresponde a las dimensiones del filtro que se desliza sobre la imagen

³Es una función de activación que convierte un conjunto de valores en una distribución de probabilidad, aplicando una función exponencial a cada valor de entrada y normaliza dividiendo por la suma de las exponenciales

o datos de entrada, con el cual se aplica la operación convolución. Sea un filtro $h[n]$ con dimensiones $2k \times 2k$ y unos datos de entrada X en una escala bidimensional, en donde cada píxel tiene coordenadas (x, y) la operación convolución se expresa como [65]:

$$O[x, y] = \sum_{i=-a}^a \sum_{j=-b}^b h[i, j] X[x - i, y - j] \quad (5.3)$$

Los tamaños de filtros que se suelen usar son de: 3×3 , 5×5 y 7×7 . El tamaño del filtro esta directamente relacionado con la cantidad de parámetros entrenables, lo cual supone un aumento en la complejidad de la CNN. Variarlo, permite extraer características a diferentes escalas (locales o generales).

- **Número de filtros:** Se denota como F_d y es el número de filtros que se usan para extraer características diferentes de la imagen o datos de entrada en cada capa convolucional.
- **Stride:** Es un parámetro que se usa para determinar el desplazamiento de píxeles tanto vertical como horizontalmente del filtro o *kernel*.
- **Relleno de ceros:** Se usa para completar de ceros los bordes de las imágenes o mapas de características que se extraen de cada capa convolucional, se suele usar para mantener las dimensiones de la entrada.
- **Tamaño de la agrupación:** Determina el tamaño de la ventana que se ajusta para realizar la operación de *maxpooling* (figura 5.5).

5.3 Redes neuronales recurrentes

En la topología de una red neuronal como el MLP (figura 5.2) se evidencia la ausencia de memoria, por lo cual, con conjuntos de datos secuenciales se dificulta el proceso de aprendizaje para hallar las relaciones y patrones complejos entre la entrada y las etiquetas. Ante este inconveniente, es necesario que en las conexiones entre las neuronas existan retroalimentaciones, que permitan capturar activamente las dependencias secuenciales y temporales de la información que pasa a través de la red [66].

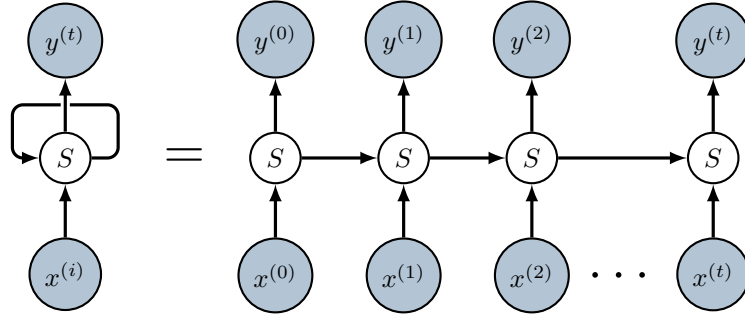


Figura 5.7: Modelado de una neurona recurrente. (Autoría propia)

5

Como se puede observar en la figura (5.7), las neuronas S comparten sesgos y pesos de la conexión anterior, las salidas y_t dependen del procesamiento de la red del conjunto de datos de entrada x_i en donde $i \in [0, t]$, el número de bloques esta relacionado con la longitud de la secuencia que se tiene en la entrada del modelo neuronal [67]. Así pues, existen diferentes topologías de las RNN como:

- **Vector a Vector:** Se tiene una entrada x_0 y una salida y_0 que se puede propagar a través de varias neuronas para aprender los patrones que relacionan al conjunto de datos. Un ejemplo de esta aplicación se encuentra en el análisis de series de tiempo para efectuar predicciones.
- **Vector a secuencia:** Se maneja una sola secuencia en la entrada x_0 y se generan múltiples salidas y_t , este tipo de RNN tienen un uso común en la generación de textos a partir de palabras claves.
- **Secuencia a vector:** Un conjunto de entradas x_i producen una salida y_0 tras pasar por las diferentes neuronas del modelo computacional. Es común encontrarse con estos modelos con los problemas de clasificación de sentimientos de acuerdo a una información ingresada.
- **Secuencia a secuencia:** Por último, este modelo de RNN implica que se tienen varias secuencias tanto a la entrada x_i como a la salida y_t de la red neuronal. Se usa comúnmente en las traducciones de textos.

La recurrencia en una RNN, se puede expresar en términos de los estados ocultos S en el instante t con la expresión (5.4):

$$S^{(t)} = f(W_H S^{(t-1)} + W_X x^{(t)} + b_H) \quad (5.4)$$

Siendo W_H la matriz de los pesos de cómputo recurrentes, $S^{(t-1)}$ es el estado oculto anterior, W_x es la matriz de pesos de la capa oculta del modelo, X son los datos de entrada, ya sean vectores o

secuencias y b_H son los umbrales o sesgos de los estados ocultos.

La salida $y_{(t)}$ se estima con la siguiente ecuación:

$$y^{(t)} = g(W_Y S^{(t)} + b_Y) \quad (5.5)$$

Donde W_Y es la matriz de pesos de la capa de salida, $S^{(t)}$ es el estado oculto actual y b_Y es el bias de la capa de salida.

5.4 Derivaciones en las arquitecturas de CNN

Las derivaciones son arquitecturas que se forman por la combinación de los modelos principales de aprendizaje profundo, estos dotan de una mayor profundidad (cantidad de capas) a los modelos neuronales, por lo cual son útiles para abordar problemas específicos que requieren de un mayor grado de precisión, ya que su funcionamiento se basa completamente en las ventajas que el MLP, las CNN o las RNN traen consigo. Existen múltiples derivaciones, entre las que destacan los modelos pre-entrenados, los *autoencoders* o para en este documento en concreto la red *denoising*.

5.4.1 Autoencoder

Es un modelo de aprendizaje con el cual se busca comprimir la información para extraer características latentes de los datos de entrada, para ser posteriormente recreados a la salida del modelo neuronal. Para ello, lo que hace en el modelo en el entrenamiento es aprender las características de los datos X y eliminar la información no útil (ruido en la entrada), lo cual permite que este red neuronal profunda sea útil para eliminar el ruido de las imágenes.

En la figura (5.8) se presenta un diagrama del modelo, en el cual se tiene 3 partes [68]:

- Las capas de codificación se usan para transformar las imágenes a la entrada en una representación comprimida de las características. Se compone principalmente de CNN, las cuales se le van variando filtros y deslizamientos hasta reducir el las dimensiones originales y generar diversos mapas de características. La información comprimida, contiene patrones y estructuras de los datos de entrada X .

- La capa intermedia es una representación sobre como se logran reducir las dimensiones espaciales de los datos de entrada, son las características latentes con las cuales se pueden recrear los datos X en la salida del modelo neuronal.
- Por último, se tienen las capas de decodificación, estas son responsables de reconstruir las imágenes con los patrones y estructuras importantes de los datos de entrenamiento. Consisten en CNN transpuestas que realizan las operaciones inversas a las capas de codificación.

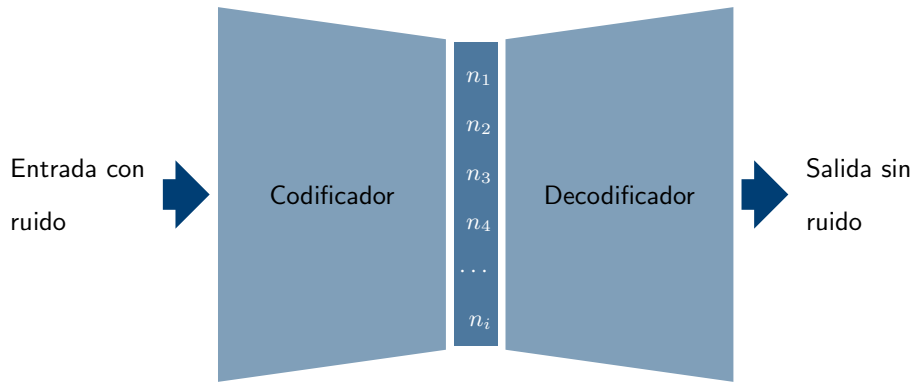


Figura 5.8: Estructura de una red Autoencoder. (Autoría propia)

5.4.2 Red *denoising* (DnCNN)

Este modelo de red neuronal basa su funcionamiento en el aprendizaje residual y la normalización por lotes, para que en la salida se obtenga el ruido que tiene la imagen en la entrada, todo ello con el fin de realizar una operación resta entre X y el ruido η . En la figura (5.9) se presenta la arquitectura de la red *denoising*:

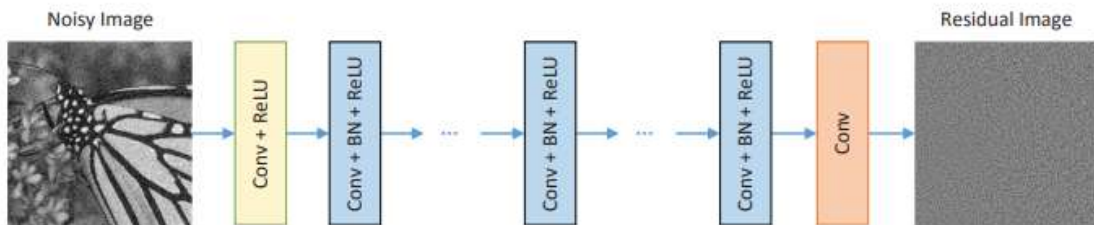


Figura 5.9: Arquitectura de una red DnCNN. [69]

Como se aprecia, se recurre al uso de capas convolucionales con la función de activación ReLU y el proceso de normalización por lotes. En [69] se encuentra que este tipo de modelos recurren

al aprendizaje residual, dadas las facilidades que hay por parte de los modelos CNN de descartar características o estructuras irrelevantes en el proceso de aprendizaje, así pues, conocer esa información permite recrear ruido a la salida del modelo neuronal. Adicional a ello, se usa *batch normalization* para acelerar el proceso de aprendizaje durante el entrenamiento, ya que calcule el descenso del gradiente en pequeñas unidades de información.

Con ello se tiene que los datos de entrada del modelo son de la forma $X_t = x_t + \eta_t$ y la salida que se genera es $y_t \approx \eta_t$, por lo cual, basta con aplicar una resta entre X_t y y_t para conseguir una imagen sin ruido, además, cuenta con la posibilidad de tener redes más profundas, con lo cual, pueden ser usadas en problemas específicos en donde los patrones y estructuras de los datos, sean más complejos y requieran de mayor procesamiento para garantizar una precisión alta en la salida.

Estado del arte

La estimación de canal es un proceso fundamental en las comunicaciones inalámbricas ya que permite optimizar la calidad y la confiabilidad de las comunicaciones entre los usuarios de una red. Determinar las condiciones del canal posibilita adaptar la transmisión para tener un sistema de comunicaciones confiables, aunado a ello mejora el rendimiento y la eficiencia espectral, ya que permite determinar qué variaciones como: tasas de transmisión, modulación o tipo de codificación se pueden emplear para que la información llegué al destino correctamente. Por su importancia, la estimación del canal se desarrolla desde diferentes enfoques, como transformadas discretas, con modelos estadísticos para soportar estimaciones a ciegas o con algoritmos de aprendizaje profundo.

Existen estimaciones de canal ciegas, porque no se requieren de señales piloto, tal como es descrito por [70] en donde se hace uso de la estadística bayesiana y más concretamente de la estimación máximo a posteriori para obtener la matriz de correlación del canal y la información transmitida. Un acercamiento similar se tiene con una estimación semiciega en [71] en donde se asume una distribución Gaussiana para los símbolos desconocidos de los datos, para ello se usa la cota de Cramér-Rao que contempla cuánta información de un parámetro desconocido se puede extraer de una muestra. Como se puede detallar, la estimación ciega se basa en la estadística de la señal recibida para inferir las condiciones del canal.

Otro enfoque que se le da a la estimación de canales es a partir del seguimiento de señales piloto, estas permiten inferir al receptor cómo se ha visto afectada la señal dadas las condiciones

que tiene el canal de comunicaciones. En [72] se hace uso de la Transformada Discreta de Coseno, aprovechando las propiedades del producto convolución y la elección de señales piloto que simplifiquen los cálculos en el dominio de las transformadas discretas trigonométricas. Otra estimación de canal es la realizada en [73] donde se usa la Transformada de Fourier Fraccional, el canal se estima por amplitudes y retardos en las señales piloto, que tienen la frecuencia modulada pulsada. En [74] se usa la transformada de Karhunen-Loèv y obtienen un desempeño del SER similar al calculado con el canal ideal. Otros estimadores usados son variaciones de las técnicas convencionales: LS y MMSE, como en [75] con el estimador M , que posee mayor robustez comparado con el estimador por mínimos cuadrados y en [76] con el LMMSE (MMSE Lineal) con ventana deslizante en el tiempo, que se desarrolló en condiciones de alta movilidad por parte del usuario. Por último, una alternativa es a partir del sensado compresivo [77] que ayuda a mitigar el impacto de las señales piloto en el ancho de banda, haciendo un envío de información eficiente y una estimación simplificada.

El uso del aprendizaje profundo para la estimación de canales se ha desarrollado con diferentes arquitecturas de redes neuronales artificiales, la premisa del uso de esta alternativa está relacionada con la robustez de los modelos y la capacidad de adaptarse a diferentes condiciones de canales. En [78] y [79] se usan las redes neuronales profundas, para ello el problema se aborda como una regresión donde se desea eliminar el ruido en el canal de comunicaciones, para ello se interpola primero con las señales piloto y luego se predice con dicha información. Un proceso similar es el desarrollado en [8], en la que se opta por elegir un modelo de red completamente conectada para estimar los coeficientes del canal desconocidos, tomando la numerología OFDM de $\mu = 1$ y utilizando el modelo de canal CDL.

Los modelos de redes neuronales convolucionales son ampliamente usados para extraer las características canal como si este fuera una imagen o un espacio tridimensional. El método empleado en [80] es usar las señales piloto CSI RS, donde encuentran que el desempeño del algoritmo mejora con más épocas y la tasa de aprendizaje reducida. La flexibilidad de los modelos CNN permiten variar parámetros de la transmisión, como en [81] donde se usa la codificación polar, que mejora la corrección de errores y en [82] se estiman canales en cascada. En ambos artículos el desempeño de dichos modelos supera notablemente a los resultados del LS y MMSE, en términos de MSE y BER. Existen variaciones como la CNN dual [83] en la que se emplean dos redes para reducir el ruido en el canal y la contaminación en las señales piloto aprovechando el dominio espacio-frecuencial; también se usa la combinación de una red convolucional con una recurrente [84], la CNN se usa para

la estimación del canal y la recurrente para capturar la correlación en el canal de comunicaciones.

Existen alternativas de modelos como el uso de las redes recurrentes como: Red de Memoria a Corto Plazo Largo (LSTM) [85] en la cual se toman como entradas al modelo, el historial de respuestas al impulso que tiene el canal y biLSTM (LSTM bidireccional) [86] en la que usan el optimizados SGD y la función de pérdida Huber, con lo que obtienen un mejor desempeño en la predicción y en la capacidad del canal, siendo 10 dB superior a las obtenidas con LSTM, MMSE y LS. El uso de las LSTM bidireccionales mejora la calidad en la estimación y un menor BER, como se sugiere en [87], donde compara frente a los modelos convencionales, CNN y redes completamente conectadas. Por último, en [88] se usa una red neuronal residual ResNET para la estimación de canal tanto de enlace de bajada como se subida, teniendo una diferencia entre 0.5 dB y 1 dB con respecto al canal ideal y aumentando la tasa de transferencia efectiva en un 25%.

Desarrollo de ingeniería

Índice

7.1	Generalidades	87
7.2	Configuración del sistema de comunicaciones en MATLAB	89
7.2.1	Consideraciones de valor	89
7.2.2	Funciones de interés	91
7.2.3	Transmisión	92
7.2.4	Canal de comunicaciones	93
7.2.5	Recepción	93
7.3	Generación de los datos de entrenamiento	96
7.4	Estimadores de canal con ANN	98
7.4.1	Preprocesamiento de los datos	100
7.4.2	Modelos con redes neuronales convolucionales	102
7.4.3	Autoencoder	105
7.4.4	Red neuronal tipo <i>denoising</i>	107

7.1 Generalidades

En el área de las telecomunicaciones es indispensable garantizar que la información que es transmitida a través de cualquier medio llegue de manera confiable, rápida e íntegra al receptor, efectuando una serie de procesos que permiten mitigar los efectos que se presentan en el medio de transmisión, como los mencionados en el *capítulo 3*: ruido, dispersión, atenuación y efectos relativos al desplazamiento del usuario.

Al hablar de un sistema como el desplegado en la quinta generación de las comunicaciones móviles, es necesario tener en cuenta ciertos aspectos fundamentales como: la inserción de pilotos para la estimación de canal (DM-RS) y de pilotos para la compensación de fase (PT-RS). Así mismo, es importante considerar las técnicas de modulación soportadas (ver sección 2.6), la numerología OFDM, las configuraciones de la portadora y organización en la cuadrícula de recursos.

Con esto presente, en este capítulo del documento se aborda el desarrollo del trabajo a partir de 3 secciones: en la primera se presenta el diseño y construcción del sistema de comunicaciones en el software de MATLAB, configurando el transmisor (inserción de pilotos, información, modulación), el canal de comunicaciones de tipo TDL considerando los factores inherentes al medio y la configuración del receptor (estimación de canal, ecualización, compensación de fase, entre otros); en la segunda parte se generan los datos de entrenamiento de acuerdo con las condiciones fijadas en el canal TDL, haciendo necesario que el conjunto de entrenamiento contenga todos los escenarios posibles en un canal para que haya una buena generalización del modelo y que este no presente un sobre-ajuste en las estimaciones de canal; por último, se crean los modelos de CNN, *autoencoder* y *red denoising* fijando los hiper-parámetros, de manera tal que se puedan medir los desempeños de la estimación a partir de métricas como el MSE.

De acuerdo con lo descrito anteriormente, se obtiene el siguiente diagrama de bloques que permite conocer la estructura que tiene el sistema de comunicaciones realizado.

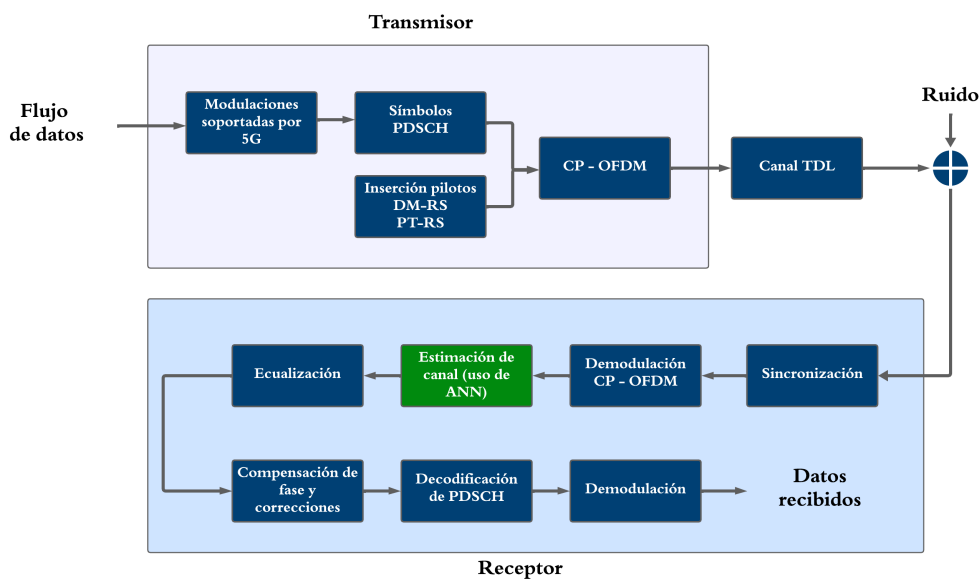


Figura 7.1: Diagrama de bloques del sistema desarrollado. (Autoría propia)

7.2 Configuración del sistema de comunicaciones en MATLAB

En esta primera etapa de desarrollo se elaboró un sistema de quinta generación desde cero empleando algunas utilidades disponibles en el software de MATLAB R2023b esta construcción implica varias etapas, que incluyen la generación de señales, la modelización de canales, la implementación de algoritmos de procesamiento de señales, y la evaluación del rendimiento del sistema. El desarrollo en módulos del sistema permite la simulación de un escenario real de comunicaciones móviles, esta construcción está dividida en los principales elementos de cualquier sistema de comunicaciones: transmisor, receptor y canal de transmisión.

En primer lugar se debe construir un escenario ideal y completo con el fin de poder identificar la sección en donde implementar las redes neuronales, para lograr esto se propuso el esquema de comunicaciones de la figura 7.1. Es importante resaltar que el sistema está elaborado a nivel de capa física (Transmisión Binaria) por lo que en los siguientes apartados se explica el modelado de cada uno de los bloques.

7.2.1 Consideraciones de valor

A continuación se muestran algunos aspectos importantes dentro de la estructura del programa, consideraciones en las medidas y demás comentarios que permiten comprender la organización, funcionamiento y proceso de desarrollo del trabajo realizado.

Existe una estructura de datos que contiene todos los elementos dentro de la simulación denominada **Pam_sim**, esta estructura almacena desde los objetos de configuración hasta los bits recibidos, por lo que se hace la notación que cuando se haga referencia a un elemento como la frecuencia central $f_c = 5$ GHz realmente presenta la configuración de la frecuencia de la portadora dentro de la simulación y podrá ser encontrada por medio de invocación del objeto **Pam_sim.fc**.

Para la obtención de las métricas de evaluación presentadas en la sección 8 y 8.4 se tuvieron las siguientes consideraciones :

- El cálculo del MSE y EVM se realiza por cada ranura procesada, por lo que para cada iteración se tendrá un vector de tamaño $(10 \frac{\text{subtramas}}{\text{trama}} \times N \text{ tramas}) \times \text{len}(SNR)$, en el caso de la simulación presentada se obtuvieron vectores de 70×26 (SNR de 0 dB-25 dB para 7

tramas). Para las gráficas presentadas en las secciones mencionadas no es necesario obtener esta métrica en cada ranura, por lo que se obtiene un valor promedio a lo largo del tiempo de transmisión obtenido en cada iteración del barrido SNR, es decir, si se envían 7 tramas con una numerología $\mu = 0$ se tendrá un total de 70 ms teóricos de transmisión para cada iteración de SNR y de este intervalo de tiempo se obtendrá el valor promedio y es el que se presenta en las gráficas.

- El término “*tiempo de estimación*” hace referencia al valor normalizado que tarda en ejecutarse la instrucción de estimación de canal dentro de la secuencia de procesamiento en MATLAB, este valor se obtiene empleando las funciones *tic* y *toc*. Cuando se hace referencia al valor normalizado significa que es un valor llevado a lo teórico ya que la utilización de estas funciones permite ver el tiempo de ejecución de la instrucción en función de la máquina encargada de realizar la simulación del programa, por lo que con el fin de evitar sesgar los resultados a la capacidad de procesamiento de los equipos utilizados, se normalizó este valor considerando el tiempo de ejecución de estimación la estimación como 1 ms, esto se obtiene dividiendo todo el vector de tiempo de salida de la función *tic* entre el vector de tiempo de estimación ideal, de modo que se podría expresar como :

$$t_{norm} = \frac{t_{ANN}}{t_{ideal}} \quad (7.1)$$

Estructura Programa

El programa depende de una gran cantidad de módulos por lo que el árbol de directorios del programa es el siguiente :

- **Estimation** : Contiene los ficheros que implementan las funciones de estimaciones de canal de cada uno de los modelos propuestos.
- **Functions** : Contiene todas las funciones necesarias para el funcionamiento del programa, generación de variables de forma dinámica, configuración de parámetros iniciales, sistema de transmisión, recepción, emulación de canal, obtención y almacenamiento de las métricas de desempeño.
- **Generation** : Contiene los archivos de configuración y entrenamiento de las ANN.
- **Graph_Functions** : Funciones que permiten la visualización y monitoreo de diversas etapas en el sistema de comunicaciones, visualización de constelaciones, canal en tiempo real y métricas

de desempeño

- **Models** : Contiene los modelos de ANN contruidos en su versión `.mat` .
- **Outputs** : Contiene todas las salida generadas por el programa exportando la estructura **Pam_sim** en un archivo `.mat` para su posterior procesamiento.

7.2.2 Funciones de interés

Existen diversas funciones que en conjunto permiten el correcto funcionamiento del sistema, las principales se listan enseguida:

Simulation_Configuration()

Esta función tiene como finalidad cargar los parámetros con los que se ejecuta el entorno de simulación para evaluar los rendimientos de los modelos CNN diseñados. Los parámetros que se configuran son:

- Generación de las señales piloto DM-RS y PT-RS para efectuar los procesos de estimación de canal y compensación de fase respectivamente.
- Generación de los símbolos PDSCH y su posterior asignación en la rejilla de recursos.
- Configuración de la portadora de acuerdo con las normas establecidas para 5G, modulación de los bits transmitidos y ajuste de las variables asociadas a la técnica de acceso múltiple CP-OFDM.
- Configuración de los *slots* o ranuras de acuerdo al tipo de información que es enviada, admitiendo tres formatos: tramas, bits o imágenes.
- Parametrización del canal TDL teniendo en cuenta los perfiles de retardo (tablas 3.1-3.5), la dispersión del retardo y el efecto Doppler presente en el canal debido a la movilidad del usuario.
- Selección del número de antenas en el transmisor y el receptor. Para este trabajo se opta por un sistema SISO.

Variable_Generation()

Permite generar n variables del nombre que se desee en función de las características cambiantes del sistema como: tipos de canal, modelos para la estimación, matrices de almacenamiento de datos y demás, consiste en la creación de espacios de memoria para los modelos de ANN seleccionados.

Finalmente, cabe resaltar que el programa al ser definido de forma modular agrega dinamismo en el cambio de parámetros y condiciones propias del sistema de comunicaciones, por lo que todos los valores son modificables y adaptables, los valores seleccionados para la simulación representan la base de construcción de un sistema 5G y permiten la implementación de todos los escenarios posibles.

7.2.3 Transmisión

El transmisor es la parte del sistema de comunicaciones encargada de convertir la información de entrada (secuencia aleatoria de datos o imágenes) en una señal portadora que asegure la calidad de la transmisión, el alcance y cobertura, que no se vea afectada en gran medida por los fenómenos presentes en el medio de transmisión. En el contexto del sistema de comunicaciones 5G desarrollado en MATLAB, el transmisor consta de varias etapas:

- **Generación de Señales:** En esta etapa, se generan las señales moduladas que llevarán la información. Esto puede incluir la generación de símbolos de modulación digital soportadas por 5G, como QPSK, 16-QAM, 64-QAM y 256-QAM.
- **Generación rejilla de recursos :** Esta etapa incluye los bloques de inserción de pilotos requeridos para la estimación de canal DM-RS y de estimación de fase PT-RS así como los encargados de transportar la información PDSCH. Como se mencionó, esta primera versión de sistema de comunicaciones está construida en capa 1 (física) por lo que estos 3 elementos son los únicos que se incluyen dentro de la rejilla de recursos.
- **CP-OFDM :** La siguiente etapa corresponde al bloque OFDM, en este apartado en primer lugar se define la cantidad de espacios de la rejilla de recursos los cuales son definidos a partir de la numerología elegida y el ancho de banda disponible, el cual define la cantidad de bloques de recursos. En este caso se implementó la numerología $\mu = 0$ de 5G la cual define un $SCS = 15$ kHz, se usan 52 bloques de recursos, lo que conlleva a operar con un ancho de banda de 9.36 MHz. Las configuraciones se fijan en la función de **Simulation Configuration()**,

por lo cual son las condiciones con las que se van a evaluar los rendimientos del sistema con los estimadores diseñados.

Terminando esta etapa de transmisión se configura el prefijo cíclico el cual en este caso se define como normal pero el cual puede ser de tipo *extendido* cuando se opte por usar la numerología $\mu = 2$ con un $SCS = 60$ kHz. Por último, es posible a partir de la rejilla de recursos, realizar la modulación CP-OFDM y generar la forma de onda que será transmitida por el canal de comunicaciones tipo TDL.

7.2.4 Canal de comunicaciones

Implementado dentro de la función **Receiver()** pero configurado a nivel de inicialización de parámetros del sistema. Se deben considerar 2 etapas diferentes en este apartado: la primera consistió en el envío de datos por medio del canal seleccionado lo que afecta directamente la forma de onda transmitida, a nivel de simulación esto implica el cambio del arreglo adicionando posiciones al final que emulan el *delay* agregado por el canal y la resincronización en función de los diversos perfiles, a esta señal de onda se le agrega un ruido AWGN en función del valor de la SNR en decibelios.

7.2.5 Recepción

Implementado dentro de la función **Receiver()** consta de varios bloques importantes y abarca la gran parte del desarrollo elaborado en este trabajo. En primer lugar se realiza la sincronización temporal de los datos, esta función permite eliminar el retardo introducido por el canal de comunicaciones y recuperar la forma de onda lista para ser demodulada, en el siguiente paso, por medio de la función **nrOFDMDemodulate()** se obtiene la rejilla de recursos. Cuando la rejilla de recursos quede incompleta debido a la asignación en bruto de recursos, se hace un relleno con ceros para evitar problemas en la estimación real del canal.

En este momento de la recepción se realiza el proceso de estimación de canal, por lo que entrarán en juego las funciones definidas en el directorio **Estimation**, una vez se obtenga la rejilla de recursos recibida se envía a las múltiples funciones de estimación, estas funciones cumplirán todas el mismo objetivo, eliminar el ruido y efectos producidos por el canal, la secuencia de pasos para lograr esto varía en función del tipo de estimación que se desea realizar, sin embargo se sigue una estructura general, para el procesamiento con cualquier red neuronal primero es necesario realizar un pre-procesamiento de los datos por medio de una interpolación lineal, a partir de la cual se conforma

la información de entrada a la red y por medio de la función **predict()** de MATLAB se obtiene la rejilla de recursos estimada, estos datos luego son procesados de modo tal que sean compatibles con la estructura de datos de un sistema de comunicaciones tradicional, expresando los datos de forma compleja, después de esto es posible obtener las métricas de desempeño como el MSE.

Por medio de la función **nrExtractResources()** es posible obtener los símbolos específicos en función a los índices, es decir que en este apartado se extraen los datos recibidos de la modulación digital, estos datos pasan a un proceso de ecualización por medio de la función **nrEqualizeMMSE()**, con los símbolos ecualizados se tiene la opción de realizar la compensación de fase por medio de los pilotos PT-RS y gracias a la construcción de la función **Phase.Compensantion()**, finalmente gracias a la función **nrPDSCHDecode()** se obtienen los bits transmitidos, por lo que en este punto es posible obtener métricas como el EVM y BER. Por medio del siguiente algoritmo se da mas claridad sobre la estructura general del programa:

Algoritmo 1 Función Configuración sistema 5G

```

1: procedure CONFIGURACIÓN SISTEMA
2:   portadora  $\leftarrow$  nrCarrierConfig                                ▷ Objeto para Portadora
3:   portadora.NSizeGrid  $\leftarrow$  52                                ▷ 52 bloques de recurso
4:   portadora.SubcarrierSpacing  $\leftarrow$  15                        ▷ Numerología ( $\mu = 0$ )
5:   profile  $\leftarrow$  [TDL-A, TDL-B, TDL-C, TDL-D, TDL-E]
6:   Se realiza la configuración de los pilotos PTRS y DMRS.
7:   v  $\leftarrow$  User_Velocity                                          ▷ Asignación de la velocidad 0-150 km/h
8:   Mod  $\leftarrow$  [QPSK, 16-QAM, 64-QAM, 256-QAM]                  ▷ Selección Modulación Digital
9:   SNRdB  $\leftarrow$  [0 : 1 : 25]                                    ▷ Intervalo relacion señal a ruido

```

Algoritmo 2 Función Configuración Modelos de estimación

```

1: procedure SELECCIÓN MODELOS
2:   Se realiza la carga de modelos y se establecen cuales serán utilizados
3:   Pam_sim.CNNEstimation  $\leftarrow$  true                                ▷ Estimación con CNN
4:   Pam_sim.Autoencoder_Estimation  $\leftarrow$  true                    ▷ Estimación con Autoencoder
5:   Pam_sim.EstimacionPractica  $\leftarrow$  true                          ▷ Estimación practica
6:   Pam_sim.Denoising_Estimation  $\leftarrow$  true                      ▷ Estimación Red Denoising
7:   Configuración del tipo de datos a transmitir
8:   Pam_sim.Transmission_IMG  $\leftarrow$  true                            ▷ Transmisión de una imagen
9:   Pam_sim.Num_Frames  $\leftarrow$  15                                ▷ Asignación de la cantidad de frames a transmitir

```

Algoritmo 3 Flujo de procesamiento simulaciones múltiples

```

1: procedure FUNCIÓN DE TRANSMISIÓN Y RECEPCIÓN
2:   for canal = profile do                                ▷ Iteración en función a los canales elegidos
3:     for modulaciones = Mod do                            ▷ Iteración en función a las Modulaciones elegidas
4:       for Velo = v do                                    ▷ Iteración en función a las Velocidades elegidas
5:         Función encargada de la asignación de configuraciones :
6:         Pam_sim ← Simulation_Configuration(Pam_sim)
7:         for snr = SNRdB do                                ▷ Iteración en función a las SNR elegidas
8:           Función encargada de la asignación de espacios de memoria :
9:           Pam_sim ← Variables_initialization(Pam_sim,snr)
10:          for Indice_Ranura = 0 : Pam_sim.slots - 1 do
11:            Bucle de Envío para N Slots
12:            Inicio Medidas de tiempo y evaluadores de rendimiento
13:            Pam_sim ← Transmitter(Pam_sim)
14:            Pam_sim ← Receiver(Pam_sim)
15:            Estimación de canales :
16:            Pam_sim ← Lineal_Estimation(Pam_sim)
17:            Pam_sim ← Perfect_Estimation(Pam_sim)
18:            Pam_sim ← CNN_Estimation(Pam_sim,Pam_sim.estimacionRNA_1)
19:            Pam_sim ← Autoencoder_Estimation(estimacionRNA_Autoencoder)
20:            Pam_sim ← Denoising_Estimation(estimacionRNA_Denoising)
21:            Pam_sim ← Practical_Estimation(Pam_sim)
22:            Pam_sim ← Metrics_Slot(Pam_sim)
23:            Pam_sim ← Matrix_metrics(Pam_sim)
24:            Pam_sim ← Save_parameters_ScriptFinal(Pam_sim)

```

El Algoritmo 3 presenta el flujo del programa principal ejecutado para la obtención de resultados mostrados en la sección siguiente, existe otro algoritmo que permite la ejecución en tiempo real, la diferencia recae en la asignación de variables, las cuales se hacen en tiempo real permitiendo emular comportamientos reales y cambios en el sistema, para lograr esto se implemento el mismo algoritmo dentro de un bucle while que envía infinitamente slots.

7.3 Generación de los datos de entrenamiento

La generación de los datos para el entrenamiento de los estimadores con redes neuronales artificiales, resulta una parte fundamental para garantizar el correcto funcionamiento de los modelos ante las variaciones que se pueden presentar en el canal y por ende, las variaciones que surgen en las señales piloto DM-RS. Así pues, es necesario tener en cuenta que se parte de la configuración inicial del canal de comunicaciones presentada en la sección anterior, sin embargo, dada la necesidad de incluir los escenarios posibles se tiene en cuenta que:

- Los perfiles del canal TDL que se usan son: TDL-A, TDL-B, TDL-C, TDL-D y TDL-E.
- La modulación en el canal físico PDSCH es QPSK.
- El generador aleatorio se fija en un intervalo de $[1001, 2000]$.
- La dispersión del retardo esta en el rango de $[1 \text{ ns}, 300 \text{ ns}]$.
- El valor máximo de la frecuencia del efecto Doppler $f_{D_{\max}}$ se tiene entre $[1 \text{ Hz}, 500 \text{ Hz}]$, lo cual supone un rango de velocidades del UE entre 0.21 km/h y 107.92 km/h .
- La relación señal a ruido se toma como el rango comprendido entre 0 dB y 25 dB .

A continuación, se define la longitud del conjunto de datos de entrenamiento N para generar un proceso iterativo, en el cual se cambien los valores aleatoriamente de los intervalos definidos anteriormente, dotando al conjunto de entrenamiento de un mayor número de muestras diferentes, permitiendo que el modelo tenga una mayor capacidad de generalización cuando sea entrenado con diferentes condiciones del canal de comunicaciones. Los datos de entrenamiento quedan de la siguiente forma:

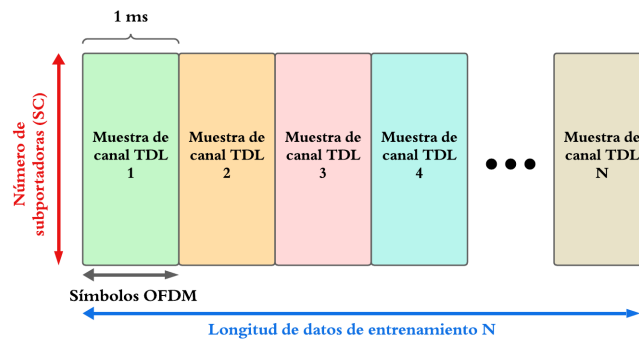


Figura 7.2: Distribución del conjunto de datos. (Autoría propia)

Como se presenta en la figura (7.2), cada muestra del conjunto de entrenamiento tiene una dimensión en función al número de subportadoras, de los símbolos OFDM y de las señales piloto que son enviadas por el canal, por lo que se componen de $[RB \times 12, \text{Símbolos}, s \in \mathbb{C}]$. Debido a que las muestras tienen una duración de 1 ms y que la cantidad de bloques de recursos es 52, se tiene que las dimensiones del conjunto de datos es $[624, 14, 2, N]$.

Conociendo las dimensiones de los datos para el entrenamiento de los estimadores de canal con ANN, es necesario tener en cuenta cuáles van a ser las entradas X y las etiquetas Y . Para X se toma el canal tal cual es recibido tras realizar la demodulación CP-OFDM, en donde se toman las coordenadas de las señales piloto DM-RS, se extraen y se aplica una interpolación lineal para generar un canal de acuerdo con dicha información, en donde hay una alta presencia de ruido; por otro lado, en Y se genera una estimación de canal perfecta, mitigando los efectos aplicados por el medio a través del cual se hace el envío de las señales piloto. Con ello, se desarrolla el siguiente algoritmo:

Algoritmo 4 Función para a generación de datos

1: procedure GENERACIÓN DE DATOS	▷ Entrada N	▷ Salida (X, Y)
2: N : Longitud de datos		
3: $profile \leftarrow [TDL-A, TDL-B, TDL-C, TDL-D, TDL-E]$		
4: Se realiza la configuración previa en el transmisor. Se toman las variables:		
5: $index$: Ubicación DM-RS, $symbols$: Símbolos DM-RS y $carrier$: Portadora		
6: $X \leftarrow zeros([624, 14, 2, N])$	$Y \leftarrow zeros([624, 14, 2, N])$	▷ Arreglos de ceros
7: for $i = 1 : N$ do		
8: $seed \leftarrow rand([1001, 2000])$	$delay \leftarrow rand([1, 300]) \times 10^{-9}$	
9: $doppler \leftarrow rand([1, 500])$	$SNR \leftarrow rand([0, 25])$	$delay_profile \leftarrow rand(profile)$
10: $channelTDL(seed, delay, doppler, delay_profile)$		
11: tx : Se genera la forma de onda en el transmisor		
12: $rx \leftarrow channelTDL(tx) + noise(SNR)$ ▷ Se agrega ruido en función del SNR a la señal enviada		
13: $timing$: Se hace la sincronización		
14: $grid \leftarrow ofdm_demod(carrier, rx)$		
15: $pilot \leftarrow grid(index) * conj(symbols)$	▷ Se recuperan las señales piloto	
16: $X(:, :, :, i) \leftarrow linear_interpolation(pilot)$	▷ Se hace la interpolación lineal con las señales	
17: $Y(:, :, :, i) \leftarrow$ Estimación ideal	▷ Se hace la estimación ideal	
18: Se almacenan en el equipo los datos X y Y .		

Algunas variaciones que se pueden obtener del conjunto de datos son:

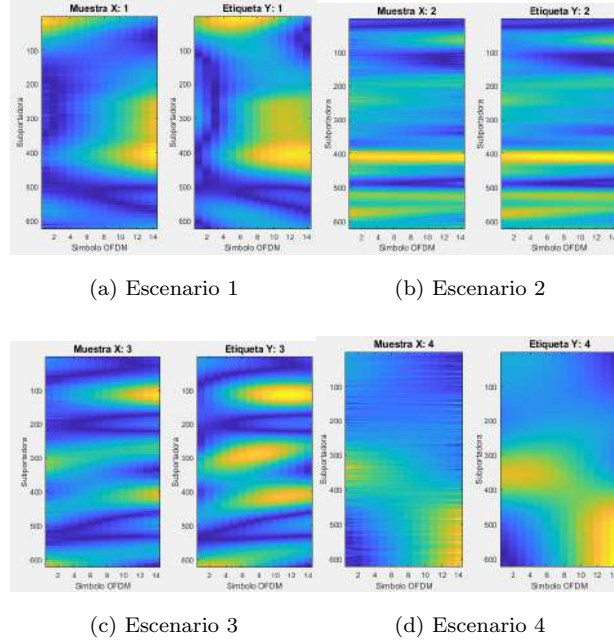


Figura 7.3: Variaciones presentes en los datos de entrenamiento. (Autoría propia)

De la figura (7.3), se identifica que las muestras están totalmente aleatorizadas, lo cual permite que en el entrenamiento, la red no tenga una dependencia hacia un orden en concreto, evitando patrones que dominen al modelo y su calidad de predicción con datos no vistos durante el entrenamiento. Así mismo, algo que se debe considerar es la longitud de los datos N , ya que no puede ser muy alta debido a los altos costos computacionales que esto supone, ocasionando que el modelo neuronal sea más complejo.

7.4 Estimadores de canal con ANN

En la figura (7.1) se presenta el diagrama de bloques del sistema elaborado en este proyecto, con las configuraciones del transmisor, canal inalámbrico y del receptor. Allí se muestra un bloque que corresponde a la estimación de canal con modelos de DL, en este caso en concreto, se recurre al uso de CNN, autoencoder y redes *denoising*, cuyas propiedades permiten abordar la predicción del canal teniendo en cuenta las 3 dimensiones del arreglo generado en la sección anterior: cantidad de sub-portadoras, símbolos OFDM en 1 ms y los valores de las respuestas del canal $s \in \mathbb{C}$ tras haber hecho la demodulación de OFDM.

La manera en la que se aborda la estimación del canal con estos modelos, es similar a la del procesamiento de una imagen RGB, solo que en la última dimensión no se tienen 3 componentes del color, sino 2, correspondientes a $\text{Re}(s)$ e $\text{Im}(s)$. El principal motivo por el cual se opta por realizar estimadores a partir de redes convolucionales, está relacionado con su capacidad de hallar patrones locales con una invariación a la traslación, por lo que una característica detectada por el filtro, puede ser identificada en otra parte de la muestra.

Con lo descrito anteriormente, el estimador del canal se compone de una serie de procesos, los cuales se listan enseguida:

1. Se tiene en la entrada los recursos tras haber hecho la demodulación de CP-OFDM. Se hace la extracción de las señales piloto DM-RS, para determinar las incidencias y efectos que tuvo el canal sobre estas.
2. Con la información de las señales piloto, se hace un proceso de interpolación lineal para llenar la rejilla de recursos en la muestra de 1 ms. Debido a que las señales piloto están afectadas por los fenómenos presentes en el canal inalámbrico, repercuten en la calidad de la estimación, por lo cual es necesario incluir un modelo neuronal que mitigue los impactos y sea capaz de recrear un canal con una mayor precisión en diferentes escenarios.
3. Se incorpora un modelo neuronal que está basado en redes CNN, tiene como entrada la muestra de 1 ms proveniente de la interpolación lineal y entrega una aproximación de la rejilla de recursos.

Una predicción en la salida del modelo neuronal cercana a la rejilla de recursos ideal permite recuperar correctamente la información enviada a través del medio y optimizar la comunicación entre el transmisor Tx y el receptor Rx . En la imagen (7.4) se muestra el diagrama de bloques que representa la operación del estimador de canal:

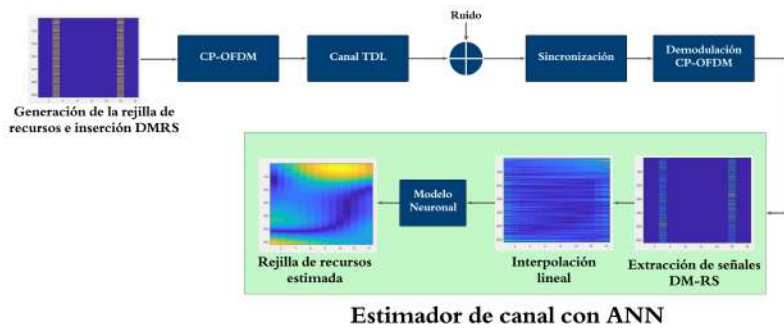


Figura 7.4: Estructura por bloques del estimador de canal. (Autoría propia)

7.4.1 Preprocesamiento de los datos

El preprocesamiento de los datos es un proceso fundamental en el cual se hace la transformación de los datos en bruto en un formato más adecuado y comprensible para el algoritmo del modelo neuronal. Su importancia está relacionada con el ajuste de los pesos cuando la ANN esté en la fase de propagación hacia atrás, ya que cuando hay valores muy altos en el conjunto de datos, los pesos son inestables en cada iteración, impactando directamente en la calidad de la estimación de canal y en el desempeño del algoritmo.

Para el estimador basado en ANN, se hace un ajuste en los datos de entrenamiento, especialmente en la dimensión correspondiente a las respuestas del canal $s \in \mathbb{C}$ y es que al tener un formato en coordenadas rectangulares $a + jb$, es necesario hacer la separación del valor real a y del valor imaginario b . Este procedimiento implica que la cantidad de muestras sea de $2N$, donde la primera mitad es la componente real y la segunda la imaginaria. El reajuste queda:

Algoritmo 5 Preprocesamiento de los datos

- | | | |
|---|--|---|
| 1: procedure PREPROCESAMIENTO | ▷ Entradas: X, Y | ▷ Salidas: $x_{train}, x_{test}, y_{train}, y_{test}$ |
| 2: Se cargan las entradas X y las etiquetas Y | | |
| 3: $ratio \leftarrow 0.75$ | ▷ Porcentaje división de datos para validación cruzada | |
| 4: $X_r \leftarrow Re(X)$ $X_i \leftarrow Im(X)$ | | |
| 5: $Y_r \leftarrow Re(Y)$ $Y_i \leftarrow Im(Y)$ | | |
| 6: Se aplica la concatenación entre los datos reales e imaginarios | | |
| 7: $X \leftarrow [X_r, X_i]$ | | |
| 8: $Y \leftarrow [Y_r, Y_i]$ | | |
| 9: $x_{train}, y_{train}, x_{test}, y_{test} \leftarrow split(X, Y, ratio)$ | | |
-

Este proceso resulta útil, ya que incrementa el conjunto de datos con nueva información, sin necesidad de generarlos de nuevo, ya que las componentes $Re(s)$ e $Im(s)$ poseen propiedades diferentes de acuerdo con las configuraciones hechas al canal. En la salida del estimador se entregan las rejillas estimadas de cada componente, que se concatenan para formar una sola con las dimensiones originales, similar a como se presenta en la figura (7.5).

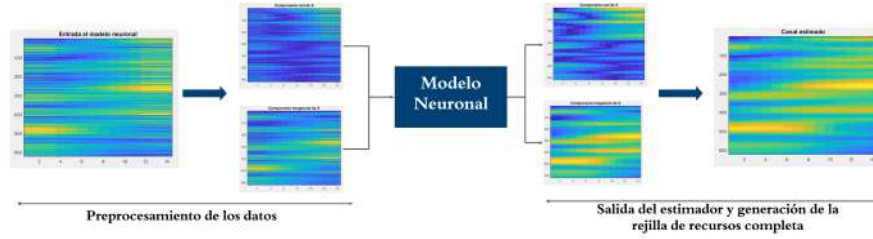


Figura 7.5: Preprocesamiento en la entrada y estimación con modelo neuronal. (Autoría propia)

Finalmente, en la capa de entrada de todos los modelos neuronales, se fijan las dimensiones de entrada en la cantidad de subportadoras, símbolos OFDM y componente única. A esta información se le aplica una normalización centrada en cero, en donde se busca que la media de los datos sea 0 y que la desviación estándar 1, evitando el desplazamiento en la distribución de datos debido a las diversas escalas que pueden presentarse al cambiar las condiciones del canal, conllevando a una optimización en la reducción de la función de error $E(w)$.

Teniendo el preprocesamiento de los datos generados, se configuran los parámetros del entrenamiento de cada modelo, los cuales son:

Parámetro	Valor
Cantidad de datos N	10000
Dimensiones datos	$624 \times 14 \times 1 \times 20000$
Validación cruzada	0.75
Dimensiones x_{train}, y_{train}	$624 \times 14 \times 1 \times 15000$
Dimensiones x_{test}, y_{test}	$624 \times 14 \times 1 \times 5000$
Optimizador	ADAM
Tasa de aprendizaje	3×10^{-4}
Tamaño del lote	32
Épocas	10
Shuffle	En cada época
Función de pérdida	MSE

Cuadro 7.1: Parámetros de entrenamiento de los estimadores de canal

7.4.2 Modelos con redes neuronales convolucionales

Los redes neuronal convolucionales ofrecen una solución a la estimación de canales ya que permiten trabajar con datos con una mayor dimensión, como es el caso de la rejilla de recursos. La importancia en su selección esta en la operación convolución entre el filtro y la muestra del canal, lo cual permite identificar patrones en el canal sin importar las condiciones que este tiene (ruido, desplazamientos, desvanecimientos, entre otros), logrando mitigar los efectos presentes en las señales piloto generados por el medio de transmisión.

La configuración de las CNN se puede hacer de múltiples formas variando algunos hiperparámetros como: cantidad de filtros, tamaño del filtro, número de capas, desplazamiento del filtro, capa de agrupación y relleno de ceros. En esta parte del documento, se parte de un modelo base hecho por MathWorks [89], que va a ser referido como **Modelo CNN 1**, el cual se usa para realizar la estimación de canal con la interpolación de las señales piloto DM-RS recibidas, la topología de este modelo tenía la siguiente configuración:

Modelo	Capas	Función	(Filtro, Número de filtros, Padding, Stride)
Modelo CNN 1	Entrada 624x14x1	-	-
	Convolucional 2D	ReLU	(9, 64, 4,1)
	Convolucional 2D	ReLU	(5,64,2,1)
	Convolucional 2D	ReLU	(5,64,2,1)
	Convolucional 2D	ReLU	(5,32,2,1)
	Convolucional 2D	ReLU	(5,1,2,1)
	Salida	Lineal	-

Cuadro 7.2: Topología de la referencia: Modelo CNN 1. Tomado de [89]

Con la configuración presentada en la tabla (7.2), se procede a diseñar nuevos modelos, con parámetros diferentes, con el propósito de hallar un estimador de canal CNN que funcione de una mejor forma en los diversos entornos del canal TDL, para ello se hacen variaciones en todos los hiperparámetros, así como la adición de otras capas como *Max Pooling*, lo que resulta en los siguientes modelos neuronales:

Modelo	Capas	Función	(Filtro, Número de filtros, Padding, Stride)
Modelo CNN 2	Entrada 624x14x1	-	-
	Convolutacional 2D	ReLU	(3, 16, Same,1)
	Max Pooling 2D	-	(2, 1 ,Same,1)
	Convolutacional 2D	ReLU	(3,8,Same,1)
	Max Pooling 2D	-	(2, 1 ,Same,1)
	Convolutacional 2D	ReLU	(3,8,Same,1)
	Max Pooling 2D	-	(2, 1 ,Same,1)
	Convolutacional 2D	ReLU	(3,1,Same,1)
	Salida	Lineal	-

Cuadro 7.3: Topología del modelo CNN 2

Modelo	Capas	Función	(Filtro, Número de filtros, Padding, Stride)
Modelo CNN 3	Entrada 624x14x1	-	-
	Convolutacional 2D	ReLU	(7, 16, 2,1)
	Max Pooling 2D	-	(4, 1 ,2,1)
	Convolutacional 2D	ReLU	(7,8,2,1)
	Max Pooling 2D	-	(4, 1 ,2,1)
	Convolutacional 2D	ReLU	(7,8,2,1)
	Max Pooling 2D	-	(4, 1 ,2,1)
	Convolutacional 2D	ReLU	(7,1,4,1)
	Salida	Lineal	-

Cuadro 7.4: Topología del modelo CNN 3

Modelo	Capas	Función	(Filtro, Número de filtros, Padding, Stride)
Modelo CNN 4	Entrada 624x14x1	-	-
	Convolutacional 2D	ReLU	(7, 16, Same,1)
	Max Pooling 2D	-	(4, 1 ,2,1)
	Convolutacional 2D	ReLU	(7,8,Same,1)
	Max Pooling 2D	-	(4, 1 ,2,1)
	Convolutacional 2D	ReLU	(7,8,Same,1)
	Max Pooling 2D	-	(7, 1 ,2,1)
	Convolutacional 2D	ReLU	(7,1,Same,1)
	Salida	Lineal	-

Cuadro 7.5: Topología del modelo CNN 4

Modelo	Capas	Función	(Filtro, Número de filtros, Padding, Stride)
Modelo CNN 5	Entrada 624x14x1	-	-
	Convolutacional 2D	ReLU	(7, 64, 4,1)
	Convolutacional 2D	ReLU	(7, 64, 2,1)
	Convolutacional 2D	ReLU	(7, 64, 2,1)
	Convolutacional 2D	ReLU	(7, 32, 2,1)
	Convolutacional 2D	ReLU	(7,1,5,1)
	Salida	Lineal	-

Cuadro 7.6: Topología del modelo CNN 5

Otro tipo de redes CNN que se usan son aquellas en las que se reduce el número de capas, con el propósito de verificar si estos modelos realizan una buena estimación teniendo en cuenta una reducción en la robustez y complejidad. En los siguientes 3 modelos, se varía el tamaño del filtro, la cantidad de filtros y el tipo de *padding*, resultando en las topologías presentadas a continuación:

Modelo	Capas	Función	(Filtro, Número de filtros, Padding, Stride)
Modelo CNN 6	Entrada 624x14x1	-	-
	Convolutacional 2D	ReLU	(9, 64, 4,1)
	Convolutacional 2D	ReLU	(5, 64, 2,1)
	Convolutacional 2D	ReLU	(5, 1, 2,1)
	Salida	Lineal	-

Cuadro 7.7: Topología del modelo CNN 6

Modelo	Capas	Función	(Filtro, Número de filtros, Padding, Stride)
Modelo CNN 7	Entrada 624x14x1	-	-
	Convolutacional 2D	ReLU	(5, 64, 4,1)
	Convolutacional 2D	ReLU	(7, 128, 2,1)
	Convolutacional 2D	ReLU	(7, 1, 2,1)
	Salida	Lineal	-

Cuadro 7.8: Topología del modelo CNN 7

Modelo	Capas	Función	(Filtro, Número de filtros, Padding, Stride)
Modelo CNN 8	Entrada 624x14x1	-	-
	Convolucional 2D	ReLU	(13, 64, 4,1)
	Convolucional 2D	ReLU	(3, 128, 2,1)
	Convolucional 2D	ReLU	(3, 1, 2,1)
	Salida	Lineal	-

Cuadro 7.9: Topología del modelo CNN 8

7.4.3 Autoencoder

El autoencoder convolucional es un modelo que es capaz de reducir las dimensiones de las características extraídas y a partir de estas, recrea una salida aproximada a la etiqueta, de esta forma se pueden eliminar redundancias en los datos como el ruido, las traslación o la rotación. Una ventaja que presenta este modelo respecto a una CNN estándar es contar con una mayor robustez porque comprime la información, derivando en una mejor calidad de predicción, pero suponiendo un mayor costo computacional para su entrenamiento.

En la figura (5.8) se observa que este modelo esta compuesto por dos etapas: codificación y decodificación. Con esta consideración, es necesario tener en cuenta las salidas que se tienen en cada etapa para que la dimensión sea igual a la del conjunto de datos de entrenamiento y validación, ya que en la decodificación se usan capas *convolucionales transpuestas* que hacen la operaciones inversas.

Se diseñan 3 modelos de autoencoders convolucionales, cada uno con propiedades diferentes ya sea en la codificación o la decodificación de los datos, de igual manera al escenario descrito con las CNN estándar, se tiene como propósito hallar un modelo que presente las mejores características para estimar el canal adecuadamente. En las capas transpuestas se usa el parámetro de recorte o *cropping* el cual es el inverso del relleno con 0 o *padding*, por lo que en las tablas de cada modelo es necesario tener en cuenta el parámetro al que se hace referencia con respecto a la etapa.

Los modelos de autoencoder son:

Etapa	Capa	Función	(Filtro, Número de filtros, Stride)	Padding/Cropping
Codificación	Entrada 624x14x1	-	-	-
	Convolutacional 2D	ReLU	(2,8,1)	Same
	Max Pooling 2D	-	(2,1,1)	Same
	Convolutacional 2D	ReLU	(2,16,1)	Same
	Max Pooling 2D	-	(2,1,1)	Same
	Convolutacional 2D	ReLU	(2,32,1)	Same
	Max Pooling 2D	-	(2,1,1)	Same
	Convolutacional 2D	ReLU	(2,64,1)	Same
	Max Pooling 2D	-	(2,1,1)	Same
Decodificación	Conv. Transpuesta 2D	ReLU	(2,64,1)	Same
	Conv. Transpuesta 2D	ReLU	(2,32,1)	Same
	Conv. Transpuesta 2D	ReLU	(2,16,1)	Same
	Conv. Transpuesta 2D	ReLU	(2,8,1)	Same
	Convolutacional 2D	ReLU	(1,1,1)	Same
	Salida	Lineal	-	-

Cuadro 7.10: Topología del modelo Autoencoder 1

Etapa	Capa	Función	(Filtro, Número de filtros, Stride)	Padding/Cropping
Codificación	Entrada 624x14x1	-	-	-
	Convolutacional 2D	ReLU	(5,8,1)	Same
	Max Pooling 2D	-	(2,1,1)	Same
	Convolutacional 2D	ReLU	(5,16,1)	Same
	Max Pooling 2D	-	(2,1,1)	Same
	Convolutacional 2D	ReLU	(5,32,1)	Same
	Max Pooling 2D	-	(2,1,1)	Same
	Convolutacional 2D	ReLU	(5,64,1)	Same
	Max Pooling 2D	-	(2,1,1)	Same
Decodificación	Conv. Transpuesta 2D	ReLU	(2,32,1)	Same
	Conv. Transpuesta 2D	ReLU	(2,16,1)	Same
	Conv. Transpuesta 2D	ReLU	(2,8,1)	Same
	Conv. Transpuesta 2D	ReLU	(2,32,1)	Same
	Convolutacional 2D	ReLU	(1,1,1)	Same
	Salida	Lineal	-	-

Cuadro 7.11: Topología del modelo Autoencoder 2

Etapa	Capa	Función	(Filtro, Número de filtros, Stride)	Padding/Cropping
Codificación	Entrada 624x14x1	-	-	-
	Convolutacional 2D	ReLU	(5,8,1)	Same
	Max Pooling 2D	-	(2,1,1)	Same
	Convolutacional 2D	ReLU	(5,16,1)	Same
	Max Pooling 2D	-	(2,1,1)	Same
	Convolutacional 2D	ReLU	(5,32,1)	Same
	Max Pooling 2D	-	(2,1,1)	Same
	Convolutacional 2D	ReLU	(5,64,1)	Same
	Max Pooling 2D	-	(2,1,1)	Same
	Convolutacional 2D	ReLU	(5,128,1)	Same
	Max Pooling 2D	-	(2,1,1)	Same
	Salida	Lineal	-	-
Decodificación	Conv. Transpuesta 2D	ReLU	(5,128,1)	Same
	Conv. Transpuesta 2D	ReLU	(5,32,1)	Same
	Conv. Transpuesta 2D	ReLU	(5,16,1)	Same
	Conv. Transpuesta 2D	ReLU	(5,8,1)	Same
	Conv. Transpuesta 2D	ReLU	(5,32,1)	Same
	Convolutacional 2D	ReLU	(1,1,1)	Same
	Salida	Lineal	-	-

Cuadro 7.12: Topología del modelo Autoencoder 3

7.4.4 Red neuronal tipo *denoising*

Una Red Neuronal Convolutacional Denoising (DnCNN) es un modelo especial que tiene como principal funcionalidad eliminar el ruido en imágenes, por lo cual, para este trabajo resulta útil en la mitigación de este efecto sobre el canal estimado. Una clave de estas redes es el uso de las conexiones residuales, cuya finalidad es la de incluir caminos directos entre las capas de la red, para evitar la degradación de los datos al propagarse hacia adelante. Estos algoritmos de DL tienen una combinación de capas convolucionales y de normalización por lotes, por lo que tienen una mayor profundidad si se comparan con los modelos diseñados previamente.

El software de MATLAB cuenta con una DnCNN pre-entrenada, la cual tiene una dimensión de entrada $50 \times 50 \times 1$, debido a ello, fue necesario crear una nueva capa de entrada que se ajuste con las dimensiones de la cuadrícula de recursos, tras ello se ensambla esta nueva entrada al resto del modelo y se procede a realizar el entrenamiento. La topología se detalla en la tabla (7.13):

Número de capas	Capa	Función	(Filtro, Número de filtros, Stride)	Padding
1	Entrada 64x14x1	-	-	-
19	Convolutacional 2D	ReLU	(3,64,1)	1
	Normalización por lotes	-	64 canales	1
1	Convolutacional 2D	ReLU	(3,1,1)	1
1	Salida	Lineal	-	-

Cuadro 7.13: Topología del modelo DnCNN

Debido a que el modelo presenta un elevado número de capas, con una mayor robustez y complejidad, se opta por realizar el entrenamiento con dos escenarios:

- Tomando los datos generados X y Y para aprender las representaciones del canal estimado perfecto, similar al proceso que se aplica con las CNN estándar y los autoencoder. Este modelo se va a referir a lo largo del documento como **Modelo Denoising 1**
- La otra forma con la que se entrena es variando las etiquetas para que el modelo no estime el canal, sino el ruido. Los datos X tienen un nivel de ruido η , las nuevas etiquetas se obtienen aplicando la resta $X - Y$ implicando que las nuevas etiquetas contengan el ruido η . En la salida se hace $Y = X - \eta$ para reducir el ruido y optimizar la calidad de la estimación. Es referido como **Modelo Denoising 2**.

Índice

8.1	Generalidades	109
8.2	Rendimiento individual de cada modelo	111
8.3	Selección de estimadores de canal ANN	114
8.3.1	Configuración del sistema de comunicaciones y simulación	115
8.3.2	Modelos de redes neuronales convolucionales	116
8.3.3	Modelos de redes Autoencoder	120
8.3.4	Modelos de redes denoising	123
8.4	Comparación entre modelos seleccionados con diferentes escenarios	125
8.4.1	Parámetros de la simulación	125
8.4.2	Canal con perfil de retardo TDL-B (NLoS)	127
8.4.3	Canal con perfil de retardo TDL-E (LoS)	133

8.1 Generalidades

En el entrenamiento de los modelos neuronales, estos realizan el ajuste de los pesos iterativamente para que la salida estimada del canal inalámbrico \hat{Y} se aproxime a las etiquetas Y . Para conocer la calidad de la estimación se usan métricas como el MSE, sin embargo, debido a que se desea evaluar la estimación de canal en un escenario de redes 5G, se incorporan las técnicas de Deep Learning en el diagrama de bloques presentado en la figura (7.1), dando cabida a la obtención de mediciones en el desempeño general del sistema como el BER y el EVM. Otra medida que se realiza es el tiempo en el que la red neuronal es capaz de predecir la salida con nuevos datos, siendo

una métrica relevante, puesto que esta relacionada con la latencia en el sistema de comunicaciones.

Con base en lo mencionado anteriormente, en este capítulo se evalúan los desempeños de los modelos neuronales y del sistema de comunicaciones planteado a partir de cuatro métricas: MSE, BER, EVM y tiempo de predicción. Como entorno de simulación se usa un programa desarrollado en el software de MATLAB, con las librerías de Deep Learning y de 5G, allí se pueden modificar los parámetros del canal con el fin de caracterizar la respuesta que tiene el sistema de comunicaciones. Para abordar los resultados se trabajan tres etapas:

- La primera etapa esta enfocada en la evaluación del rendimiento individual de los estimadores de canal con el 25% de los datos (datos de evaluación: x_{test} , y_{test}). El proceso consiste en comparar las predicciones \hat{Y} de cada muestra con las etiquetas Y de los nuevos datos y calcular el MSE promedio de cada modelo con todo el conjunto de evaluación. También se calcula la latencia del estimador usando el toolbox de MATLAB “Deep Learning HDL Toolbox Support Package for Intel FPGA and SoC Device” [90], desde el cual es posible obtener dicha métrica del desempeño del modelo a partir de la emulación por capas en un procesador dedicado en algoritmos de aprendizaje profundo.
- En la segunda, se obtienen las métricas de desempeño del sistema de comunicaciones con los estimadores diseñados, con el propósito de realizar una comparativa entre las topologías de cada red neuronal artificial y posteriormente seleccionar los mejores modelos de acuerdo con su rendimiento. Para realizar la prueba, se dejan fijas algunas condiciones del canal como el perfil de retardo, la dispersión del retardo, la modulación de los bits para la carga útil en el canal PDSCH, la velocidad del usuario y se hace el envío de bits aleatorios con una duración de 50 ms (5 tramas). La variación aplicada al canal se hace con el nivel de ruido para graficar MSE, BER, EVM y tiempo en función del SNR.
- La tercera etapa, parte de la selección de los modelos que presentaban un mejor rendimiento con las condiciones fijas en el canal de comunicaciones. Teniendo en cuenta ello, se hace una comparativa entre cada modelo variando el tipo de canal, la velocidad del usuario y la modulación aplicada a los bits aleatorios transmitidos en 70 ms (7 tramas). La comparativa incluye la estimación tradicional basada en mínimos cuadrados, una estimación práctica incorporada en una función de MATLAB y la estimación ideal. De igual forma que en la anterior etapa, se hace un barrido de niveles de ruido para graficar las métricas de desempeño en función de dicho parámetro.

8.2 Rendimiento individual de cada modelo

El rendimiento en una red neuronal artificial es la capacidad que tiene el modelo para realizar la tarea a la que fue diseñado, de manera efectiva y eficiente. Evaluar el rendimiento de los estimadores tras haber hecho el entrenamiento con los hiper-parámetros de la tabla (7.1), supone un punto de partida para realizar una comparación de los estimadores en un entorno en el que no se tiene en cuenta la estructura del sistema de comunicaciones, sino únicamente, la calidad de predicción que estos tienen con muestras de 1 ms de duración. En esta primera etapa se extraen dos métricas: MSE con los datos de evaluación y la latencia de la ANN.

El procedimiento para medir el MSE consta principalmente de tomar uno a uno los datos de evaluación x_{test} , realizar la estimación de canal $y_{estimada}$ e ir calculando la función de error respecto a las etiquetas de evaluación y_{test} , debido a que este subconjunto corresponde al 25% del dataset original, se genera un vector de longitud $0.25 N$, por lo que se calcula la media del vector generado. El MSE promedio permite cuantificar la calidad de la predicción con un conjunto de datos no visto.

La otra métrica es la latencia, la cual es el tiempo que le toma a la red neuronal para producir una predicción con una sola muestra de entrada. Para medir la latencia en MATLAB se usa la librería de Deep Learning HDL¹, desde la cual es posible emular el despliegue de la arquitectura del modelo en un procesador dedicado a algoritmos de Deep Learning, permitiendo adquirir los tiempos independientes a los recursos del computador. Para obtener la latencia es necesario seleccionar una referencia de FPGA o de SoC, en este caso se elige el SoC: *AMD Zynq 7000 SoC ZC706 Evaluation Kit*², se fija la frecuencia del reloj en un valor cercano a la del procesador de un computador, para esta instancia, $f_{clock} = 1800 \text{ MHz}$; por último se usa una función incorporada en la librería que calcula la latencia en cada capa y la latencia total, tanto en segundos como en ciclos de reloj.

El rendimiento obtenido con los procedimientos descritos anteriormente, es calculado con cada uno de los estimadores ANN diseñados, por lo que con ellas se puede establecer una relación entre la calidad de la predicción y el tiempo que tarda el modelo en generar una predicción. Los resultados se muestran en el cuadro (8.1):

¹Ejemplo usado como referencia: [Estimate Performance of Deep Learning Network](#)

²Documentación e información sobre [AMD Zynq 7000 SoC ZC706 Evaluation Kit](#)

Tipo de red	Modelo	Función de error (MSE)	Latencia (ms)
Red Neuronal Convolutacional	CNN 1	0.0115	17.13
	CNN 2	0.0359	0.35
	CNN 3	0.0341	0.97
	CNN 4	0.0206	1.32
	CNN 5	0.0108	39.08
	CNN 6	0.0132	7.85
	CNN 7	0.0127	42.96
	CNN 8	0.0150	7.48
Autoencoder Convolutacional	Autoencoder 1	0.0221	19.28
	Autoencoder 2	0.0237	22.21
	Autoencoder 3	0.0096	113.15
CNN de tipo Denoising	Modelo Denoising 1	0.0130	32.52
	Modelo Denoising 2	0.0098	32.52

Cuadro 8.1: Evaluación del desempeño de los modelos con los datos de evaluación

8

Un resultado que se puede determinar fácilmente, es la relación inversa que existe entre la calidad de la predicción y la latencia, por ejemplo, el Autoencoder 3 tiene una mayor latencia, pero el MSE es bajo si se compara con los otros modelos, caso contrario ocurre con las redes CNN2 y CNN3, las cuales son las más rápidas con latencias inferiores a 1 ms, pero suponen un MSE más alto, el principal factor es la complejidad del modelo y algunos hiper-parámetros en la construcción del mismo. Graficando los resultados de la tabla, se pueden apreciar notables diferencias:

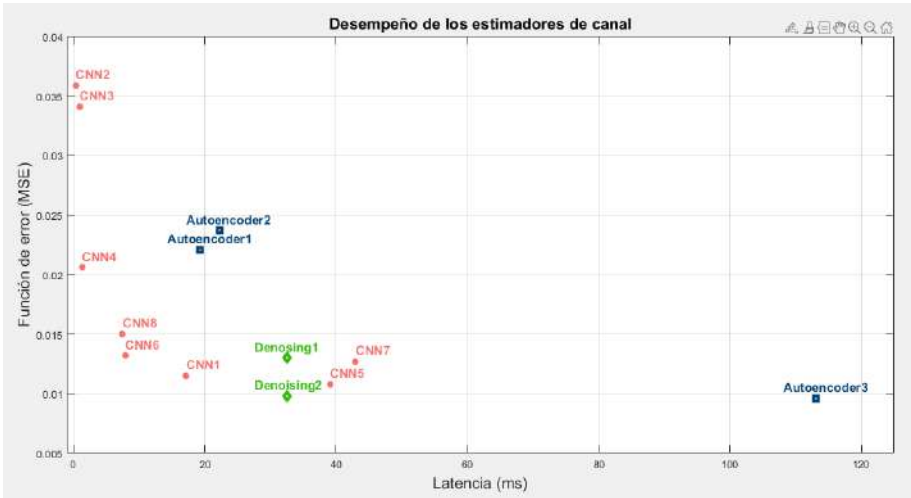


Figura 8.1: Comparación de rendimiento entre los modelos con los datos de evaluación

En la figura (8.1) se pueden contrastar de una mejor forma el desempeño de los modelos diseñados para la estimación de canales, basado en la ubicación de los puntos. En primera instancia, se obtiene que el Autoencoder 3 y el modelo Denoisng 2, son los que tienen una mejor calidad de estimación con valores MSE promedio inferiores a 0.01, sin embargo, al revisar las latencias se encuentra que la red DnCNN es más rápida y esto se debe a la utilización de las capas de **normalización por lotes**, las cuales evitan el desvanecimiento de la información en redes profundas, optimizando el ajuste de pesos durante el entrenamiento.

Comparando el desempeño de las redes CNN estándar, se puede notar que presentan notables variaciones entre si, teniendo diferentes escenarios como redes rápidas o redes con mejor calidad de predicción. Las redes que presentan una mejor calidad de estimación son la CNN1 y la CNN5, sin embargo, la marcada diferencia de latencias del 43% se debe al tamaño de los filtros, ya que estos influyen en la cantidad de información que se toma de la muestra en cada capa, los filtros de menor tamaño son más selectos que los filtros de un mayor tamaño. Las redes CNN2 y CNN3, como se mencionó anteriormente, son más rápidas pero con valores MSE promedios más altos comparados con sus pares. De las redes CNN6 y CNN8 se puede observar que la latencia es similar debido a que tienen la misma profundidad, sin embargo, la CNN6 tiene un MSE promedio inferior, lo cual se debe al tamaño de los filtros y el número de filtros configurados. Por último, comparando el rendimiento de la CNN3 con la CNN4 se evidencia un valor similar en la latencia dada la misma estructura por capas, pero claramente el modelo 4 de las CNN estándar tiene una mayor calidad de predicción, ocurre como consecuencia de la selección del hiper-parámetro *padding* en igual, lo cual preserva la información de los bordes y mantiene la resolución espacial.

En las redes Autoencoder se encuentra marcada la diferencia entre el modelo 1 y 2 con el modelo 3, ya que este último presenta una mejor calidad de predicción con un mayor tiempo de procesamiento. Los modelos de Autoencoder 1 y 2, tienen desempeños similares, ya que poseen la misma profundidad, el factor que implica un mejor rendimiento en el Autoencoder 1 es el tamaño del filtro convolucional de su topología (2×2), ya que este permite extraer patrones locales de menor escala, haciendo que la compresión contenga información sin muchas redundancias, como consecuencia de ello, en la decodificación se puede reconstruir la muestra de 1 trama con la información más selecta del conjunto de datos de forma más óptima.

Finalmente, en los modelos DnCNN, se puede notar que ambos tienen la misma latencia y

esto se debe a que comparten la misma estructura por capas y configuración de hiper-parámetros, sin embargo, el modelo Denoising 2 tiene una mejor calidad de estimación y esto se debe en el aprendizaje del ruido η , ya que es una información siempre presente en el canal de comunicaciones (ya sea a menor o mayor escala), comparado con la forma de aprender del otro estimador de canal. El desempeño de la red Denoising 2, tiene que ver en gran medida con la operación resta que se hace en la salida de esta, ya que maneja los escenarios de menor SNR dándole mayor importancia a la estimación comparada con la muestra en la entrada, mientras que si el SNR es alto, la muestra de 1 ms tiene un mayor peso si se compara al nivel de ruido estimado.

8.3 Selección de estimadores de canal ANN

8

En los sistemas de comunicaciones la estimación de canal es un proceso fundamental ya que permite conocer las condiciones del medio inalámbrico y de esta forma, optimizar la comunicación entre el transmisor y el receptor. Por ello, en esta segunda etapa, se evalúa el rendimiento de los estimadores de canal ANN en un escenario dentro del sistema propuesto para este trabajo, de allí se extraen las métricas de MSE y de tiempo de estimación, sin embargo, como se desea evaluar su desempeño en un entorno de red 5G, se hace el envío de bits aleatorios en una duración de N tramas, esta adecuación da pie a obtener métricas del desempeño del sistema como: EVM para medir la calidad de la modulación y BER para evaluar la calidad y confiabilidad de la transmisión de datos.

El procedimiento consiste en enviar tramas con bits aleatorios, los cuales pasan por los bloques del sistema de comunicaciones, en la estimación de canal se usan cada uno de los algoritmos ANN diseñados: CNN estándar, autoencoder y red denoising. Para la adquisición de las métricas: MSE, tiempo de estimación, EVM y BER, se fijan algunos parámetros del canal y se hace un barrido de niveles de ruido AWGN, de ahí se puede medir el desempeño tanto de la estimación como del sistema de comunicaciones propuesto. Como parte final de esta etapa, se hace una comparación entre los modelos neuronales del mismo tipo y se hace un proceso de selección de acuerdo con los resultados obtenidos, evaluando su desempeño y adaptabilidad ante valores de SNR diferentes.

8.3.1 Configuración del sistema de comunicaciones y simulación

En la configuración del sistema, se mantienen los valores de los procesos asociados con el transmisor (sección 7.2.3). Esta prueba para obtener las métricas de desempeño consiste en fijar las condiciones de un canal TDL variando únicamente los niveles de ruido, por lo que el perfil de retardo, la dispersión del retardo y la velocidad del usuario se mantienen constantes a lo largo de esta prueba. Para evaluar el rendimiento del sistema de comunicaciones se envían 5 tramas con bits aleatorios y se hace un barrido del SNR para cada uno de los modelos, debido a que en el entrenamiento de los estimadores se usó un rango entre 0 dB y 25 dB se opta por mantener este intervalo. Los parámetros usados para esta prueba son:

Parámetro	Valor
Frecuencia portadora	5 GHz
Bloques de recursos	52
Número de subportadoras	624
Modelo del canal	TDL-E (NLOS)
SNR	0 dB hasta 25 dB
Velocidad Usuario	15 km/h
Frecuencia máxima de Doppler	69.49 Hz
Dispersión del retardo	300 ns
Canal físico	PDSCH
Mapeo símbolos PDSCH	A
Espaciado de subportadoras	15 kHz
Símbolos OFDM	14
Duración del símbolo	66 μ s
Esquema de modulación	16QAM
Tramas enviadas	5
Tipo de sistema	SISO

Cuadro 8.2: Parámetros para la simulación con condiciones fijas en el canal de comunicaciones

Tras completar la simulación con los parámetros del cuadro (8.2), se grafican los resultados de las métricas respecto a los valores de SNR y se agrupan en sus respectivos tipos: CNN estándar, autoencoder o red denoising. A manera de tener una comparativa entre las calidades de predicción

de cada modelo, se toman muestras de 1 ms de la estimación hecha con la interpolación lineal (entrada de los modelos neuronales) y la estimación ideal (salida objetivo de cada ANN), la muestra se toma con la primera iteración de la simulación, es decir, con $SNR = 0$ dB. Las referencias para establecer la comparación son:

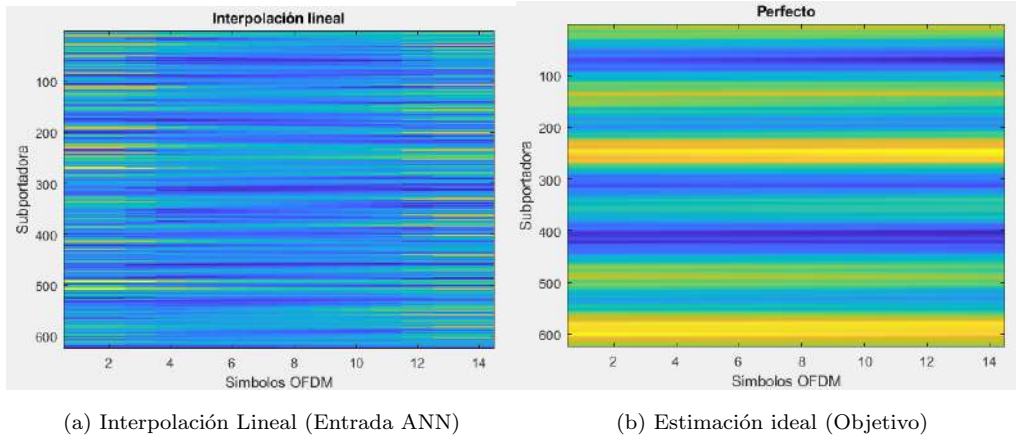


Figura 8.2: Muestras de referencia para la comparativa de la calidad de predicción con un SNR de 0 dB

8

Los resultados se van a presentar con dos tipos de figuras: la comparación de la muestra estimada respecto a las referencias de la imagen (8.2) para cada uno de los modelos con una relación señal a ruido de 0 dB, esta situación sirve para ilustrar la capacidad de adaptabilidad y la calidad de predicción en un escenario en el que la potencia de la señal y el ruido es la misma, lo cual es más cercano a un entorno real de sistemas de comunicaciones; el otro tipo de figura es el que contiene las gráficas de cada métrica respecto al barrido en el intervalo SNR, para esta último adjuntan las leyendas de cada curva, para etiquetar la información resultante de la simulación.

8.3.2 Modelos de redes neuronales convolucionales

Para la primera figura, se parte de que es necesario disponer de la misma muestra de la rejilla de recursos estimada para todos los algoritmos de CNN estándar desarrollados, es por ello que se opta por tomar la primera subtrama de la iteración SNR de 0 dB. La muestra seleccionada permite visualizar de una manera más apropiada el proceso hecho por cada uno de los modelos y los efectos que generan las ligeras variaciones en los hiper-parámetros configurados en las capas: tamaño de filtro, cantidad de filtros, relleno de ceros, capas de agrupación o desplazamiento del filtro. Los resultados se muestran en la figura (8.3):

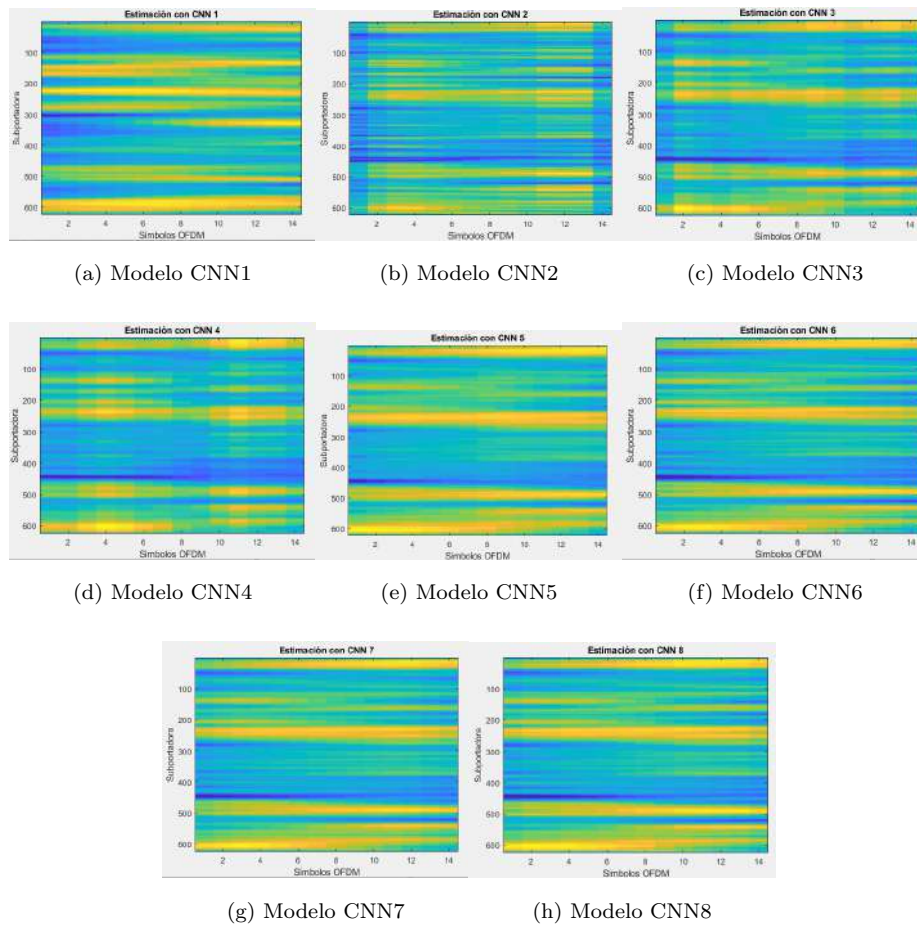


Figura 8.3: Muestras de la estimación hecha por cada modelo CNN estándar con 0 dB

Como se puede observar en la imagen (8.3) la calidad de la estimación hecha por cada uno de los modelos es apropiada teniendo en cuenta que se están trabajando con un escenario en el que hay mayor presencia de ruido y por ende, más redundancias que deben ser eliminadas por el estimador ANN.

En las estimaciones hechas con los modelos CNN1, CNN5, CNN6, CNN7 y CNN8 se tienen resultados parecidos, esto se puede corroborar con la comparación hecha en la figura (8.1), donde justamente estos se ubicaban en una región en la que el MSE promedio era inferior comparado con el resto de modelos, también, esto refleja la versatilidad que tienen las CNN, ya que pese a variar los hiper-parámetros, siguen presentando resultados que se aproximan a la estimación ideal.

Para los algoritmos de CNN2, CNN3 y CNN4 se puede evidenciar un fenómeno de variación

de la intensidad de los coeficientes del canal a lo largo de eje x , como consecuencia del uso de las capas *max pooling* ya que estas cumplen con el objetivo de extraer únicamente las características dominantes y descartar la información detallada de la muestra de entrada para optimizar el costo computacional, esto ocasiona que en la predicción se pierdan los detalles finos y suavizados, dando indicios de discontinuidades entre símbolos.

Para las gráficas de las métricas de los estimadores y del sistema de comunicaciones propuesto, es necesario tener en consideración las etiquetas asociadas a las curvas de cada modelo:

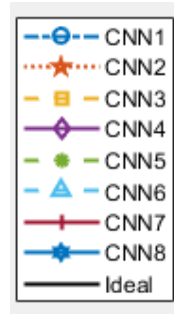


Figura 8.4: Etiquetas de las curvas de cada modelo CNN estándar

Las métricas en función del valor de SNR son:

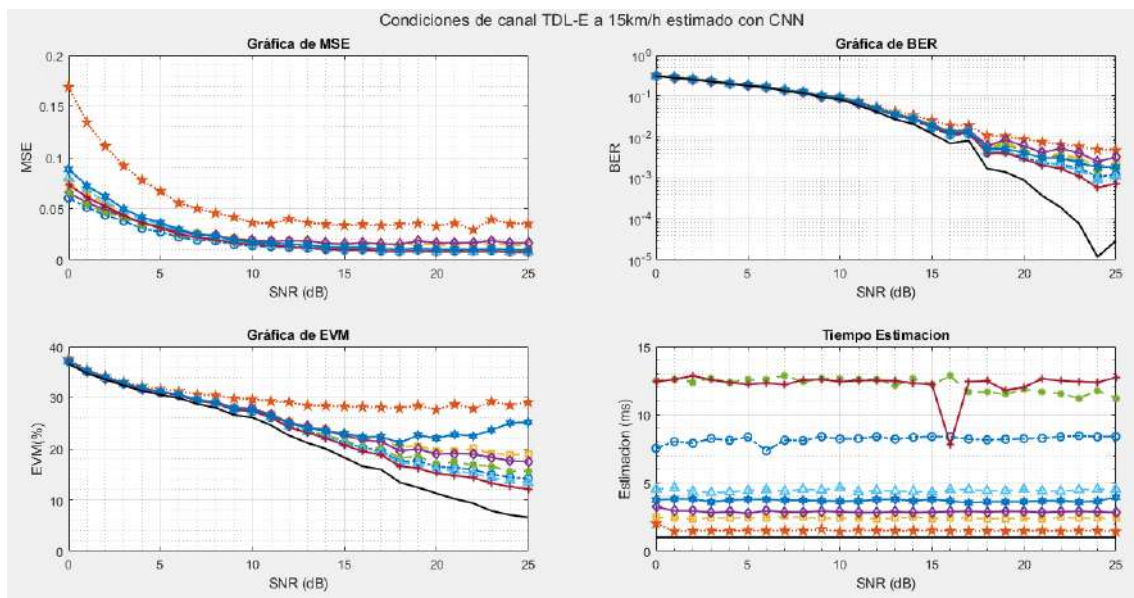


Figura 8.5: Métricas de desempeño de las CNN estándar

De la gráfica del MSE se puede detallar que a medida en la que el valor del SNR aumenta, decrece el error de predicción, esto se debe en gran medida a que mejora la calidad de la muestra de entrada al modelo CNN, por lo cual, ya no es necesario que este se encargue de eliminar tantas redundancias como si ocurre con los primeros valores de SNR. El modelo CNN2 presenta un valor alto de MSE y esto refuerza más los resultados del rendimiento presentados en la primera etapa. Del resto de modelos se puede notar que tienen valores MSE próximos entre si, lo cual es un indicio de que presentan una buena respuesta con escenarios en los cuales existe una variación constante de los niveles de ruido.

En la gráfica del BER se aprecia un comportamiento similar al descrito con la métrica de MSE, y es que ante el aumento del SNR se reduce considerablemente la cantidad de bits recibidos incorrectamente. Sin embargo, si se hace una comparación con la estimación ideal al rededor de los 18 dB se puede notar que en este punto mantienen la tendencia mientras que en la estimación ideal, se produce una tasa de decaimiento superior, esta parte se puede solucionar enviando más bits o empleando un esquema de modulación con menos símbolos, como QPSK. Un aspecto que se puede destacar es que la red CNN2 sigue teniendo un desempeño bajo.

La métrica del EVM permite medir la calidad del esquema de modulación usado en la simulación. Para los modelos CNN2 y CNN8 se observa que tienen un aumento del EVM cuando se completa la simulación, el punto en el que ocurre este cambio fue en 18 dB y en los 25 dB llegan a tener valores de 29% y 25% respectivamente, denotando un desajuste en el rendimiento cuando se usa la modulación 16QAM. Por otro lado, en los estimadores restantes, se contempla que mantienen la tendencia en el decaimiento del EVM, siendo los modelos CNN1, CNN4, CNN5, CNN6 y CNN7 los que mejor rendimiento tienen, indicando que se ajustan de una manera más eficiente con la modulación de la simulación.

Por último, el tiempo de estimación proporciona información fundamental del rendimiento general de los modelos dentro del sistema de comunicaciones, de allí se puede destacar que los modelos CNN5 y CNN7 pese a tener mejor rendimiento en las métricas previas, son los que mayor tiempo toman para generar una salida, haciendo que tarden hasta 12 veces más que la referencia ideal, a su vez, el CNN1 también tiene un tiempo de estimación elevado comparado con el ideal. En la gráfica se demuestra que el tiempo de estimación de cada modelo es aproximadamente constante sin importar las condiciones de ruido en el canal, significando la invarianza ante la presencia de

redundancias en mayor o menor escala en la muestra de entrada.

Finalmente de acuerdo con las métricas adquiridas, se opta por seleccionar los **modelos CNN4 y CNN6**, debido a que mantienen un rendimiento constante a lo largo de la simulación, presentando un balance entre la capacidad de generalización y el tiempo en el que tardan en generar una salida aproximada al canal ideal, factores que dotan al sistema de una mayor capacidad de respuesta ante entornos dinámicos.

8.3.3 Modelos de redes Autoencoder

El autoencoder convolucional es un modelo que ofrece una robustez superior si se compara con las CNN estándar, ya que este cuenta con más capas y una mayor profundidad, haciendo que tenga en las primeras muestras y bajo las condiciones de simulación, una mejor calidad de predicción para el valor de 0 dB, tal como se observa en la figura (8.8):

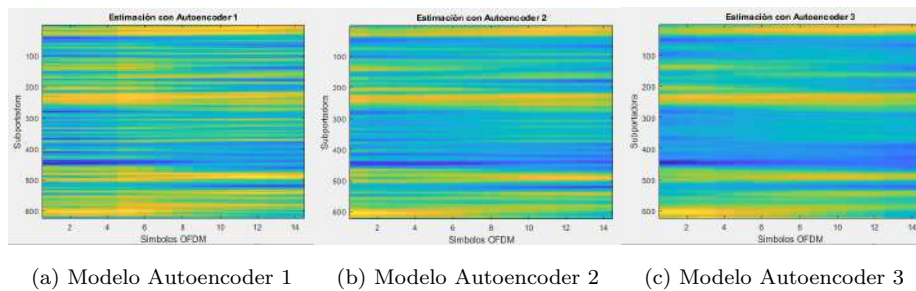


Figura 8.6: Muestras de la estimación hecha por cada modelo Autoencoder con 0 dB

Algo que se puede destacar de los tres modelos de Autoencoder es el hecho de que pese a contar todos con capas *max pooling*, no cuentan discontinuidades entre los símbolos como si ocurría con las CNN2, CNN3 y CNN4. La razón principal por la cual se mitiga este fenómeno en la predicción, es la adición de las capas de decodificación, las cuales son capaces de generar la rejilla de recursos únicamente con las características dominantes de la muestra de entrada, lo cual posibilita la inferencia de los detalles suavizados independientemente de la compresión hecha en las capas de codificación.

Evaluando los resultados de la figura (8.8), se aprecia una alta similitud entre las estimaciones hechas, no se distinguen los efectos debido al tamaño del filtro o el número de filtros de las capas

de codificación y decodificación. Con base en ello, se puede remarcar que la robustez y complejidad de los modelos dificulta la capacidad de distinguir la calidad de la estimación de canal ante condiciones fijas, por ello se opta por evaluar su desempeño con la variación de los niveles de ruido.

Para las métricas de desempeño de los estimadores con autoencoder de cada modelo, se tienen en cuenta las siguiente etiquetas:



Figura 8.7: Etiquetas de las curvas de cada modelo Autoencoder

Las métricas en función del valor de SNR son:

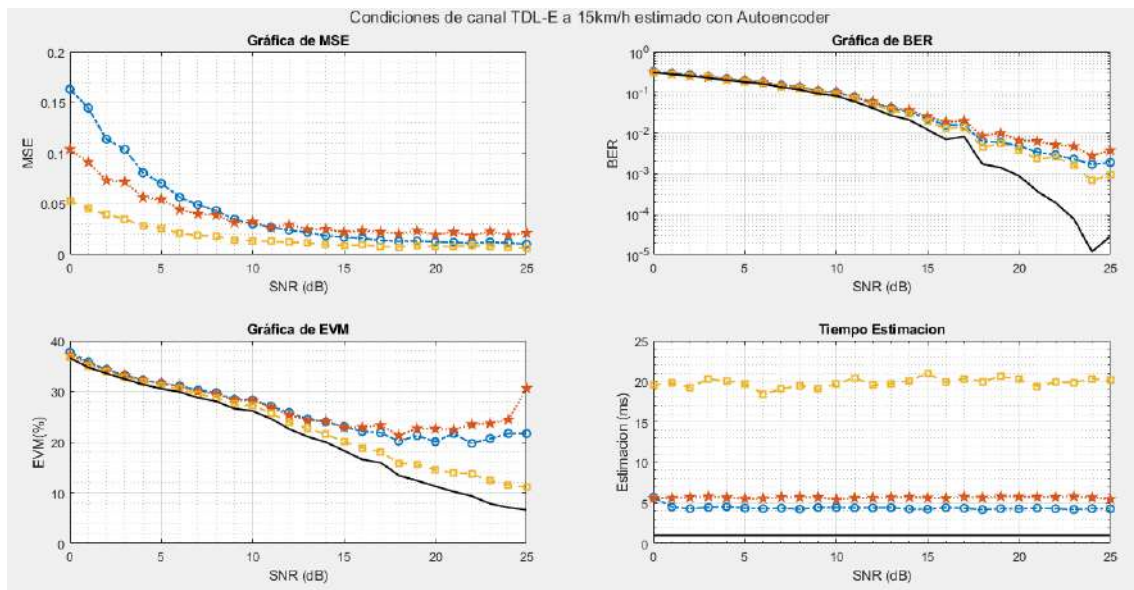


Figura 8.8: Métricas de desempeño de los Autoencoder

En el caso del MSE respecto al SNR se puede observar la tendencia decreciente que hay a medida en la que la potencia de la señal es superior a la del ruido introducido en el canal de comunicaciones. La red Autoencoder 3 es la que mantiene una calidad de predicción superior comparada con los otros dos modelos a lo largo del intervalo de simulación, por otro lado, el Autoencoder 2 presenta un mejor rendimiento que el Autoencoder 1 para valores de SNR inferiores a 11 dB, sin embargo, en el resto de valores, decae su rendimiento, lo cual puede ser consecuencia de una falta

de muestras con estas características durante su entrenamiento.

En el BER se observa que los tres modelos comparten similitudes, que hace que se superpongan en el inicio de la simulación y que así mismo, manteniendo la tendencia de la disminución de bits incorrectos con los aumentos en la relación señal a ruido. Sin embargo, al rededor de los 18 dB ocurre un cambio en la tasa de bajada del BER, reflejándose como un distanciamiento de la referencia de estimación ideal. El modelo que se ve más afectado por este comportamiento es el Autoencoder 2, seguido del 1 y el 3.

El esquema de modulación 16QAM tiene un mejor desempeño cuando el canal es estimado por el Autoencoder 3, debido a que la métrica del EVM mantiene su tendencia de reducción con este modelo, caso contrario ocurre con los otros dos algoritmos, ya que desde los 18 dB empiezan a oscilar los valores, llegando incluso a subir en determinados SNR. Al tener en cuenta las especificaciones de estos modelos, se encuentra que tienen menos capas comparadas con el modelo 3, lo cual repercute en la sensibilidad que tienen de detectar ligeras variaciones en el canal y por ende, incurrir en pérdidas de información sobre los coeficientes del canal.

Finalmente, en el tiempo de estimación se presenta una vez más la estrecha relación que hay entre la calidad del estimador de canales y el tiempo en el que tarda en hacer este proceso, ya que el modelo 3 a lo largo de las simulaciones llega a presentar mejores rendimientos, sin embargo es aproximadamente $\times 4.5$ y $\times 3.5$ veces superior respecto a los modelos 1 y 2 respectivamente, haciendo que el proceso de la estimación de canal tenga una duración superior a la de 1 trama enviada, produciendo ineficiencias en el sistema de comunicaciones y retardos en la entrega de información. Los tiempos normalizados entre los autoencoders 1 y 2 son ligeramente diferentes, siendo el primer modelo un 25% más rápido.

Con base en las métricas obtenidas durante la simulación de cada uno de los modelos, se opta por elegir el **Autoencoder 1**, ya que presenta un balance entre la calidad de la estimación dentro del sistema de comunicaciones y el tiempo que tarda en generar una salida con los datos provenientes de la interpolación lineal de las señales de referencia.

8.3.4 Modelos de redes denoising

Para la selección de los últimos dos modelos, se recuerda que ambas DnCNN comparten las mismas características y configuraciones en las capas, el único cambio hecho es el tipo de estimación a realizar, ya que una genera la rejilla de recursos mientras que la otra extrae el ruido y realiza una posterior resta con el resultado de la interpolación lineal. Los resultados para una muestra de 1 ms son:

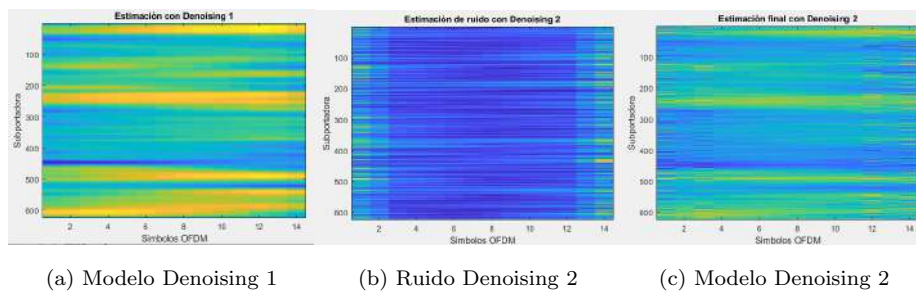


Figura 8.9: Muestras de la estimación hecha por cada modelo Denoising con 0 dB

En la figura (8.11) se tienen las salidas de cada uno de los modelos. El modelo Denoising 1 tiene una mayor cercanía con la muestra tomada de la estimación ideal, mientras que el Denoising 2 tiene una distribución de coeficientes similar pero con cierta atenuación, esto se debe a que la información del ruido tiene un mayor peso al hacer la resta comparado con los datos provenientes de la interpolación lineal. Así que en un principio, se puede establecer que con un escenario en el que el ruido y las señales tienen la misma potencia, existe una ligera mejora cuando la red es entrenada para hallar los patrones locales de la rejilla de recursos por sobre los patrones en el ruido (redundancias).

Para la simulación de las variaciones del SNR se tienen en cuenta las siguientes etiquetas:



Figura 8.10: Etiquetas de las curvas de cada modelo Denoising

Los resultados obtenidos son:

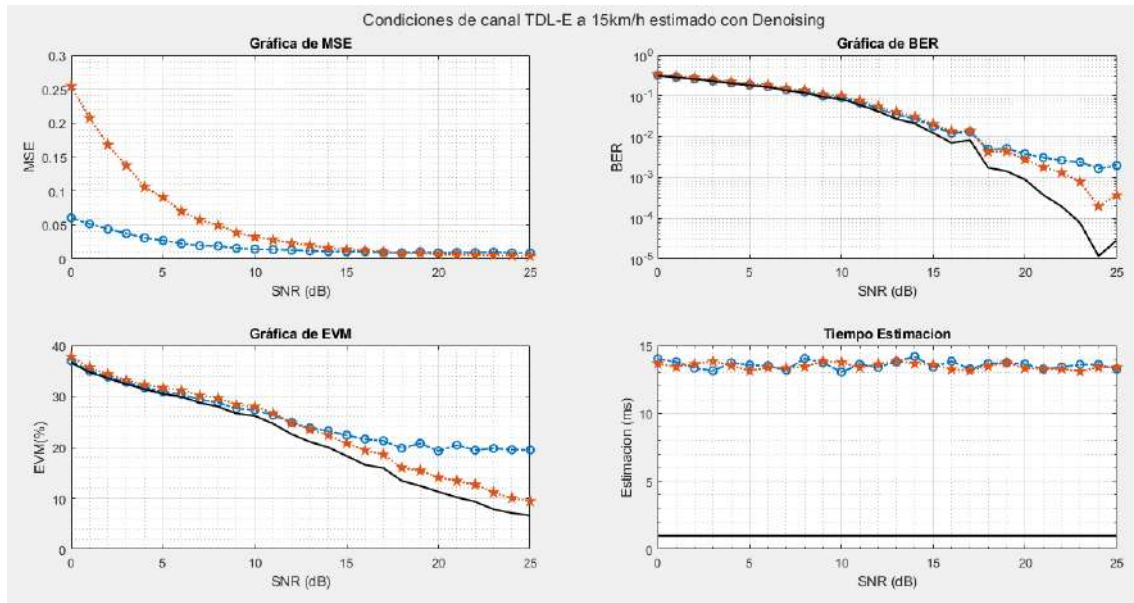


Figura 8.11: Métricas de desempeño de las DnCNN

En el MSE hay una notable diferencia entre los dos modelos, ya que Denoising 1 tiene un mejor desempeño para valores inferiores a 15 dB, mientras que para los valores superiores, la red Denoising 2, presenta una mejor calidad de predicción, la explicación de este suceso se debe en gran medida a que al haber menores niveles de ruido, tiene un mayor peso la muestra de entrada, la cual no contiene tantas redundancias y por ende, se hace una estimación de ruido más precisa, eliminando remanentes que pueden quedar y que en otros modelos no se mitigaban completamente.

En las gráficas del BER y del EVM se puede notar que ambos modelos llegan incluso a solaparse con la estimación ideal en el intervalo comprendido por los valores de 0 dB a 10 dB, esto denota la alta capacidad que tienen para asemejarse a un escenario ideal con condiciones más adversas, como lo son aquellas en la que el ruido es más elevado. Sin embargo, como ha ocurrido con el resto de modelos, existe un punto en el cual, se alejan de la referencia, este se ubica en los 18 dB, el cambio más brusco ocurre con el EVM del modelo 1, el cual llega a bajar lentamente comparado con el modelo 2, el cual mantiene su tendencia.

Debido a que ambos modelos comparten las mismas configuraciones de hiper-parámetros, era de esperarse que compartieran el mismo tiempo de estimación, por lo cual, la capacidad de generar la salida de un modelo es completamente independiente de la calidad de los datos que se usaron en el entrenamiento.

Dado que ambos modelos comparten el tiempo de estimación y su rendimiento es similar en las condiciones con mayor ruido, se opta por seleccionar ambas redes: **Denoising 1 y Denoising 2**, con el objetivo de evaluar su capacidad de generalización cuando se cambian las condiciones de la simulación.

8.4 Comparación entre modelos seleccionados con diferentes escenarios

En esta sección se presentan los resultados obtenidos en una comparación directa entre cada uno de los modelos a partir de la selección realizada en la etapa anterior para los modelos CNN, Autoencoder y Denoising. Esta comparación permite observar el comportamiento del sistema de comunicaciones realizado ante los diferentes modelos tradicionales y neuronales diseñados para la estimación de canal, de igual manera permite observar el comportamiento de dichos modelos ante cambios propios de un sistema de comunicaciones móviles como lo son modulación, relaciones señal a ruido, velocidad de usuario y perfil de canal.

En este análisis solo se muestran las gráficas explicadas para 2 perfiles de canal, uno sin línea de vista y otro con línea de vista, sin embargo, las simulaciones de los demás modelos de canal se encuentran disponibles en el anexo A.

8.4.1 Parámetros de la simulación

Un modelo de comunicaciones tan grande y variable presenta muchos parámetros de salida y métricas de valor que son útiles con el fin de medir el desempeño general, por lo que en este apartado se analizan las métricas de EVM, MSE, BER y tiempo de estimación.

Por lo tanto todas las siguientes figuras se componen de 4 imágenes en donde se presentan estas métricas para cada modelo de estimación seleccionado. Las gráficas se hacen de acuerdo con: 3 tipos de modulación, una baja (QPSK), una mediana (64QAM) y una grande (256QAM), tres velocidades, una velocidad de usuario en un movimiento lento (5 km/h), un usuario en un movimiento moderado como lo podría ser un automóvil en una ciudad (30 km/h) y un escenario a gran velocidad emulando un desplazamiento en carretera o tren (100 km/h).

Con el fin de facilitar la lectura de las gráficas se asignó una secuencia de colores y simbología para las curvas asociadas a cada modelo, con esto es posible identificar un modelo específico de forma ágil. Las nomenclaturas empleadas se ilustran en la figura 8.12, se recuerda al lector que en la sección anterior se dieron las pautas de selección de los modelos, por lo que el modelo CNN corresponde a la red neuronal CNN modelo 4 (cuadro 7.5) y el modelo CNN₂ corresponde a la red neuronal CNN modelo 6 (cuadro 7.7), Autoencoder corresponde al algoritmo 1 (cuadro 7.10), Denoising a la topología descrita en (cuadro 7.13) y Denoising₂ a la arquitectura encargada de estimar el ruido.

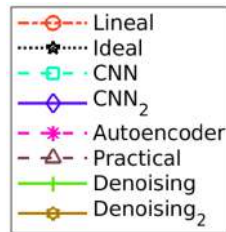


Figura 8.12: Nomenclaturas simulaciones asociadas Canales TDL

8

En la tabla 8.3 se aprecian los parámetros utilizados para la obtención de los resultados de las figuras siguientes.

Parámetro	Valor
Frecuencia de la portadora	5 GHz
Bloques de recursos	52
Número de subportadoras	624
Modelos del canal	TDL-B (NLoS) y TDL-E (LoS)
SNR	0 dB hasta 25 dB
Velocidad Usuario	[5, 30, 100] km/h
Frecuencias máximas de Doppler	[23.16, 138.98, 463.28] Hz
Dispersión del retardo	300 ns
Canal físico	PDSCH
Mapeo símbolos PDSCH	A
Espaciado de subportadoras	15 kHz
Símbolos OFDM	14
Duración de símbolos	66 μ s
Esquemas de modulación	[QPSK, 64QAM, 256QAM]
Tramas enviadas	7
Tipo de sistema	SISO

Cuadro 8.3: Parámetros utilizados durante las simulaciones

Con el fin de caracterizar el sistema propuesto para cualquier modelo de canal TDL, se hace una comparación entre las modulaciones para un entorno de canal NLoS, allí se hacen las variaciones de velocidad y se observa los efectos que ambos parámetros tienen sobre la estimación del canal elegido. Posteriormente, se repite la prueba con otro tipo de canal, concretamente un LoS.

8.4.2 Canal con perfil de retardo TDL-B (NLoS)

En esta sección se presentan los resultados correspondientes a un perfil de canal TDL-B el cual fue seleccionado entre los modelos usados para caracterizar escenarios sin línea de vista descritos en el capítulo 3.

Modulación QPSK

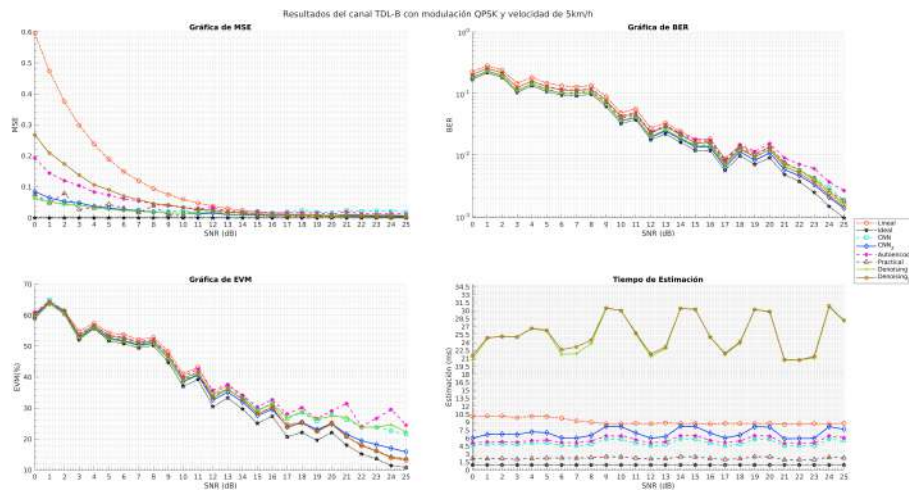


Figura 8.13: Velocidad Usuario 5 km/h

En la figura 8.13 se aprecia el primer escenario con una modulación baja la cual presenta el menor valor de BER comparado con los resultados para otras velocidades, lo que representa un valor esperado para un escenario con velocidad baja en donde todos los modelos tienen la misma tendencia y prácticamente el mismo valor, siendo el peor modelo el Autoencoder comparado directamente con los otros modelos desarrollados, sin embargo, no es un mal resultado comparándolo con el ideal. El mejor modelo es el obtenido por la CNN 6 y muy cerca a este se presenta la red Denoising 2, estas 2 redes presentan mejores resultados en términos de BER que modelos como el lineal y práctico, comparando estos resultados con el aumento de velocidad apreciado en las gráficas 8.14 y 8.15 se evidencia el decaimiento de la tasa de error binaria, en donde para el escenario de 100 km/h los modelos se alejan del valor ideal a valores altos de SNR.

Para el EVM de igual manera para el escenario de 5 km/h se presenta un comportamiento casi de que ideal hasta valores de 17 dB en donde a partir de este valor no se presenta una mejoría en esta métrica para los modelos de CNN 4, Autoencoder y Denoising, en esta métrica el mejor resultado lo obtuvo la red Denoising 2 con el mismo valor de una estimación practica, seguido de la CNN 6, esta tendencia se mantiene para una velocidad mayor, sin embargo a una velocidad de 100 km/h para valores mayores de 14 dB todos los modelos diseñados se alejan del ideal siendo el que mejor desempeño presenta el modelo de la red Denoising, por lo que claramente se aprecia el impacto de la velocidad en la estimación con los modelos desarrollados.

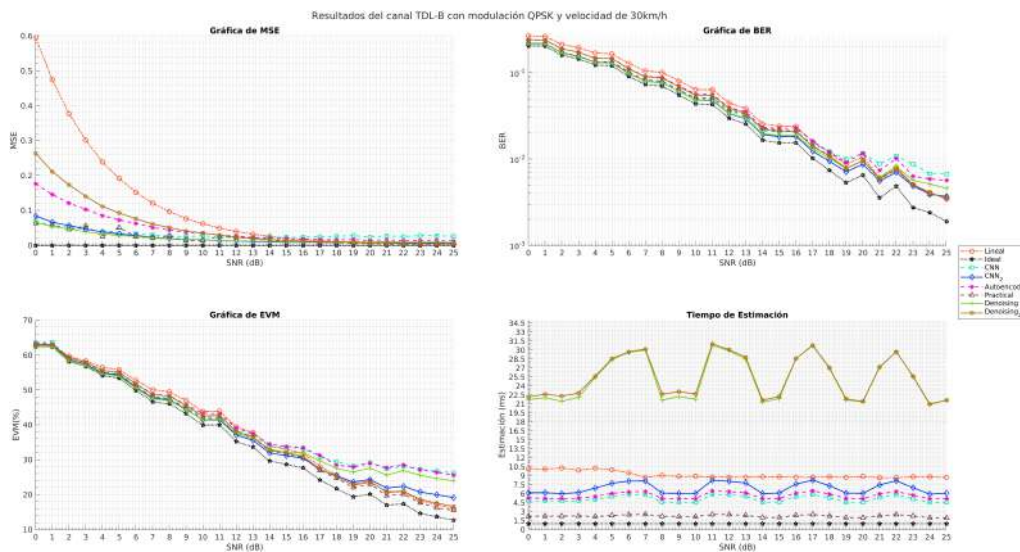


Figura 8.14: Velocidad Usuario 30 km/h

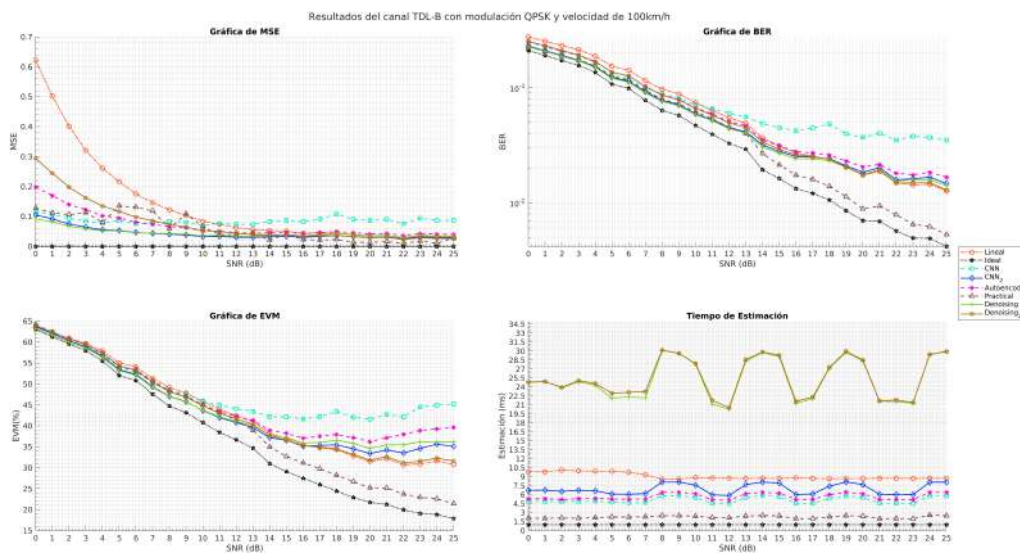


Figura 8.15: Velocidad Usuario : 100 km/h

Un factor notable es que el tiempo de estimación se ve afectado por la calidad de la predicción,

en donde las redes denoising presentan un retardo cuya duración esta entre 2 y 3 tramas, por lo que ciertamente un modelo mas robusto y con mejores resultados en otras métricas de evaluación tendrá un mayor tiempo en procesar la muestra de entrada. El modelo del Autoencoder presenta un retardo de media trama (5 ranuras) aproximadamente, por su parte el modelo práctico es el que obtiene mejores resultados con tan solo un retardo de 2 *slots*, esto es coherente con el planteamiento que este modelo representa un sistema dedicado. En relación al impacto de la velocidad, el tiempo de estimación cambia ligeramente sin embargo no representa una variación significativa, haciendo que el tiempo se independiente del efecto Doppler en el canal.

La gráfica de MSE no presento cambios ante la velocidad lo que quiere decir que la calidad de la estimación no se vio afectada. La tendencia marca que los algoritmos: CNN 6, Denoising y estimación práctica presentan los mejores resultados desde valores muy bajos de SNR

Modulación 64 QAM

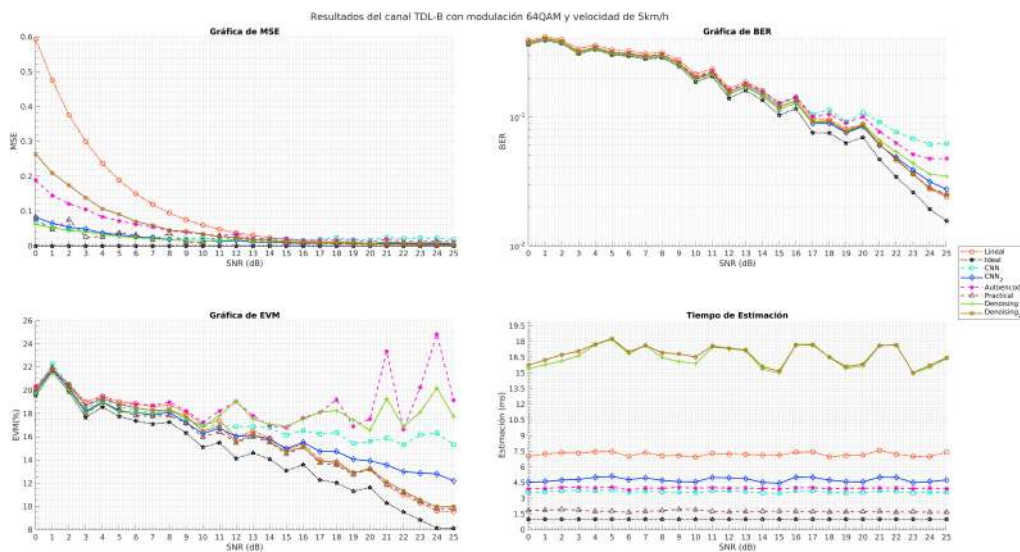


Figura 8.16: Velocidad Usuario 5 km/h

El impacto de la modulación en el BER, EVM y tiempo de estimación son considerables, mientras que el MSE para los modelos diseñados no se ve afectado. En primera instancia se podría concluir que el cambio de modulación no afecta la calidad de la estimación realizada, esto no sucede con el práctico el cual se ve ligeramente afectado llevándolo a ser el segundo peor modelo para una velocidad de 5 km/h.

En términos de BER se presenta un valor menor en comparación a la modulación QPSK, sin embargo esto se asocia a la cantidad de datos enviados por 7 tramas simuladas, por lo que si se

deseada obtener un valor mayor de BER sería necesario transmitir mas datos, esto no afecta el comportamiento obtenido por lo que se puede apreciar la misma tendencia a la modulación QPSK en donde el modelo Denoising 2 y CNN 6 obtienen los mejores resultados. En cuanto a los cambios de velocidad. para 64QAM se evidencia el mismo comportamiento descrito para QPSK, para una velocidad media las métricas de BER se mantienen, como se aprecian en los resultados de la figura 8.18 los modelos se separan del comportamiento ideal siendo los mas cercanos la CNN 6 y la red Denoising 2.

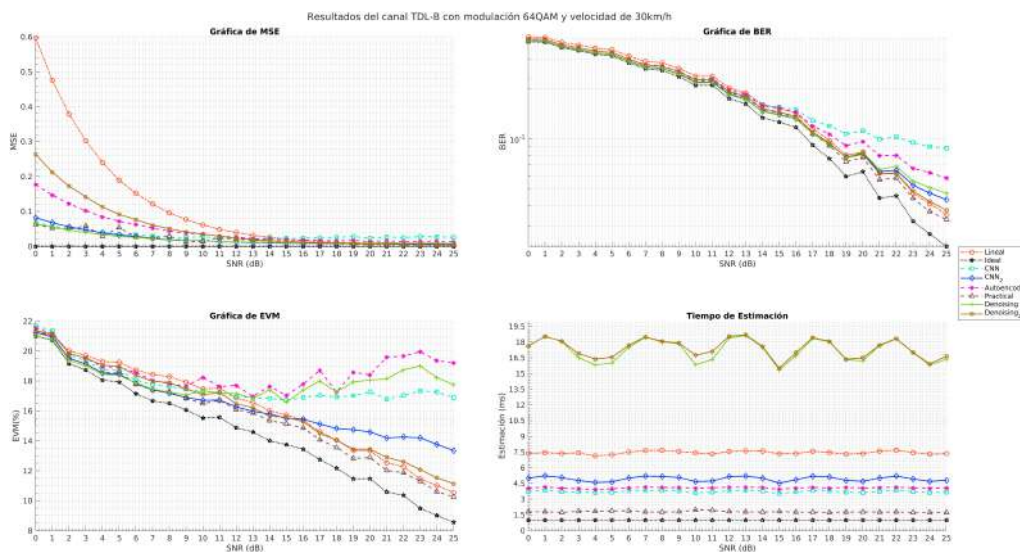


Figura 8.17: Velocidad Usuario 30 km/h

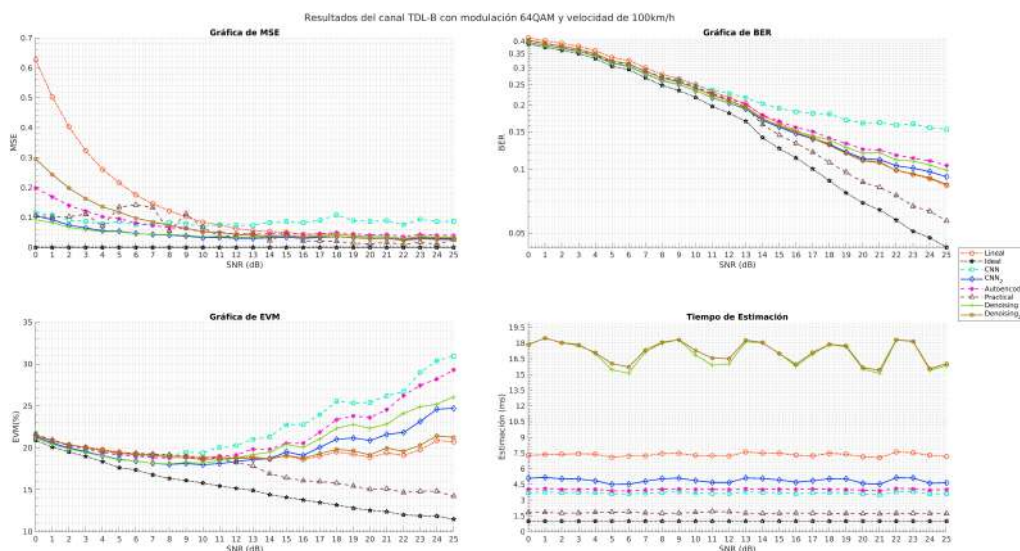


Figura 8.18: Velocidad Usuario : 100 km/h

El valor de EVM también se ve claramente afectado, principalmente para las redes Autoencoder y Denoising que presentan mayor dispersión en los símbolos estimados, y en donde a medida que

aumenta la velocidad y a valores altos de SNR los demás modelos se alejan de igual manera del comportamiento ideal.

En términos de tiempo de estimación se presenta una mejora considerable con respecto a la modulación QPSK, lo cual difiere con la hipótesis de que para una modulación mas robusta se tardaría mas en estimarse el canal, estos resultados se deben a que al empaquetarse una mayor cantidad de bits por trama se requiere un menor tiempo de procesamiento para realizar el mismo trabajo que con una QPSK, este cambio se ve principalmente en los modelos de las redes Denoising en donde se pasa a tener tiempo cercano a 2 tramas, a su vez. los demás modelos se mantienen consistentes con los resultados obtenidos.

Modulación 256 QAM

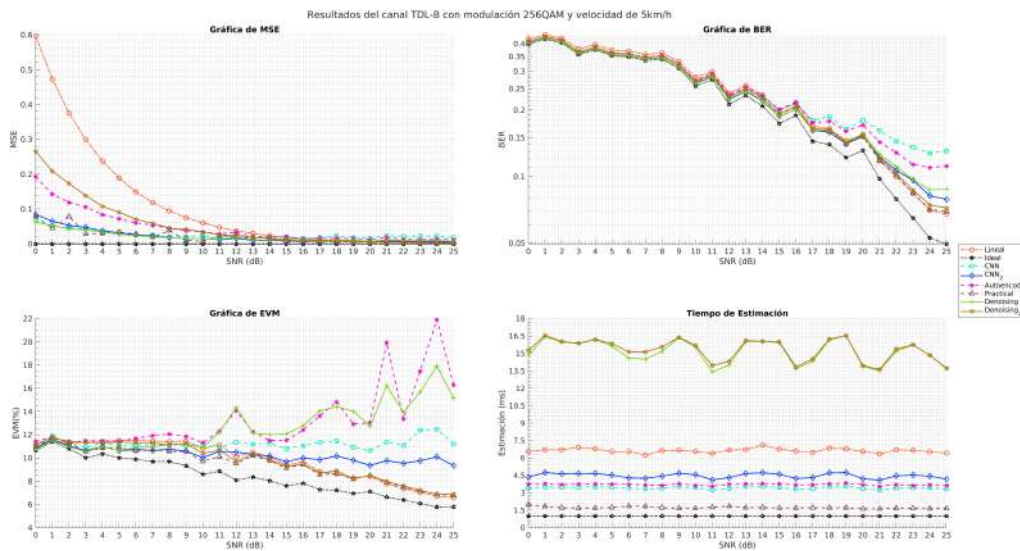


Figura 8.19: Velocidad Usuario 5 km/h

Los resultados obtenidos para esta modulación son bastante consistentes puesto que a nivel de MSE los valores no cambian por que definitivamente se aprecia que la modulación no influye en la calidad de la estimación realizada, así como los valores de BER, en donde se sigue el mismo comportamiento de las modulaciones anteriores, en donde el verdadero impacto lo tiene la velocidad del usuario. El tiempo de estimación mejora con respecto al escenario con modulación 64QAM.

En términos de EVM se presentan la misma tendencia de mayor dispersión de símbolos a medida que aumenta la velocidad, no obstante, se obtienen valores mas bajos que con las modulaciones QPSK y 64QAM esto debido a que una modulación 256QAM es mucho mas compleja y los puntos

de la constelación están más juntos, haciendo que la modulación de mayor orden tenga más sensibilidad ante interferencias, por lo que esto se cumple en el sistema de comunicaciones modelado, nuevamente el mejor modelo para EVM es la red Denoising seguido de la red CNN6

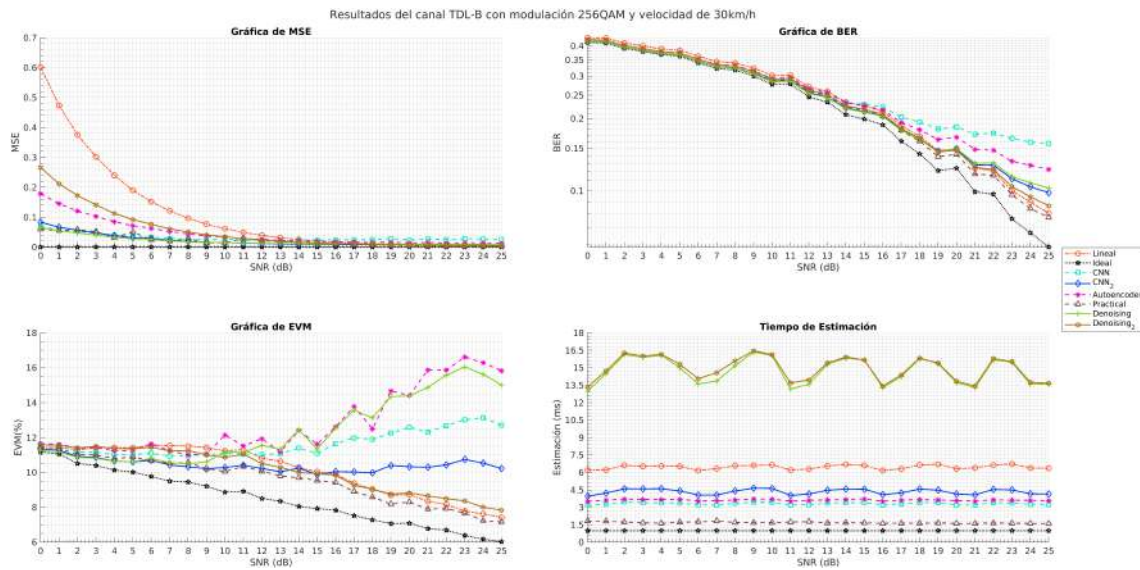


Figura 8.20: Velocidad Usuario 30 km/h

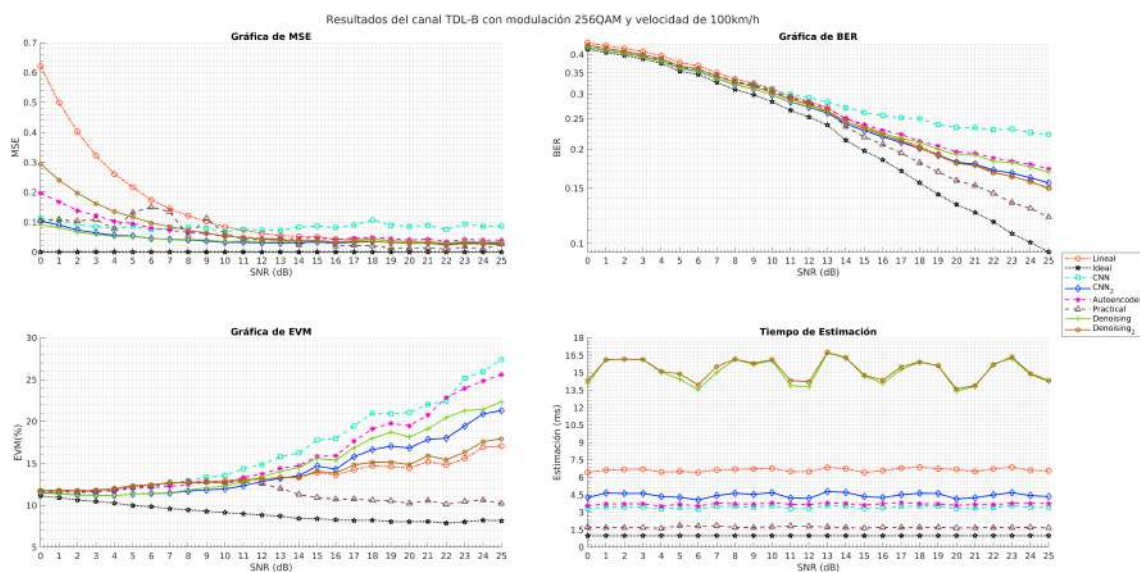


Figura 8.21: Velocidad Usuario : 100 km/h

8.4.3 Canal con perfil de retardo TDL-E (LoS)

En esta sección se presentan los resultados correspondientes a un perfil de canal TDL-E el cual fue seleccionado entre los modelos usados para caracterizar escenarios con línea de vista sustentados en el capítulo 3.

Modulación QPSK

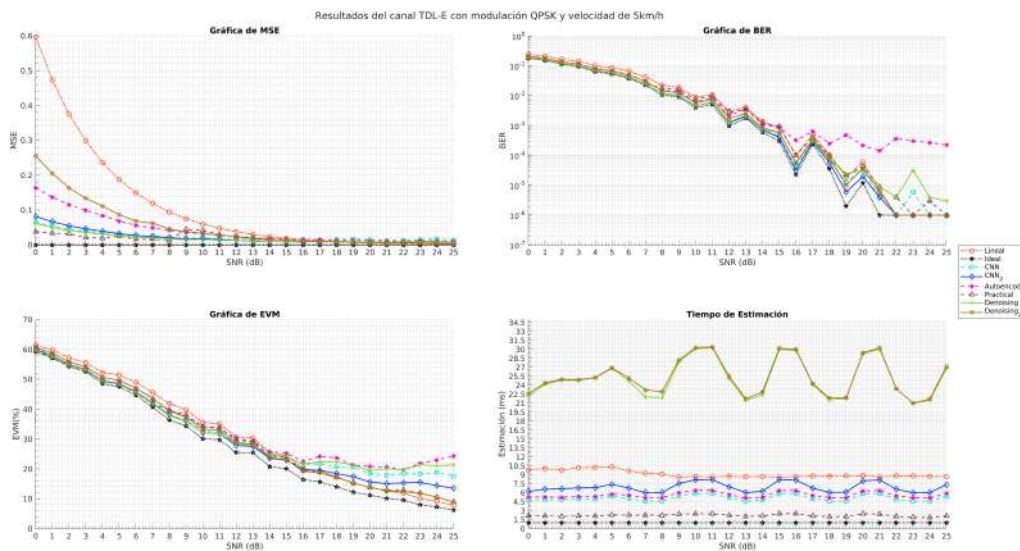


Figura 8.22: Velocidad Usuario 5 km/h

La primer observación que se puede realizar para este perfil de canal es que se tienen mejores resultados para las métricas de BER y EVM, en cuanto al MSE se evidencia que no varía en gran medida con respecto a los resultados anteriores para el perfil de canal TDL-B, por lo que se prueba que la calidad de la estimación no se ve afectada entre modelos de canal LoS o NLoS.

En la figura 8.22 se aprecia que el valor de BER continúa constante a partir de 21 dB lo que significa que en este punto se logró el envío total de los datos (7 tramas) sin ningún error, esto comparado antes las mismas condiciones en un perfil de canal con línea de vista representa una mejora significativa, lo que conlleva a analizar un mejor comportamiento de las redes desarrolladas en escenarios adversos.

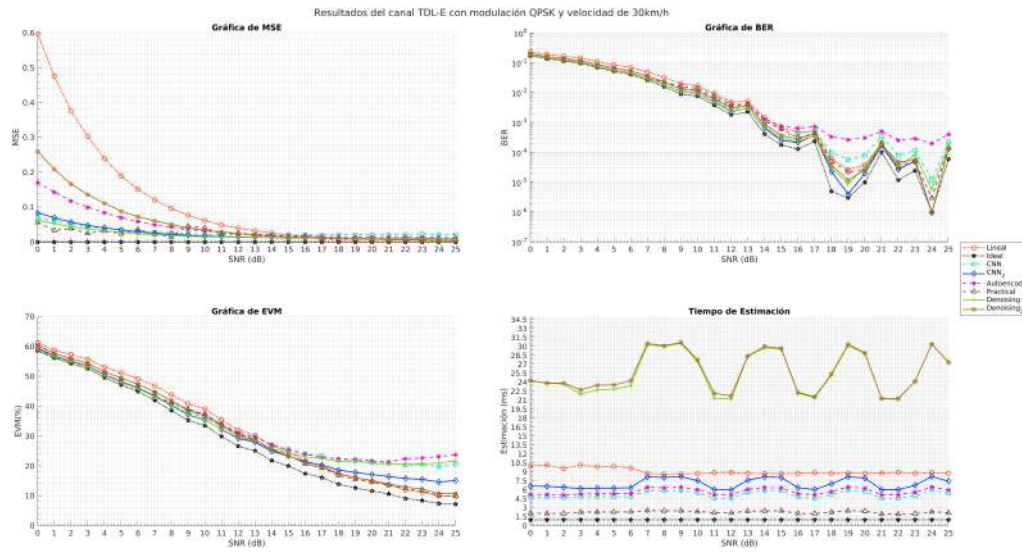


Figura 8.23: Velocidad Usuario 30 km/h

En términos de EVM se aprecia la misma tendencia descrita previamente en donde el modelo de la red denoising presenta los mejores resultados junto con la estimación práctica seguido por la red CNN 6, el comportamiento de los modelos es consistente ante cambios de la velocidad en donde el incremento de esta representa el aumento en la dispersión de los símbolos recibidos, sin embargo, para estos primeros resultados en un canal sin línea de vista el comportamiento del EVM ante el aumento de la relación señal a ruido no se descontrola para la modulación QPSK y continua decreciendo siguiendo un comportamiento ideal.

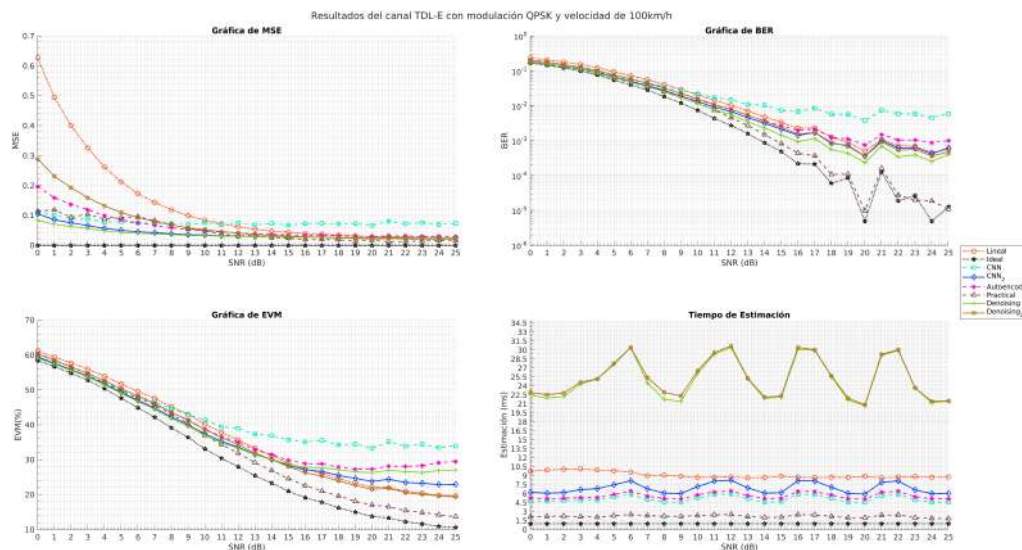


Figura 8.24: Velocidad Usuario : 100 km/h

Finalmente el tiempo de estimación se mantiene constante con los escenarios ya analizados por

lo que el cambio de canal no incide en el tiempo de estimación requerido por modelos, esto conlleva a comprender la adaptabilidad de los modelos diseñados ante los diversos perfiles de canal TDL.

Modulación 64 QAM

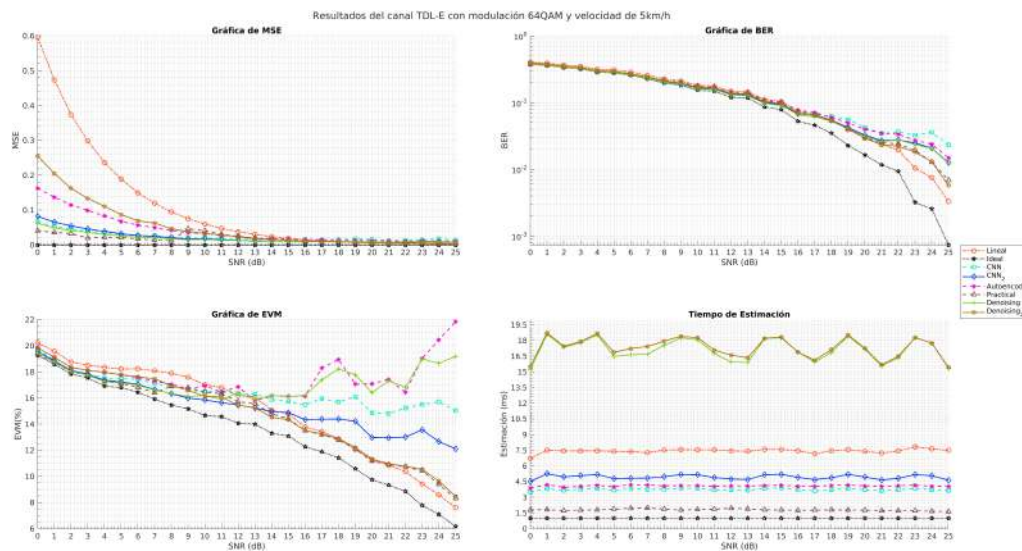


Figura 8.25: Velocidad Usuario 5 km/h

El impacto del cambio de modulación presentado en las gráficas 8.25, 8.26 y 8.27 se evidencia en el valor del EVM, donde se presenta una gran dispersión de los símbolos desde una velocidad baja, evidenciando la influencia de una simulación con más bits por símbolos. En el tiempo de estimación, se genera una vez más una disminución comparada con la modulación previa.

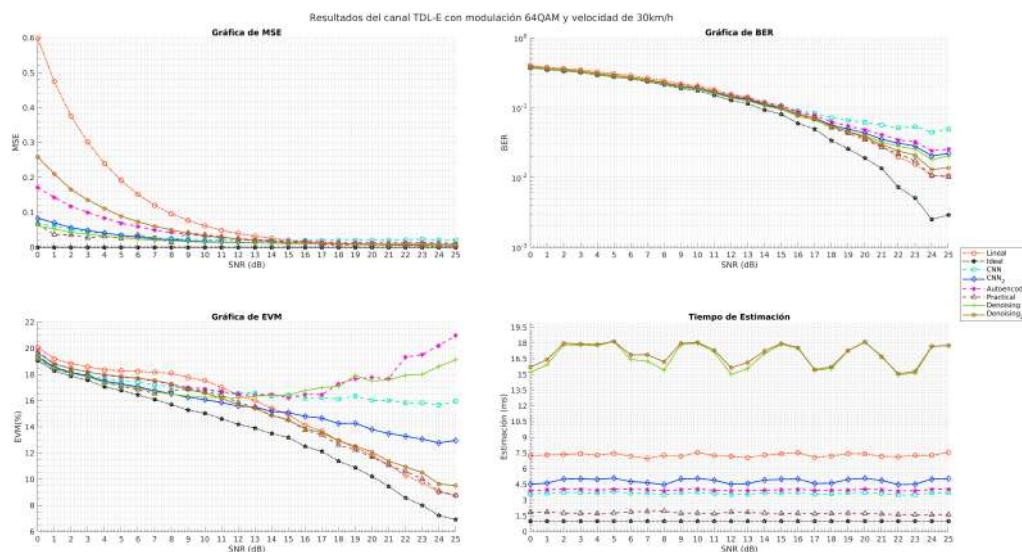


Figura 8.26: Velocidad Usuario 30 km/h

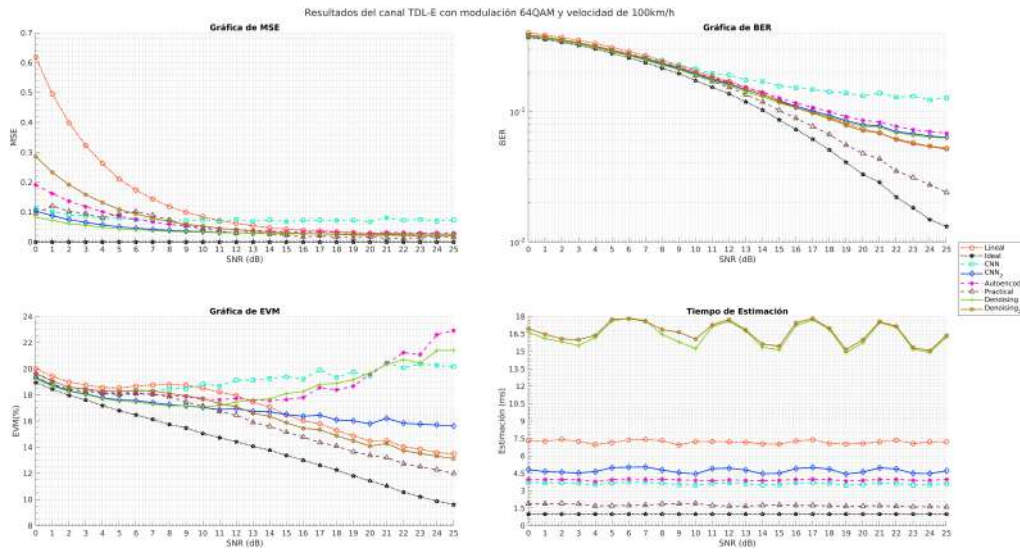


Figura 8.27: Velocidad Usuario : 100 km/h

Modulación 256 QAM

Un factor importante de la influencia del tipo de canal es la suavidad en los resultados de BER, lo que representa un comportamiento normal a medida que aumenta la SNR, para esta modulación especialmente se observa que la dispersión es mayor dada la alta sensibilidad que esta tiene ante interferencias y ruido, lo cual es el resultado esperado para esta modulación, a pesar de ello, los desempeños del Autoencoder y la red denoising así como la red CNN 4 son bastantes malos, el mejor modelo como es de esperar es la red denoising 2 seguido de la red CNN 6, siendo solamente un resultado bastante bueno el obtenido por la red Denoising 2 el cual es muy cercano al ideal.

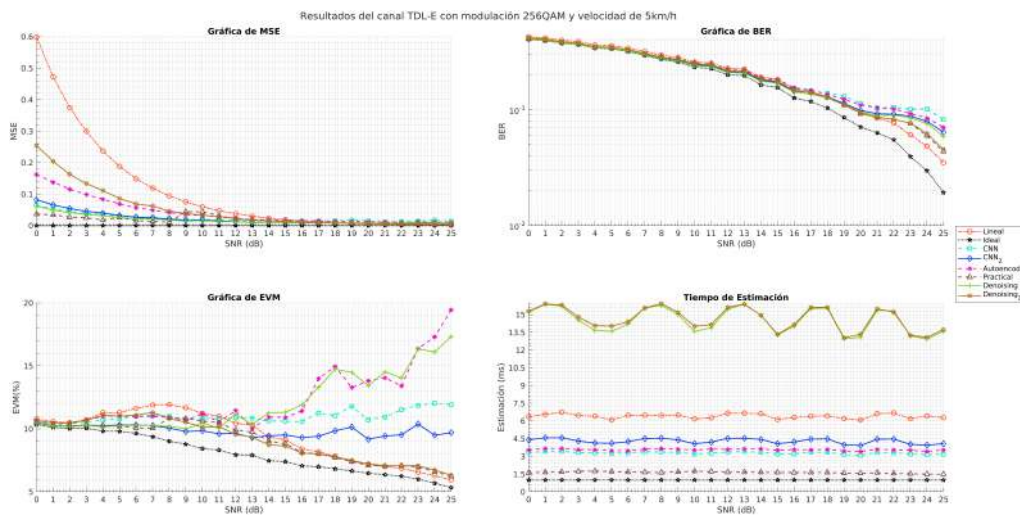


Figura 8.28: Velocidad Usuario 5 km/h

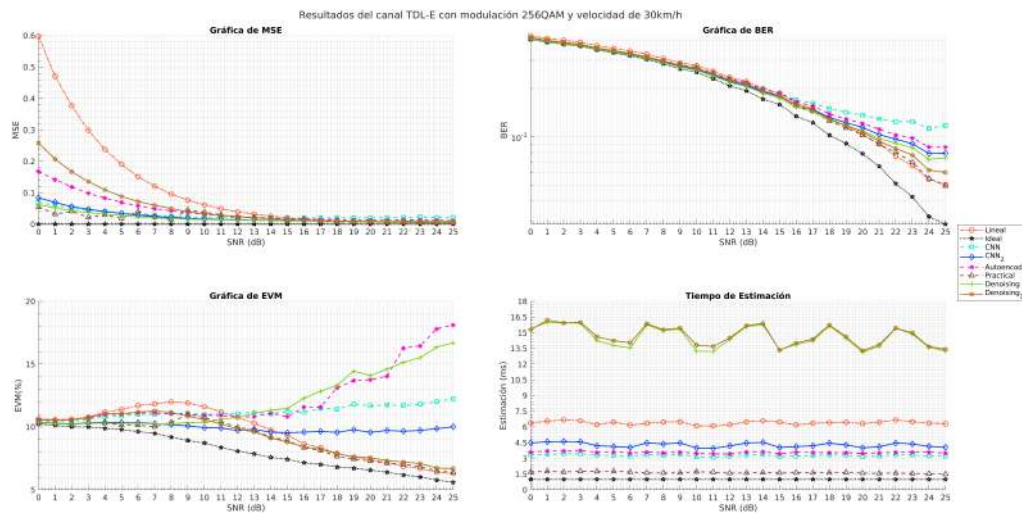


Figura 8.29: Velocidad Usuario 30 km/h

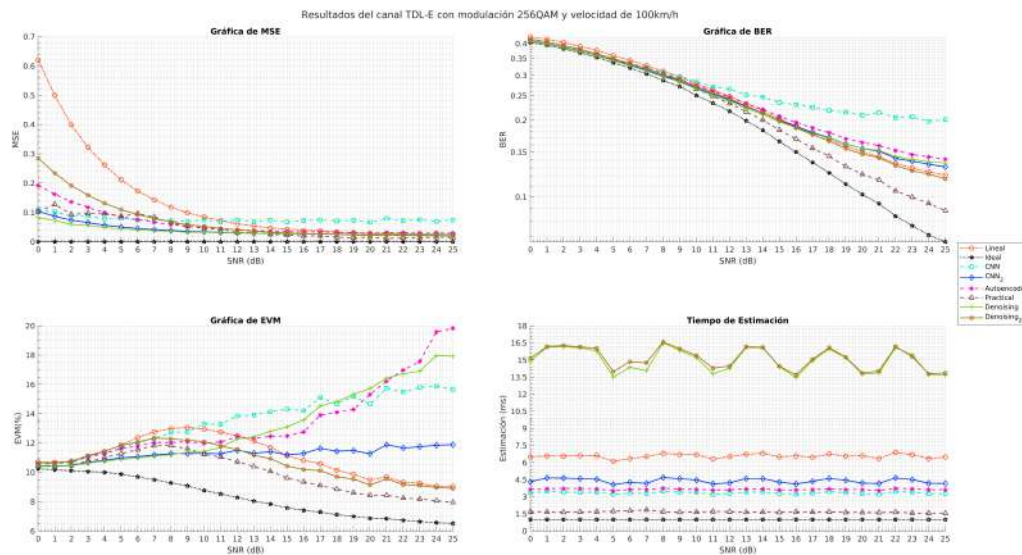


Figura 8.30: Velocidad Usuario : 100 km/h

Tras haber presentado los resultados de las pruebas hechas para cada uno de los modelos en diferentes escenarios, es pertinente destacar algunos aspectos generales, con los cuales se puede comprender de una mejor forma su desempeño y comportamiento dentro del sistema de comunicaciones propuesto.

Efectos de la modulación

La modulación afecta directamente en la dispersión de los símbolos recibidos, como consecuencia de ello se genera un valor mayor en la magnitud del vector de error, ocasionando un incremento de

errores en el receptor, derivando en la reducción de la métrica BER. Un aspecto que se debe tener en consideración, es que el fenómeno de dispersión afecta en igual medida a la estimación ideal por lo que los resultados finales de cada prueba, demuestran la capacidad de los modelos de tener una tendencia similar a la ideal en algunos intervalos de SNR, además, mejoran los resultados de los estimadores prácticos y lineales.

El tiempo de estimación es otro aspecto que se ve fuertemente influenciado por el tipo de modulación en donde para una modulación mas robusta el tiempo de estimación se reduce considerablemente, según lo apreciado el cambio de una modulación QPSK a una 64QAM mejora el tiempo de estimación entre un 40 - 45 % aproximadamente, y con respecto a 256QAM se presenta una mejora cercana al 13 % comparada con 64QAM.

Efectos de la velocidad

La velocidad tiene un impacto sustancial en el EVM ya que claramente para escenarios en donde el movimiento del usuario es alto la dispersión de los símbolos es mayor lo que afecta considerablemente el valor del BER y de la calidad de la información recibida, a pesar de esto, los modelos que destacan son la CNN 6 y Denoising, ya que estas responden de forma casi ideal logrando realizar una muy buena estimación de canal para velocidades superiores a 100 km/h.

Respecto a los tiempos de estimación y la métrica de la calidad de la estimación, se puede observar que las variaciones en ambas mediciones son mínimas, indicando la adaptabilidad y generalización de los modelos diseñados, siendo completamente independientes de este parámetro asociado con la dispersión por efecto Doppler.

Efectos canal LoS y NLoS

Los cambios en el tipo de canal TDL afectan los resultados de las métricas de EVM, los escenarios NLoS introducen en mayor medida distorsiones en las señales piloto, por lo que la estimación de canal es más desafiante cuando hay más atenuaciones y retardos. Como es de esperarse, cuando se presentan estas distorsiones los símbolos de las constelaciones se ven afectados, haciendo que se desplacen de sus posiciones ideales y aunado a ello, con la variación de la modulación, decae el rendimiento del sistema, ya que entre mayor orden de modulación, menos robusta se hace ante las interferencias. Por último, como es de esperarse, los estimadores ANN operan de una mejor forma en modelos de canal LoS, ya que la propagación de las señales es directa.

Conclusiones y líneas futuras

9.1 Conclusiones

- En el presente trabajo se desarrolló un sistema de comunicaciones móviles de quinta generación capaz de incorporar modelos neuronales para estimación adaptativa de canales, en total se diseñaron y probaron 13 modelos neuronales capaces de estimar correctamente el canal, de igual manera se construyeron 3 estimaciones de comparación (ideal, práctico e interpolación lineal). A partir de una primer etapa de selección se encontró que los mejores modelos para CNN eran las redes CNN 4 Y CNN 6, Autoencoder 1 y las 2 redes Denoising, los cuales se compararon en una etapa final de la que se encontró que los mejores modelos son **CNN 6 y Denoising 2**, siendo este último el mejor en cuanto resultados para métricas como EVM o BER con valores casi ideales en múltiples escenarios a costo de un tiempo de estimación considerable; por su parte, la red CNN 6 presenta muy buenos resultados siendo mas sensible a la dispersión de símbolos, no obstante, ofrece una eficiencia 3 veces mayor en términos de tiempo de estimación teniendo un tiempo de 4.5 ms lo cual es un resultado excelente considerando la cantidad de escenarios testeados, además esta totalmente abierta la posibilidad de optimización.
- Para las métricas de la calidad de la estimación y el tiempo de estimación, se halló la estrecha relación que tienen entre sí y es que los modelos que requieren de una mayor cantidad de parámetros entrenables, son aquellos en las que se obtienen los valores inferiores de MSE, pese a ello, debido a su complejidad y robustez requieren de un considerable costo computacional,

ocasionando un aumento en los tiempos de procesamiento para generar las salidas de acuerdo con los datos de entrada. Es por ello, que es necesario recurrir a técnicas de optimización que doten a los algoritmos de una mayor robustez manteniendo un equilibrio entre la calidad y la latencia.

- Los 6 modelos neuronales seleccionados para la etapa final son capaces de realizar una muy buena estimación de canal, con resultados cercanos a lo ideal. Cabe resaltar que se logró obtener en algunas condiciones mejores resultados que los estimadores práctico y lineal para valores muy bajos de SNR (entre 5 - 15 dB), lo que demuestra el potencial de la implementación de estos sistemas para escenarios reales y que representarían una oportunidad de mejora significativa con respecto a las tecnologías existentes.
- Los efectos asociados al entrenamiento de una red neuronal siempre son un riesgo latente cuando se trabaja con estas técnicas de aprendizaje profundo y este caso no es la excepción, se apreció en los resultados que a valores mayores de SNR en donde la presencia de ruido es menor en el sistema, la calidad de la respuesta de los modelos decae y se presenta mayor dispersión en los símbolos, por lo que se concluye que la solución planteada se ajusta a un rango de 0 dB hasta 20 dB en función del tipo de escenario, posterior a este valor dependerá fuertemente de las condiciones una mejor respuesta que se acerque a la referencia ideal.
- Se observó que la modulación tiene un impacto significativo en la dispersión de los símbolos recibidos, sobre todo con esquemas que usan más bits por símbolos, lo que afecta la tasa de errores de bits (BER). Aunque la modulación puede aumentar el valor del BER, los resultados muestran que los modelos de estimación logran resultados buenos, llegando a superar en múltiples condiciones a los estimadores prácticos y lineales. Además, se encontró que modulaciones más robustas reducen el tiempo de estimación de manera considerable, lo que sugiere una mejora en la eficiencia del sistema.
- En cuanto a la velocidad del usuario, se evidencia que esta tiene un impacto sustancial en el EVM, especialmente en escenarios de movimiento extremo. Sin embargo, los modelos de estimación muestran una capacidad casi ideal para estimar el canal incluso a velocidades superiores a 100 km/h, lo que indica una robustez considerable del sistema ante condiciones dinámicas.
- Los efectos del tipo de canal NLoS o LoS, muestran que el canal TDL puede introducir distorsiones significativas en la señal recibida, lo que aumenta el EVM. Respaldando las expectativas, los modelos neuronales demostraron un mejor rendimiento en escenarios LoS,

ya que en estos hay un trayecto directo para la propagación de las señales, y mostraron adaptabilidad óptima a diferentes perfiles de canal. Esto sugiere que los modelos neuronales tienen una alta capacidad de generalización, ya que un solo modelo es capaz de funcionar adecuadamente en cualquier entorno.

- En conjunto, estos hallazgos respaldan la eficacia y robustez del sistema de comunicaciones diseñado, demostrando la capacidad de los modelos de estimación neuronal para adaptarse y proporcionar estimaciones precisas del canal en una variedad de condiciones operativas. Estos resultados no solo son prometedores para la implementación práctica del sistema, sino que también, sugieren áreas de mejora y optimización para futuros desarrollos.

9.2 Futuras líneas de investigación

Este trabajo presenta la combinación de dos grandes mundos de la ingeniería electrónica que están en continua evolución como lo son las telecomunicaciones e inteligencia computacional, en ambos campos este trabajo presenta diversas oportunidades de mejora.

9.2.1 Enfocadas a Inteligencia Computacional

- Realizar la estimación de canales con diversos modelos de Deep Learning, como es el caso de las redes neuronales recurrentes las cuales pueden extraer información de estados previos para predecir el resto de la información y se puede reducir la dimensionalidad de los datos; también se propone el uso de modelos como las redes generativas adversarias (GAN), en la que el generador se encarga de estimar el canal a partir de las señales de referencia mientras que el discriminador sería capaz de distinguir las etiquetas de las estimaciones del generador para optimizar el desempeño del modelo.
- Generar los datos de entrenamiento con muestras distribuidas de manera uniforme en los valores SNR, de esta forma se podría aumentar el rango de funcionalidad de los modelos, sin verse afectada por diferentes niveles de ruido.
- Optimización de los modelos propuestos en términos de rendimiento y precisión, a partir de técnicas de regularización como *dropout* o haciendo variaciones en hiper-parámetros de las capas de cada modelo o del entrenamiento (optimizadores, inicialización de pesos, tasas de aprendizaje).

- Cambiar el proceso de interpolación lineal por una estimación más robusta como la MMSE y usar esto como los nuevos datos de entrada al modelo neuronal tanto en la fase de entrenamiento como de prueba, de esta forma se pueden completar los procesos de estimación con el objetivo de optimizar los resultados en diferentes escenarios.
- Migración del sistema diseñado a un lenguaje programación enfocado al Deep Learning como Python.

9.2.2 Enfocadas a las Telecomunicaciones

- En el ámbito de las comunicaciones el sistema puede ser mejorado mediante la inserción de sistemas MIMO, el trabajo abre la oportunidad de analizar el comportamiento para estos sistemas, de igual manera es posible escalar el sistema a capa 2 permitiendo realizar el envío de mayor información de forma eficaz, así como en este trabajo se construyo la base para poder agregar codificación, empaquetado y retransmisión de datos usando procesos HARQ.
- Exploración de otros tipos de canales de propagación como por ejemplo CDL o comunicaciones satelitales, incorporación de otro tipo de canal de transporte DL-SCH, PUSCH.
- Variar la densidad de las señales piloto para contrastar los efectos que estas ocasionan tanto en la calidad de la estimación como en la capacidad del canal de comunicaciones.
- Pruebas de funcionamiento con diversas numerologías disponibles en 5G así como diversos valores de anchos de banda.
- Simulación con la inserción de otros símbolos presentes en los sistemas de comunicaciones 5G como podrían ser las señales de control.
- Implementación del sistema actual en un ambiente real usando SDR, a través de las librerías incorporadas en MATLAB o en Python.

9.3 Aportes

1. Generación de un conjunto de datos de entrenamiento con las diversas condiciones de los canales de comunicaciones 5G tipo TDL.
2. Emulación de un sistema de comunicaciones real en MATLAB, el cual sienta las bases para la construcción de nuevas líneas de investigación.

3. Implementación de redes de neuronales CNN y modelos innovadores como Autoencoders y redes Denoising para la estimación de canales de quinta generación.
4. Implementación de bloques de medida de EVM, BER, MSE y tiempo de estimación para un sistema 5G sobre MATLAB, validación con procesos convencionales y teórico, adicionalmente, se hacen simulaciones para múltiples canales TDL.
5. Implementación de modulaciones QPSK, 16QAM, 64QAM y 256QAM en sistemas móviles con velocidades de usuario entre (0 y +100 km/h), entorno para la evaluación de parámetros de calidad.
6. Desarrollo de mecanismos para la transmisión de datos (imagenes o información binaria) sobre un canal de comunicaciones móviles 5G empleando técnicas de deep learning con resultados próximos a las referencias ideales.
7. Desarrollo de herramientas de simulación y enseñanza para la transmisión y caracterización del comportamiento de modelos estimadores de canal en sistemas 5G.

A

Anexo : Resultados canales TDL-A, TDL-C, TDL-D

En este apartado se presentan las simulaciones resultantes para otros perfiles de canales.

A

A.1 TDL-A (NLoS)

A.1.1 Modulación QPSK

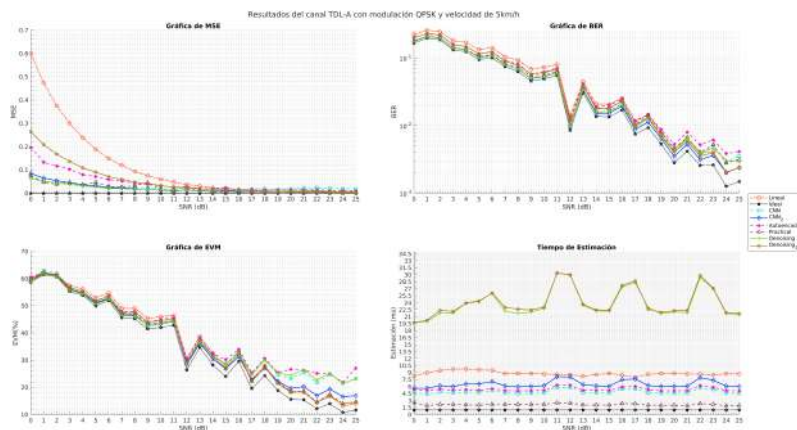


Figura A.1: Velocidad Usuario 5 km/h

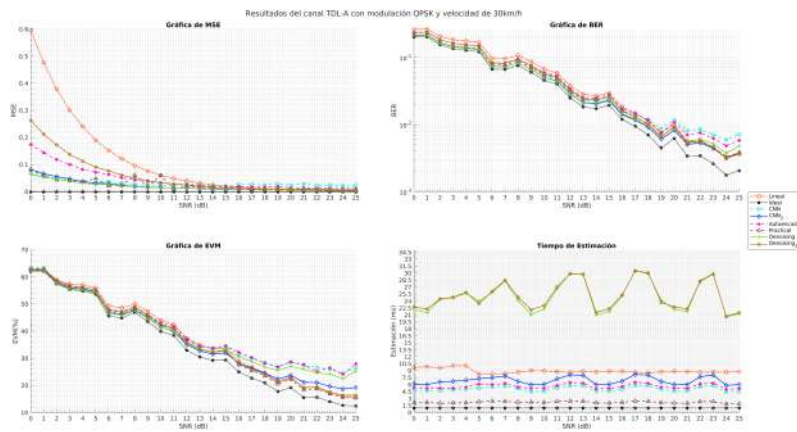


Figura A.2: Velocidad Usuario 30 km/h

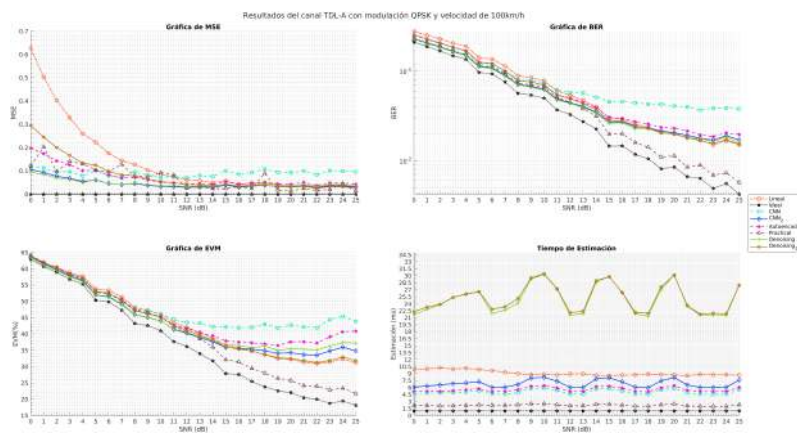


Figura A.3: Velocidad Usuario : 100 km/h

A.1.2 Modulación 64 QAM

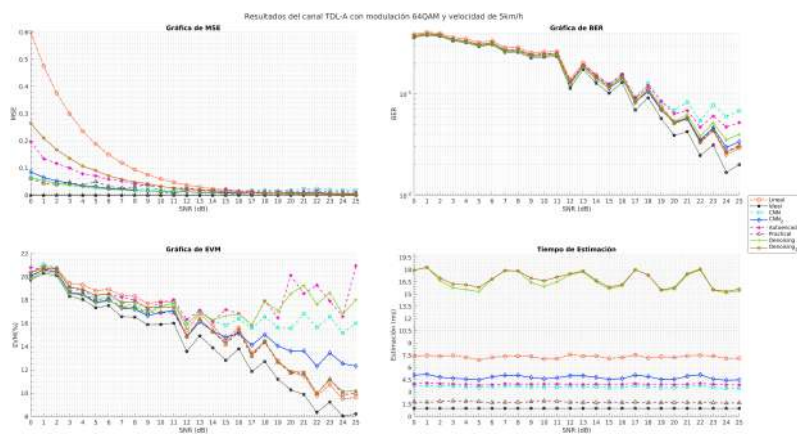


Figura A.4: Velocidad Usuario 5 km/h

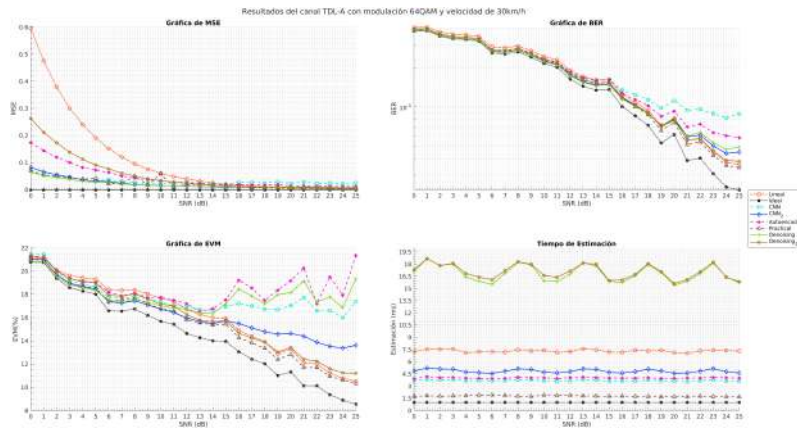


Figura A.5: Velocidad Usuario 30 km/h

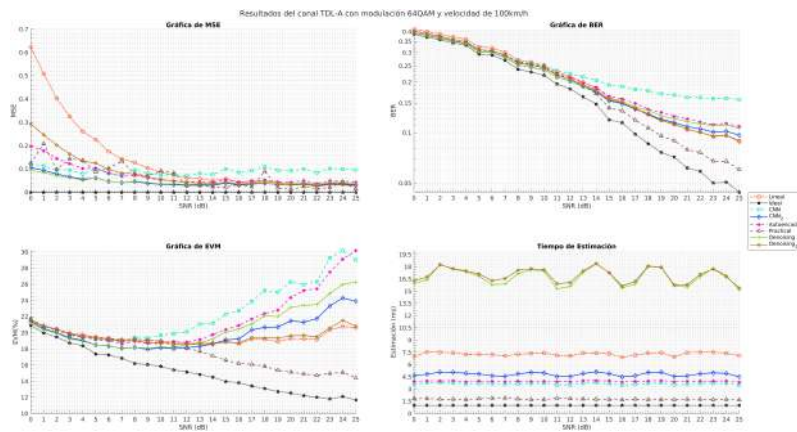


Figura A.6: Velocidad Usuario : 100 km/h

A.1.3 Modulación 256 QAM

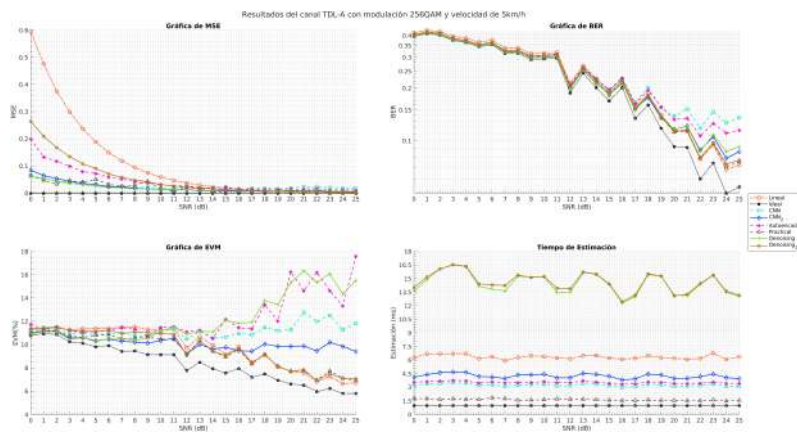


Figura A.7: Velocidad Usuario 5 km/h

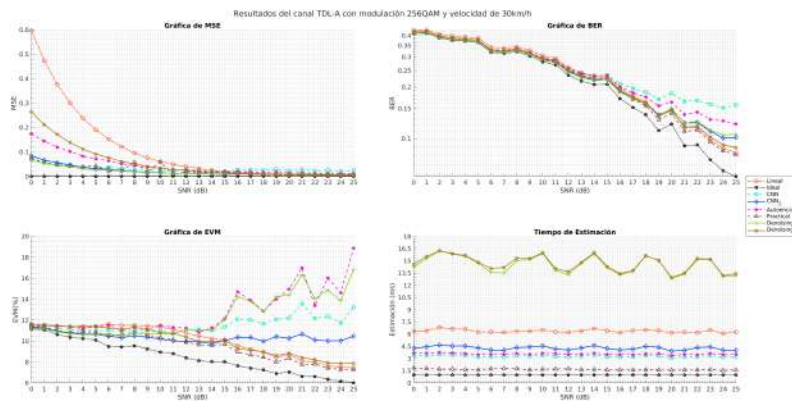


Figura A.8: Velocidad Usuario 30 km/h

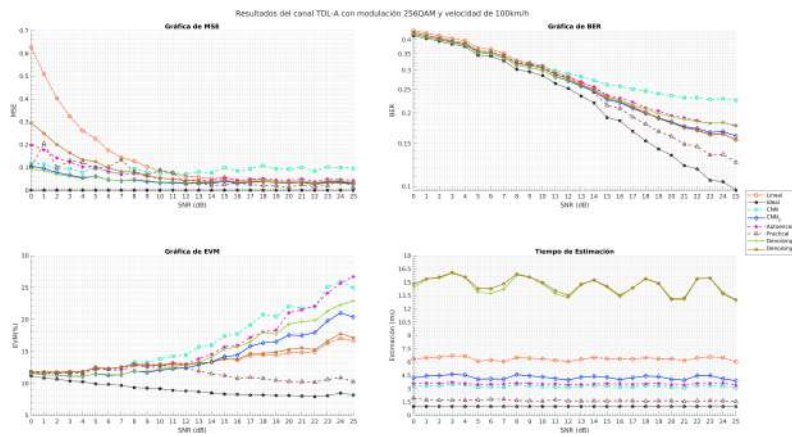


Figura A.9: Velocidad Usuario : 100 km/h

A.2 TDL-C (NLoS)

A.2.1 Modulación QPSK

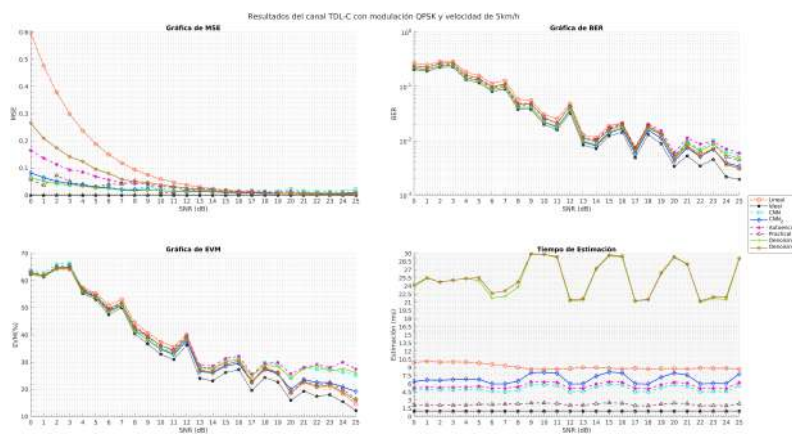


Figura A.10: Velocidad Usuario 5 km/h

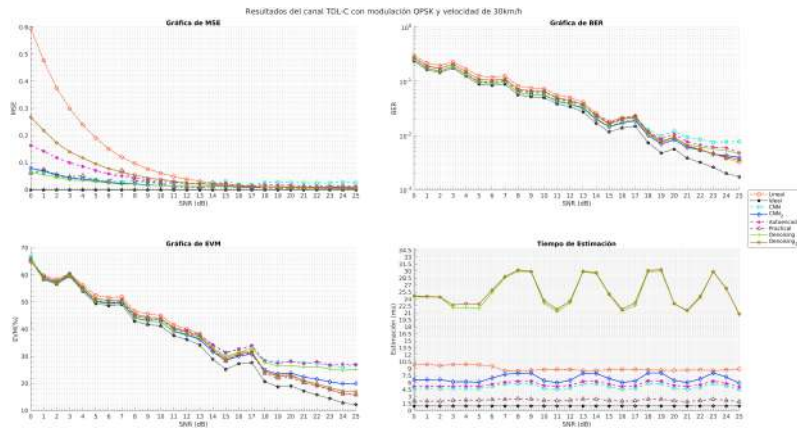


Figura A.11: Velocidad Usuario 30 km/h

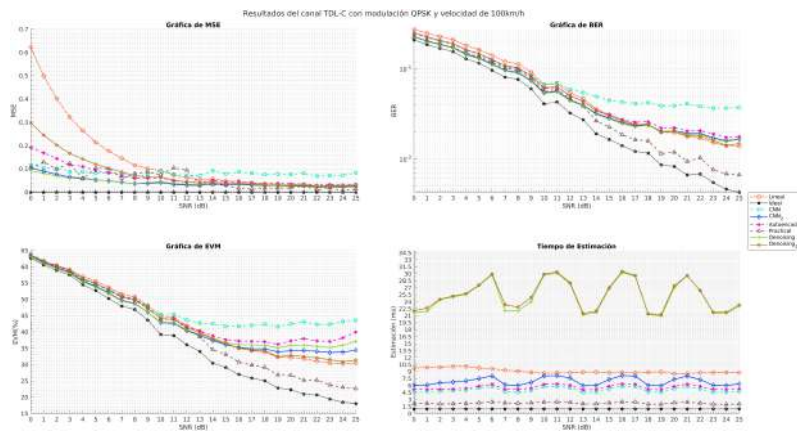


Figura A.12: Velocidad Usuario : 100 km/h

A

A.2.2 Modulación 64 QAM

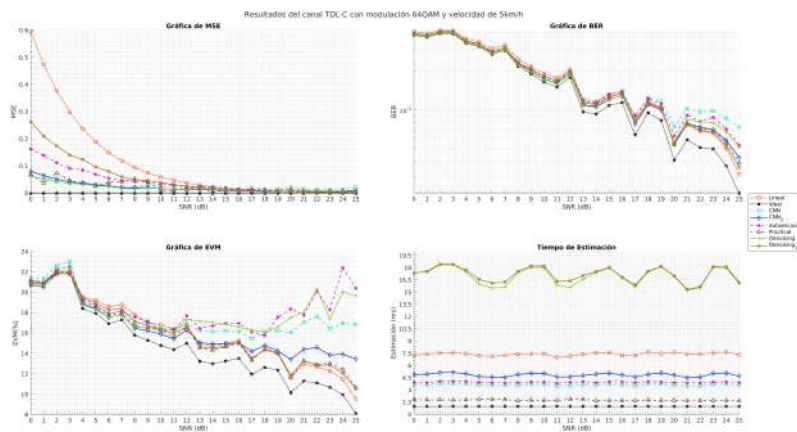


Figura A.13: Velocidad Usuario 5 km/h

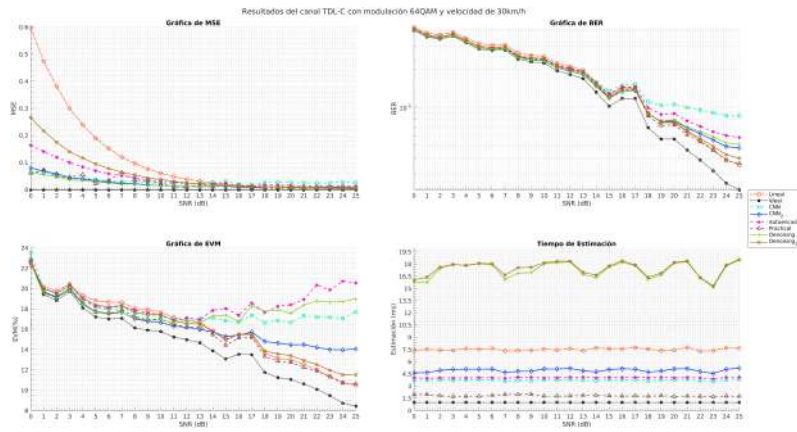


Figura A.14: Velocidad Usuario 30 km/h

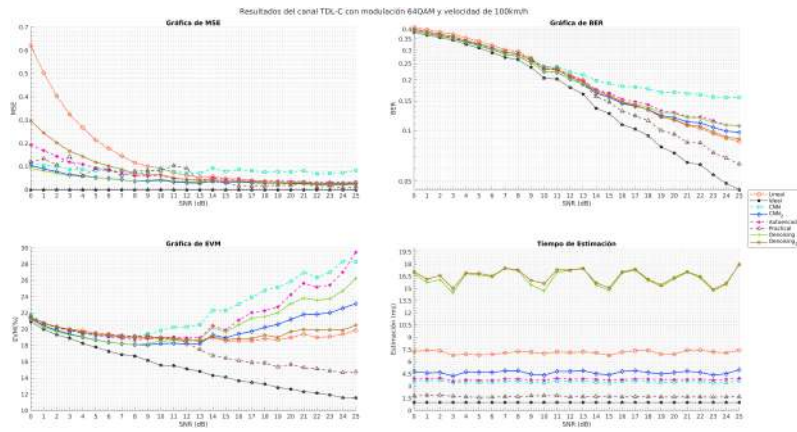


Figura A.15: Velocidad Usuario : 100 km/h

A.2.3 Modulación 256 QAM

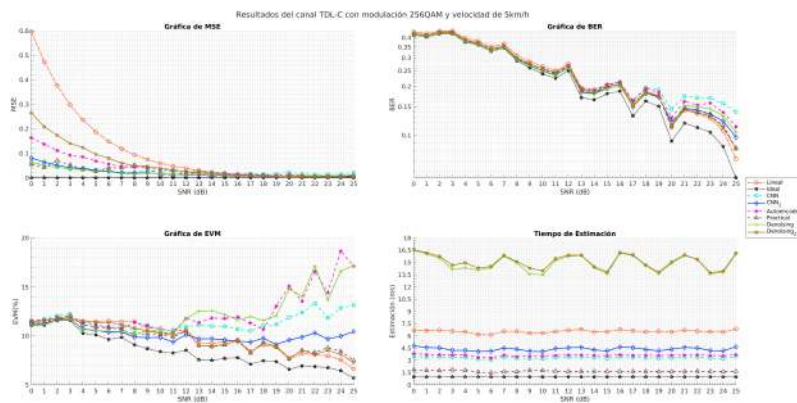


Figura A.16: Velocidad Usuario 5 km/h

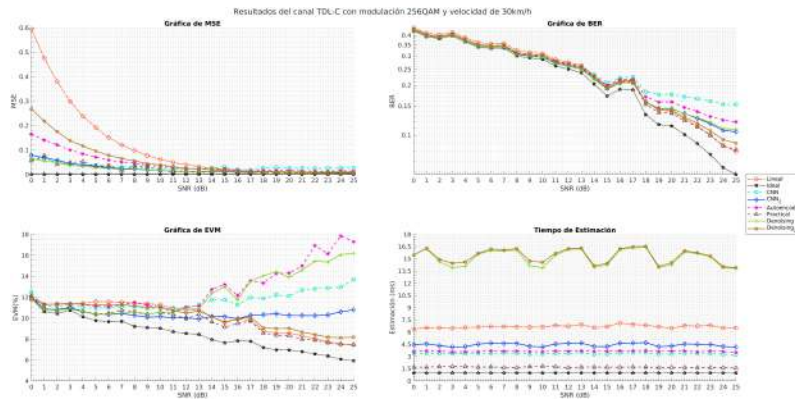


Figura A.17: Velocidad Usuario 30 km/h

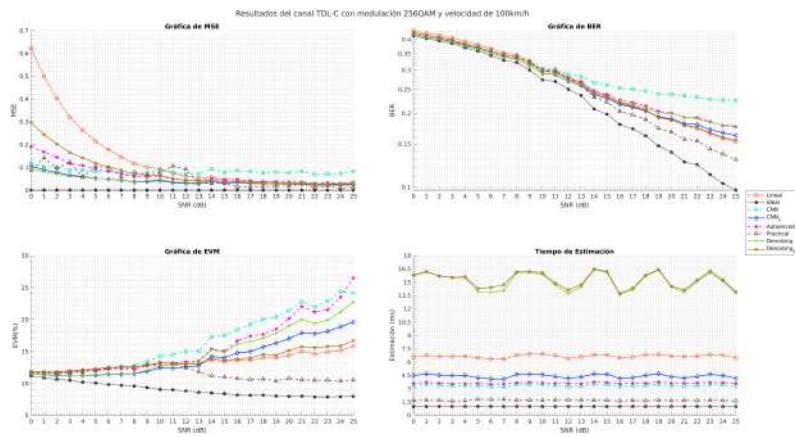


Figura A.18: Velocidad Usuario : 100 km/h

A

A.3 TDL-D (LoS)

A.3.1 Modulación QPSK

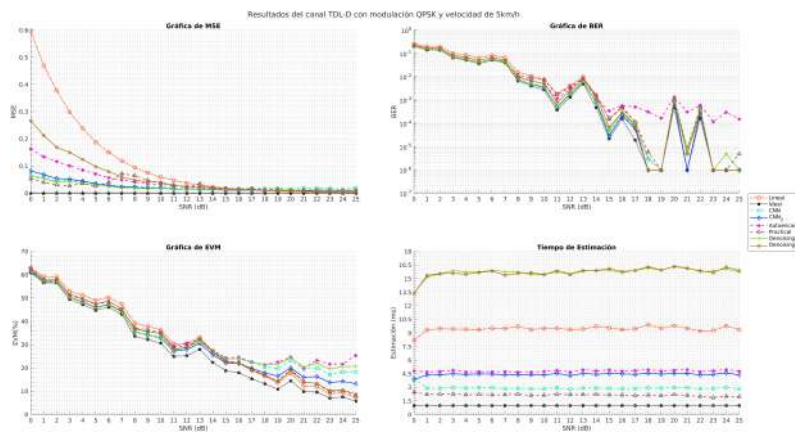


Figura A.19: Velocidad Usuario 5 km/h

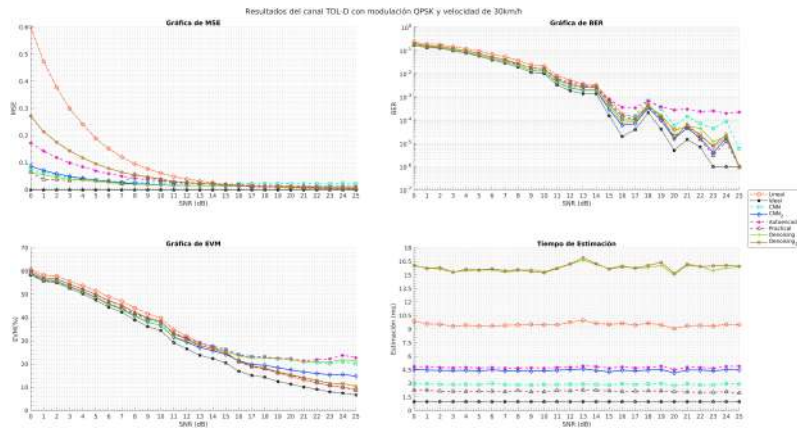


Figura A.20: Velocidad Usuario 30 km/h

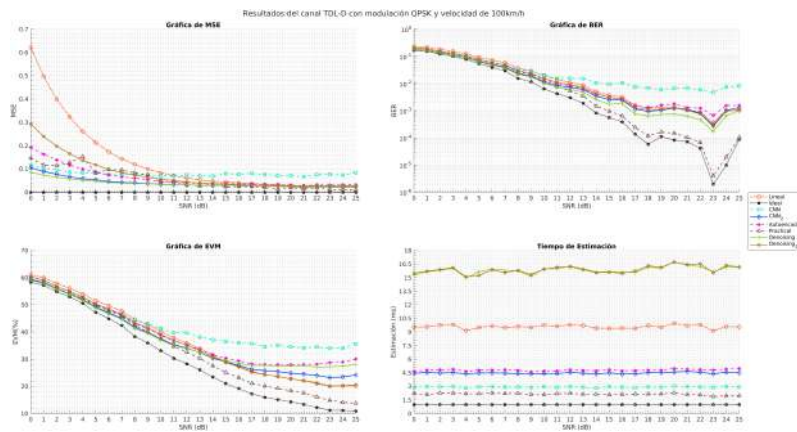


Figura A.21: Velocidad Usuario : 100 km/h

A

A.3.2 Modulación 64 QAM

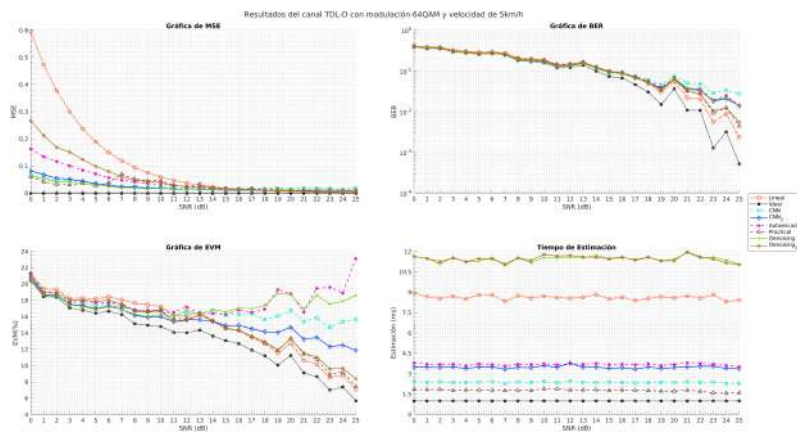


Figura A.22: Velocidad Usuario 5 km/h

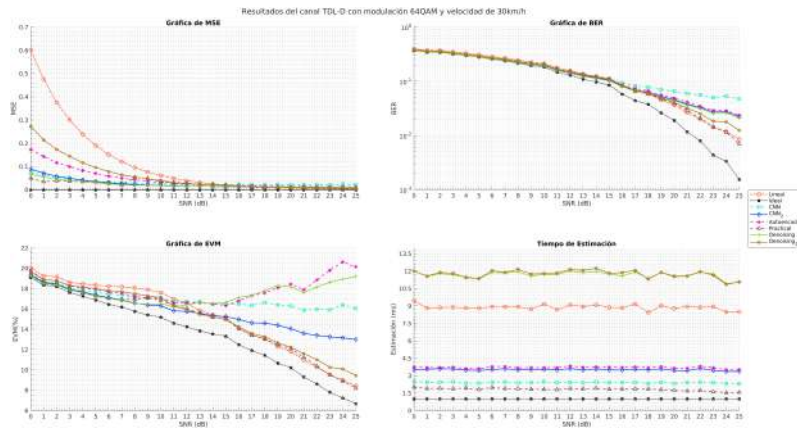


Figura A.23: Velocidad Usuario 30 km/h

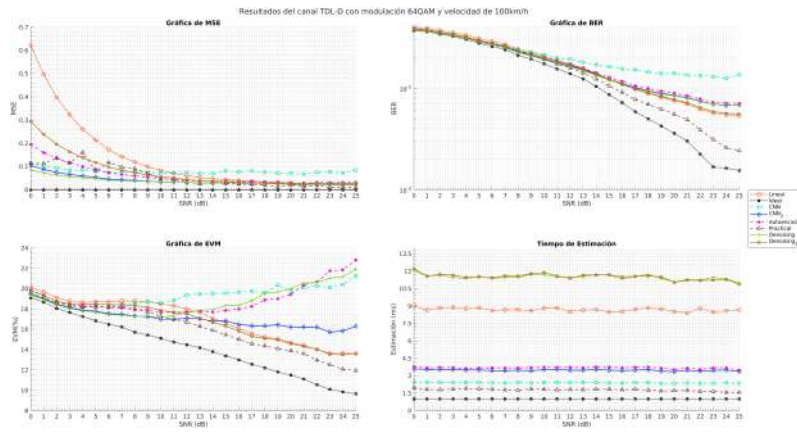


Figura A.24: Velocidad Usuario : 100 km/h

A.3.3 Modulación 256 QAM

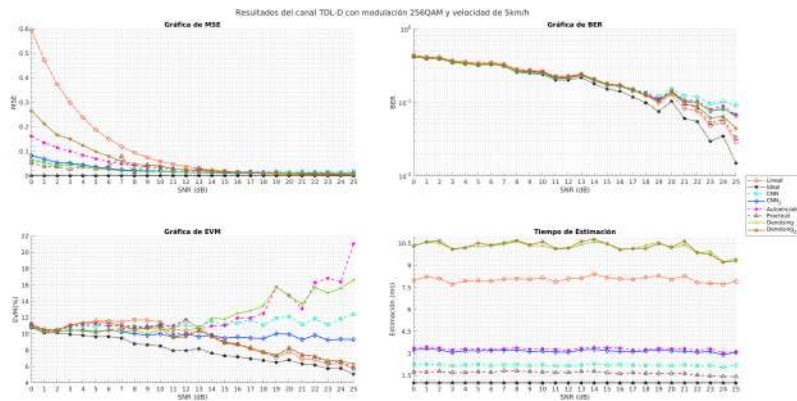


Figura A.25: Velocidad Usuario 5 km/h

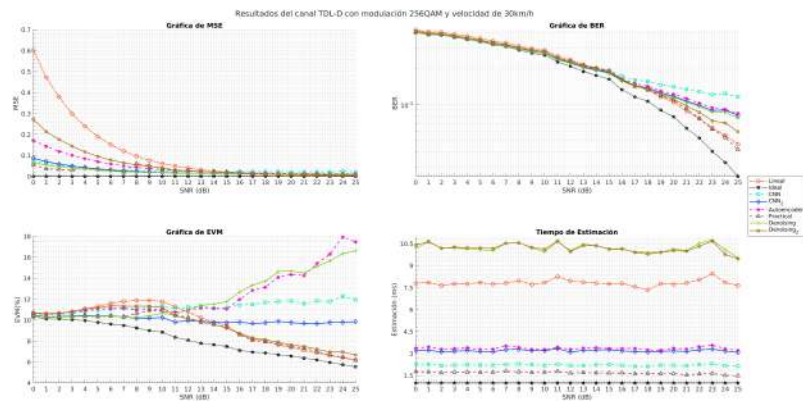


Figura A.26: Velocidad Usuario 30 km/h

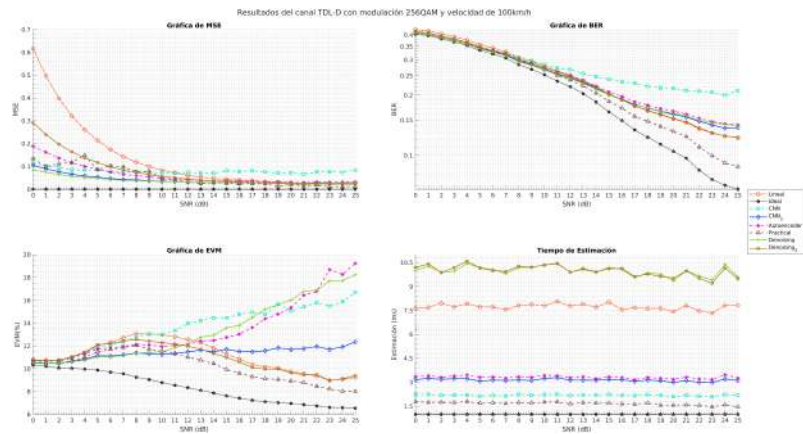


Figura A.27: Velocidad Usuario : 100 km/h

B

Anexo : Herramienta de simulación y aprendizaje 5G NR (DLS5G)

En este documento se presenta el manual de usuario para la App diseñada que permite interactuar con los modelos desarrollados, crear nuevos modelos, realizar pruebas en tiempo real, analizar datos ya simulados y probar nuevas configuraciones, el código fuente se encuentra disponible en el siguiente repositorio :

Repositorio 5G NR - Técnicas de Deep Learning enfocadas a la estimación adaptativa de canales en redes de quinta generación.

B

B.1 Arquitectura del sistema

En este apartado se presentan y enumeran los módulos, complementos y recursos involucrados en el funcionamiento del programa, en primer lugar se presentarán a nivel general la distribución de archivos en los folders principales diseñados, esta distribución se aprecia en la figura B.1.

Posterior a esto se presenta una listado de los archivos principales dentro de cada carpeta y de los cuales se realizara una explicación mas o detalle en los siguientes apartados.

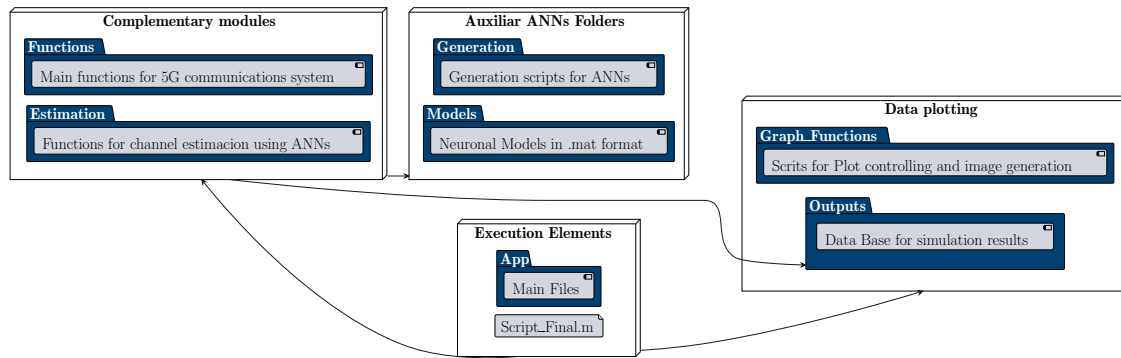


Figura B.1: Estructura principal de directorios

Como se aprecia existen 8 carpetas principales distribuidas en 4 secciones, cada una de estas carpetas a su vez contienen archivos y directorios complementarios requeridos por la APP, la primera carpeta correspondiente a módulos complementarios contiene las funciones necesarias para emular el sistema de comunicaciones, así como para la inclusión de la estimación de canal para diversos modelos de redes neuronales B.5, estos modelos son generados y almacenados en 2 carpetas presentadas en la figura B.4, finalmente el software cuenta con una zona de almacenamiento que a su vez funciona como base de datos con la cual graficar los resultados obtenidos, la distribución de archivos se aprecia en B.2

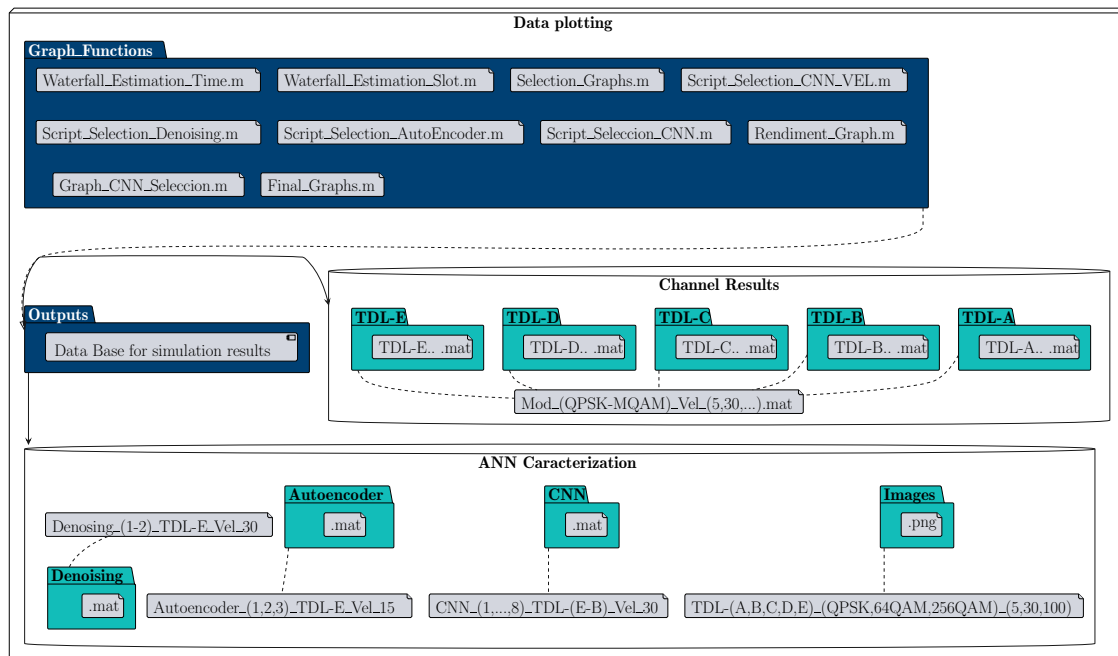


Figura B.2: Archivos resultantes simulaciones

Un directorio importante por ser listado es el correspondiente a la interfaz gráfica del programa, los archivos involucrados para el correcto funcionamiento de esta se presentan en B.3

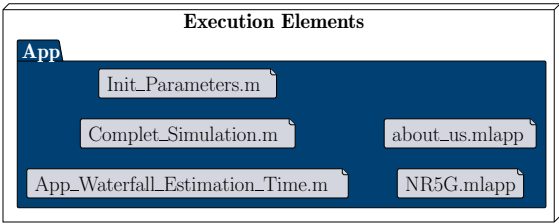


Figura B.3: Ficheros GUI

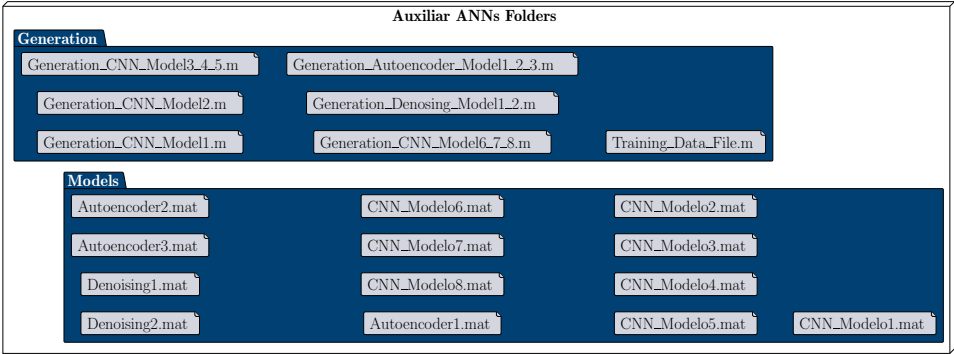


Figura B.4: Ficheros Auxiliares

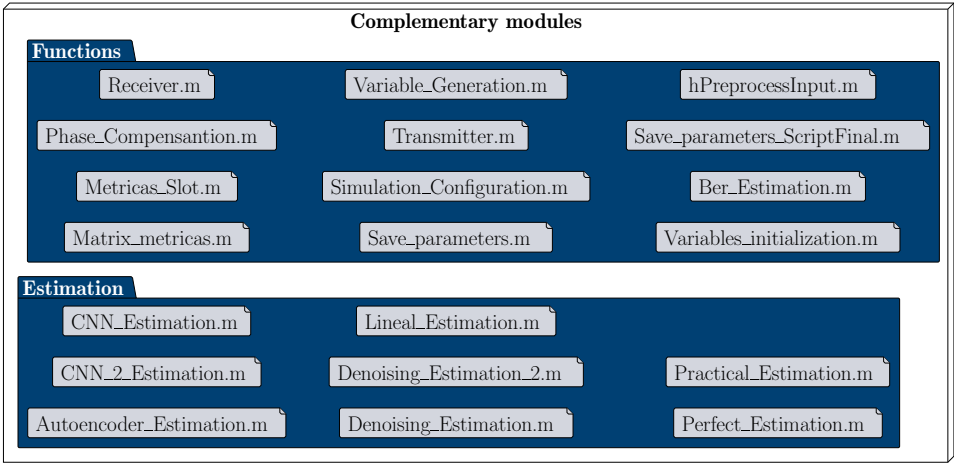


Figura B.5: Ficheros Módulos complementarios

B.2 Diseño del modelamiento de datos

Existe una estructura de datos que contiene todos los elementos dentro de la simulación denominada **Pam_sim**, esta estructura almacena desde los objetos de configuración hasta los bits recibidos, por lo que se hace la notación que cuando se haga referencia a un elemento como la frecuencia central $f_c = 5\text{ GHz}$ realmente presenta la configuración de la frecuencia de la portadora dentro de la simulación y podrá ser encontrada por medio de invocación del objeto **Pam_sim.fc**.

La modularidad implica un gran problema a nivel de control global de variables por lo que la estructura **Pam_sim** soluciona este problema, por lo que la salida de cada una de estas funciones siempre retorna la estructura con los cambios o nuevas variables utilizadas en la función.

Para la obtención de las métricas de evaluación presentadas en la sección 8 y 8.4 se tuvieron las siguientes consideraciones :

- El cálculo del MSE y EVM se realiza por cada ranura procesada, por lo que para cada iteración se tendrá un vector de tamaño $(10 \frac{\text{subtramas}}{\text{trama}} \times N \text{ tramas}) \times \text{len}(\text{SNR})$, en el caso de la simulación presentada se obtuvieron vectores de 70×26 (SNR de 0 dB-25 dB para 7 tramas). Para las gráficas presentadas en las secciones mencionadas no es necesario obtener esta métrica en cada ranura, por lo que se obtiene un valor promedio a lo largo del tiempo de transmisión obtenido en cada iteración del barrido SNR, es decir, si se envían 7 tramas con una numerología $\mu = 0$ se tendrá un total de 70 ms teóricos de transmisión para cada iteración de SNR y de este intervalo de tiempo se obtendrá el valor promedio y es el que se presenta en las gráficas.
- El término *Tiempo de estimación* hace referencia al valor normalizado que tarda en ejecutarse la instrucción de estimación de canal dentro de la secuencia de procesamiento en MATLAB, este valor se obtiene empleando las funciones *tic* y *toc*. Cuando se hace referencia al valor normalizado significa que es un valor llevado a lo teórico ya que la utilización de estas funciones permite ver el tiempo de ejecución de la instrucción en función de la máquina encargada de realizar la simulación del programa, por lo que con el fin de evitar sesgar los resultados a la capacidad de procesamiento de los equipos utilizados se normalizó este valor considerando el tiempo de ejecución de estimación de canal del modelo Ideal como 1 ms, esto se obtiene dividiendo todo el vector de tiempo de salida de la función *tic* entre el vector de tiempo de estimación ideal, de modo que se podría expresar como :

$$t_{norm} = \frac{t_{ANN}}{t_{ideal}} \quad (\text{B.1})$$

Este funcionamiento es el mismo a nivel de script como de GUI, sin embargo a nivel de GUI se agrega otro nivel de encapsulamiento en donde ahora la estructura **Pam_sim.fc** se almacena en el objeto principal de la aplicación **app**, por lo que para interactuar con los elementos descritos en el ejemplo anterior se utilizaría **app.Pam_sim.fc**.

B.2.1 Funcion Simulation_Configuration

Esta función llamada *Simulation_Configuration* realiza la configuración de una simulación de transmisión en el contexto de comunicaciones móviles 5G NR, en la figura B.6 se aprecia el diagrama de clases de esta función.

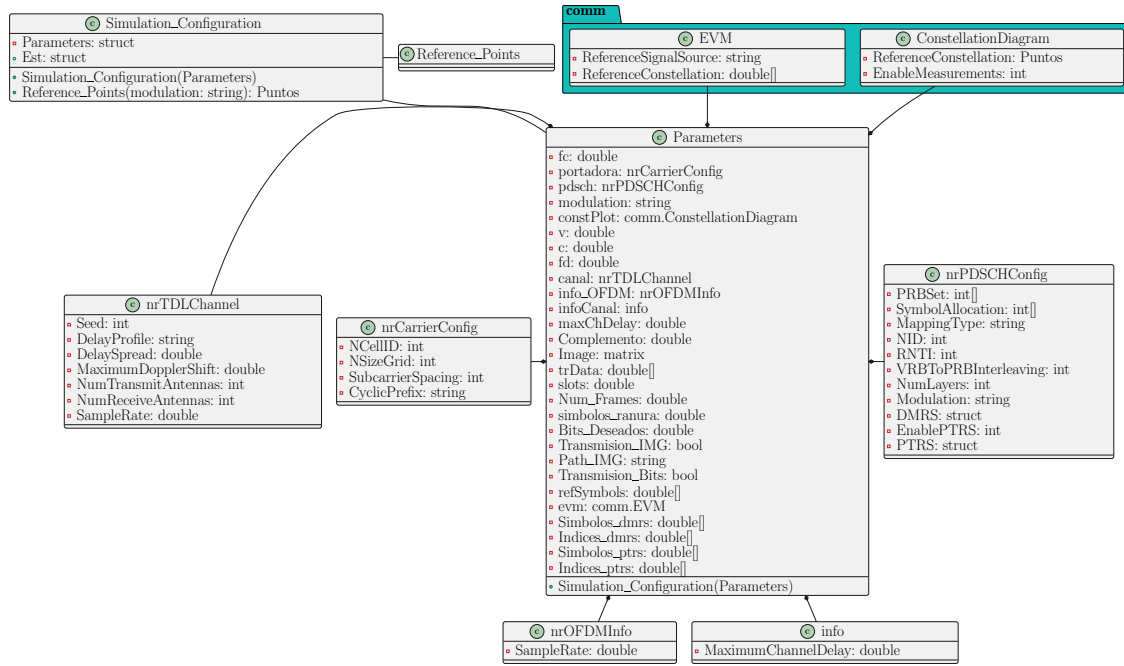


Figura B.6: Diagrama de clases función de configuración de simulación

A continuación se detalla cada parte de la función:

1. **Entrada de parámetros:** La función toma un único parámetro de entrada **Parameters**, que es una estructura que contiene todos los parámetros necesarios para configurar la simulación.
2. **Configuración de la portadora 5G NR:**
 - **fc:** Frecuencia de la operadora, en el caso de ejemplo del proyecto realizado establecida en 5 GHz.
 - **portadora:** Se inicializa y configura el objeto **nrCarrierConfig** para definir los parámetros de la portadora, como el número de celdas (**NCellID**), el tamaño de la cuadrícula (**NSizeGrid**), el espaciado entre subportadoras (**SubcarrierSpacing**), y el prefijo cíclico (**CyclicPrefix**).
3. **Configuración de los parámetros PDSCH (Physical Downlink Shared Channel):**

- **pdsch**: Se configuran los parámetros del canal físico de enlace descendente compartido, como el conjunto de recursos de bloques de recursos físicos (**PRBSet**), la asignación de símbolos (**SymbolAllocation**), el tipo de mapeo (**MappingType**), el identificador de celda (**NID**), la identidad de Radio Network Temporary Identifier (**RNTI**), entre otros.
4. **Definición de los parámetros DM-RS (Demodulation Reference Signal):**
 - Se especifican los parámetros relacionados con las señales de referencia para la demodulación, como los puertos DM-RS (**DMRSPortSet**), la posición del tipo A DM-RS (**DMRSTypeAPosition**), la longitud de DM-RS (**DMRSLength**), entre otros.
 5. **Definición de los parámetros PT-RS (Phase Tracking Reference Signal):**
 - Se activan y configuran los parámetros de las señales de referencia para el seguimiento de fase.
 6. **Configuración del diagrama de constelación:**
 - Se establece un objeto para graficar la constelación y se activa la medida del EVM (Error Vector Magnitude).
 7. **Generación del canal de propagación TDL (Tapped Delay Line):**
 - Se establecen los parámetros del canal de propagación, incluyendo el perfil de retardo, la dispersión de retardo, y el corrimiento Doppler máximo.
 8. **Configuración de elementos adicionales:**
 - Se establecen otros parámetros y objetos necesarios para la simulación.
 9. **Generación de datos de transmisión:**
 - Dependiendo de las condiciones establecidas en **Parameters**, se generan datos de transmisión, ya sea a partir de una imagen (**Parameters.Transmission_IMG**), bits aleatorios (**Parameters.Transmission_Bits**), o transmisión de una cantidad específica de frames.
 10. **Cálculo del RMS EVM (Error Vector Magnitude):**
 - Se calcula el error RMS EVM utilizando los símbolos de referencia.
 11. **Obtención de símbolos e índices DM-RS y PT-RS:**
 - Se generan los símbolos e índices necesarios para las señales de referencia DM-RS y PT-RS.

12. Reinicio de variables:

- Se reinician algunas variables que son necesarias en cada iteración de la simulación.

13. Salida de resultados:

- La función devuelve la estructura **Parameters**, que contiene todos los parámetros configurados para la simulación.

B.2.2 Variable_Generation

La función *Variable_Generation*, genera variables con un formato específico, que consiste en Complement1_NomModel_Complement2, donde NomModel es un nombre de modelo contenido en la estructura **Parameters**. La función toma los siguientes parámetros:

- **Parameters**: Estructura que contiene los parámetros necesarios para la generación de variables.
- **Complement1**: Complemento inicial para el nombre de la variable.
- **Complement2**: Complemento final para el nombre de la variable.
- **arreglo**: Booleano que indica si se debe inicializar la variable como un arreglo vacío (**true**) o con un valor cero (**false**).

La función itera sobre los modelos contenidos en **Parameters.models** y, para cada uno de ellos, genera una variable según el formato especificado. Si **arreglo** es **true**, la variable se inicializa como un arreglo vacío; de lo contrario, se inicializa con el valor cero. Esta función es de gran importancia ya que permite crear todas las variables requeridas para el almacenamiento de las métricas en función de los modelos seleccionados, es decir que permite ahorrar espacios de memoria en caso de no requerirse todos los modelos, así como permite el dinamismo para agregar nuevos modelos, cambiar nombres y demás criterios como oportunidad de mejora, el diagrama de clases se presenta en la figura B.7.

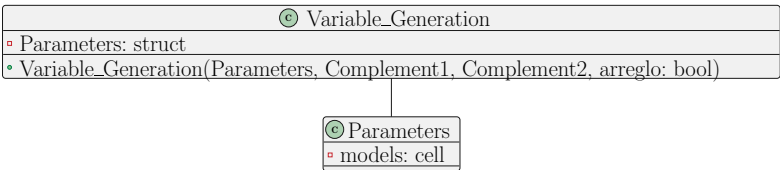


Figura B.7: Diagrama de clases función de Generación de variables

B.2.3 Variables_initialization

La función *Variables_initialization*, inicializa variables en cada transmisión a un valor específico de SNR. La función toma los siguientes parámetros:

- **Parameters:** Estructura que contiene los parámetros necesarios para la inicialización de variables.
- **snr:** Valor de la relación señal-ruido para la inicialización.

La función realiza las siguientes acciones:

1. Inicializa algunas variables específicas.
2. Crea variables utilizando la función *Variable_Generation* con diferentes complementos (*_EVM*, *_MSE*, *_Time*, etc.), según se requiera.
3. Asigna algunos valores iniciales específicos a ciertas variables.

Finalmente, devuelve la estructura **Parameters** actualizada, el diagrama de clases se presenta en la figura B.8.

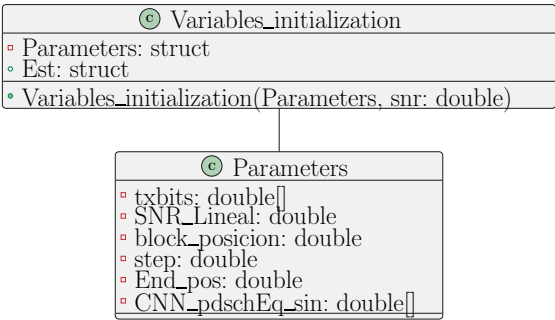


Figura B.8: Diagrama de clases función de Inicialización de variables

B.2.4 Transmitter

Esta función se encarga de generar y modular los datos a transmitir, utiliza los parámetros configurados en la estructura **Pam_sim** la cual es enviada a la función y empleada bajo el nombre **Parameters**, la funcion realiza el siguiente procedimiento :

1. **Definición del número de slots:** Establece el número de slots de acuerdo al índice de ranura proporcionado en `Parameters`
2. **Generación de codewords aleatorias:** Si se va a transmitir una imagen u otros datos especificados en `Parameters.trData`, la función calcula cuántos bits quedan por enviar y procede a dividirlos en bloques de tamaño especificado por `Parameters.step`. Si quedan más bits por enviar que el tamaño del bloque, se envían los bits restantes y el resto del bloque se completa con ceros. Si no se van a enviar datos específicos (`Transmission_IMG` y `Transmission_Bits` son falsos), se generan datos aleatorios perfectos para cada codeword.
3. **Modulación de los bits:** Los bits generados o extraídos se modulan según el esquema de modulación especificado en `Parameters.pdsch.Modulation`.
4. **Creación de la grilla de recursos:** Se crea una grilla de recursos que representa la disposición de los símbolos modulados en el espacio de frecuencia y tiempo. Se insertan los símbolos PDSCH (Physical Downlink Shared Channel), así como las señales de referencia DM-RS (Demodulation Reference Signal) y PT-RS (Phase Tracking Reference Signal) en las ubicaciones correspondientes de la grilla.
5. **Modulación OFDM:** La grilla de recursos se modula utilizando el esquema OFDM (Orthogonal Frequency Division Multiplexing), generando la forma de onda de la señal transmitida.

Finalmente, la función devuelve la estructura actualizada `Parameters`, que contiene la forma de onda de la señal transmitida y otros parámetros relevantes para la transmisión, el diagrama de clases se presenta en la figura B.9.

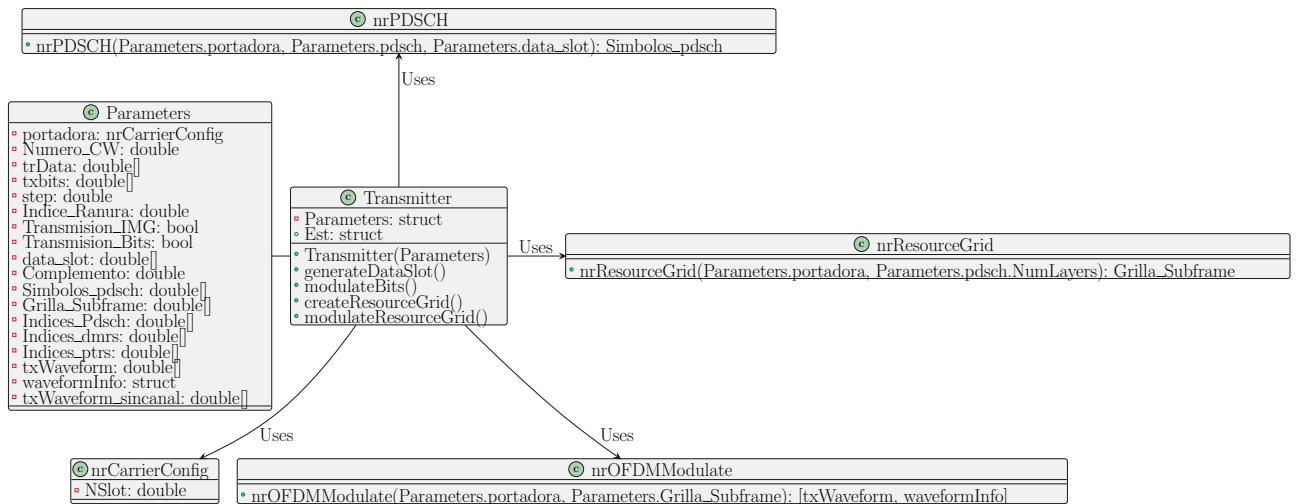


Figura B.9: Diagrama de clases Transmisor

B.2.5 Receiver

El diagrama de clases de la función de recepción se presenta en la figura B.10.

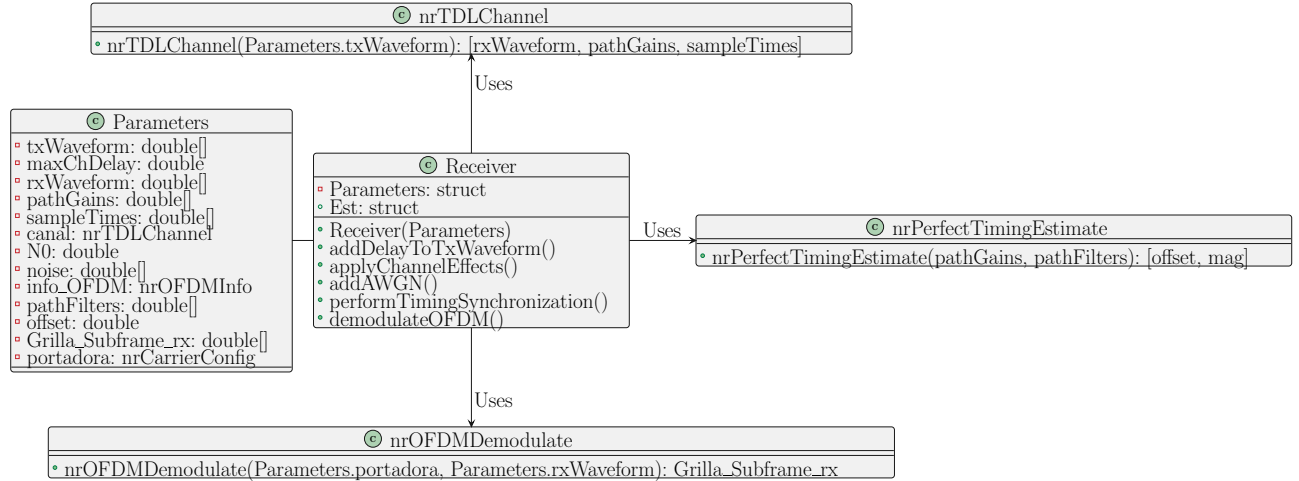


Figura B.10: Diagrama de clases Receptor

La función **Receiver**, se encarga de simular el receptor en el sistema de comunicaciones móviles 5G NR. La función toma un único parámetro de entrada **Parameters**. A continuación, se detalla el proceso realizado por la función:

1. Se simula la propagación de la señal transmitida a través del canal de comunicaciones. Para ello, se añaden retardos de canal y se calculan los caminos de propagación (*path gains*) y los tiempos de muestreo (*sample times*).
2. Se añade ruido blanco gaussiano aditivo (AWGN, por sus siglas en inglés) a la señal recibida. El nivel de ruido (N0) se calcula en función de la relación señal-ruido lineal (SNR_Lineal) y otros parámetros del sistema.
3. Se sincroniza el tiempo de la señal recibida con el tiempo de la señal transmitida para eliminar cualquier desfase causado por el canal de comunicaciones. Se utilizan filtros de trayectoria (*path filters*) y un estimador de sincronización perfecto (**nrPerfectTimingEstimate**).
4. Se procede a la demodulación OFDM (*Orthogonal Frequency Division Multiplexing*) de la señal recibida para recuperar la cuadrícula de recursos (*resource grid*) transmitida.
5. En caso de que la demodulación OFDM sea incompleta y no se obtengan todos los símbolos esperados, se completa la cuadrícula de recursos con ceros.

6. Se devuelve la estructura **Parameters** actualizada con la señal recibida demodulada y sincronizada.

B.2.6 Función Phase_Compensation

El diagrama de clases de la función de compensación de fase se presenta en la figura B.11.

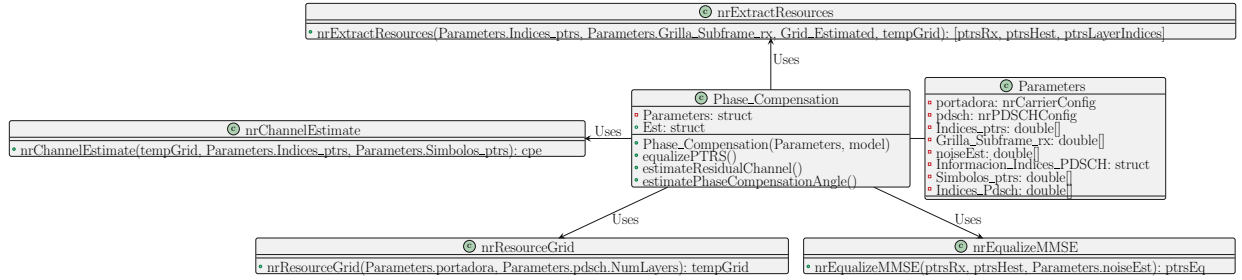


Figura B.11: Diagrama de clases función de compensación de fase

La función **Phase_Compensation**, se encarga de realizar la compensación de fase en el receptor para corregir las distorsiones introducidas por el canal de comunicaciones. La función toma dos parámetros de entrada:

- **Parameters:** Estructura que contiene todos los parámetros necesarios para la compensación de fase.
- **model:** Modelo utilizado para la estimación del canal y la ecualización.

A continuación, se describe el proceso realizado por la función:

1. Se inicializa una rejilla temporal para almacenar los símbolos ecualizados.
2. Se extraen los símbolos PT-RS (*Phase Tracking Reference Signal*) de la rejilla recibida y estimada.
3. Se ecualizan los símbolos PT-RS utilizando el algoritmo de ecualización MMSE (*Minimum Mean Squared Error*) y se asignan a la rejilla temporal.
4. Se estima el canal residual en las ubicaciones PT-RS en la rejilla temporal.
5. Se estima el ángulo de compensación de fase.

6. Se asignan los símbolos PDSCH (*Physical Downlink Shared Channel*) ecualizados a la rejilla temporal.
7. Se aplica la compensación de fase en cada símbolo OFDM (*Orthogonal Frequency Division Multiplexing*) dentro del rango de referencia PT-RS OFDM.
8. Finalmente, se actualiza la estructura `Parameters` con los símbolos PDSCH ecualizados y compensados en fase.

B.2.7 Función `Metricas_Slot`

La función `Metricas_Slot`, se encarga de calcular métricas de desempeño para cada ranura de transmisión. A continuación, se describe el proceso realizado por la función:

1. Se liberan los recursos utilizados por el objeto `Parameters.evm`.
2. Se calcula el EVM (*Error Vector Magnitude*) lineal para la ranura actual utilizando los símbolos ecualizados (`Parameters.Lineal_pdschEq`). Luego, se agrega este valor al vector `Parameters.Lineal_EVM` y se calcula la media de todos los valores para obtener `Parameters.LinealEVM`.
3. Se calcula el EVM ideal para la ranura actual utilizando los símbolos ecualizados (`Parameters.Perfect_pdschEq`). Luego, se agrega este valor al vector `Parameters.Perfect_EVM` y se calcula la media de todos los valores para obtener `Parameters.PerfectEVM`.
4. Si se habilita la estimación CNN (`Parameters.CNNEstimation`), se calcula el EVM para la ranura actual utilizando los símbolos ecualizados (`Parameters.CNN_pdschEq`). Se agregan estos valores al vector `Parameters.CNN_EVM` y se calcula la media de todos los valores para obtener `Parameters.CNNEVM`.
5. Si se habilita otra estimación CNN (`Parameters.CNNEstimation_2`), se realiza un proceso similar al punto anterior.
6. Si se habilita la estimación mediante un autoencoder (`Parameters.Autoencoder_Estimation`), se realiza un proceso similar al punto anterior.
7. Si se habilita la estimación mediante denoising (`Parameters.Denoising_Estimation`), se realiza un proceso similar al punto anterior.

8. Si se habilita otra estimación mediante denoising (`Parameters.Denoising_Estimation_resta`), se realiza un proceso similar al punto anterior.
9. Si se habilita una estimación práctica (`Parameters.EstimacionPractica`), se realiza un proceso similar al punto anterior.
10. Finalmente, se actualiza la estructura `Parameters` con los resultados de las métricas calculadas.

B.2.8 Función `Matrix_metrics`

La función *`Matrix_metrics`*, se encarga de calcular métricas agregadas a lo largo de múltiples ranuras de transmisión. La función toma un único parámetro de entrada `Parameters`. A continuación, se describe el proceso realizado por la función:

1. Se actualizan las matrices de métricas lineales, MSE (*Mean Squared Error*), EVM (*Error Vector Magnitude*) y BER (*Bit Error Rate*) con los valores correspondientes de la ranura actual.
2. Se calcula el BER para la ranura actual utilizando la función `Ber_Estimation`.
3. Si se habilita la estimación CNN (`Parameters.CNNEstimation`), se actualizan las matrices de métricas correspondientes con los valores calculados.
4. Si se habilita otra estimación CNN (`Parameters.CNNEstimation_2`), se realiza un proceso similar al punto anterior.
5. Si se habilita la estimación mediante un autoencoder (`Parameters.Autoencoder_Estimation`), se realiza un proceso similar al punto anterior.
6. Si se habilita la estimación mediante denoising (`Parameters.Denoising_Estimation`), se realiza un proceso similar al punto anterior.
7. Si se habilita otra estimación mediante denoising (`Parameters.Denoising_Estimation_resta`), se realiza un proceso similar al punto anterior.
8. Si se habilita una estimación práctica (`Parameters.EstimacionPractica`), se realiza un proceso similar al punto anterior.
9. Finalmente, se actualiza la estructura `Parameters` con las matrices de métricas calculadas.

B.2.9 Función `Perfect_Estimation`

Esta función realiza la estimación perfecta del canal utilizando el método `nrPerfectChannelEstimate` proporcionado por el paquete de comunicaciones inalámbricas de MATLAB. Después de la estimación, se calcula el MSE entre la estimación perfecta del canal y la estimación interpolada previamente realizada. Luego, se extraen los recursos PDSCH (*Physical Downlink Shared Channel*) y se ecualizan utilizando el método MMSE (*Minimum Mean Squared Error*). Finalmente, se decodifican los bits recibidos y se almacenan en la estructura de parámetros junto con otros resultados relevantes.

B.2.10 Función `Practical_Estimation`

Esta función realiza la estimación práctica del canal utilizando el método `nrChannelEstimate` proporcionado por el paquete de comunicaciones inalámbricas de MATLAB. Después de la estimación, se calcula el MSE y el MAPE (*Mean Absolute Percentage Error*) entre la estimación práctica del canal y la estimación perfecta. Luego, se realiza un proceso similar al de la función `Perfect_Estimation` para la ecualización y decodificación de los bits recibidos.

B.2.11 Funciones `CNN_Estimation` y `CNN_2_Estimation`

Estas funciones utilizan modelos de redes neuronales convolucionales (CNN) previamente entrenados para estimar el canal. Ambas funciones siguen un proceso similar: primero, se prepara la entrada para la red neuronal concatenando las partes real e imaginaria de la cuadrícula interpolada del canal. Luego, se utiliza la red neuronal para predecir la estimación del canal. Después de la predicción, se calcula el MSE y el MAPE entre la estimación de la red neuronal y la estimación perfecta. A continuación, se realizan procesos similares a los anteriores para la ecualización y decodificación de los bits recibidos.

B.2.12 Función `Autoencoder_Estimation`

Esta función utiliza un modelo de autoencoder previamente entrenado para estimar el canal. Sigue un proceso similar al de las funciones de estimación CNN, pero con un modelo de autoencoder en lugar de una CNN. Después de la predicción, se calculan el MSE y el MAPE entre la estimación

del autoencoder y la estimación perfecta. Luego, se realizan procesos similares a los anteriores para la ecualización y decodificación de los bits recibidos.

B.2.13 Función Lineal Estimation

Esta función realiza una estimación lineal del canal utilizando la cuadrícula interpolada del canal. No implica el uso de redes neuronales ni otros métodos de aprendizaje automático. Después de la ecualización y decodificación de los bits recibidos, se almacenan los resultados en la estructura de parámetros.

B.2.14 Función Denoising Estimation

Esta función realiza la estimación del canal utilizando un modelo de denoising (eliminación de ruido) mediante una red neuronal. Primero, se prepara la entrada concatenando las partes real e imaginaria de la cuadrícula interpolada del canal. Luego, se utiliza la red neuronal para predecir la estimación del canal. Después de la predicción, se calcula el MSE y el MAPE entre la estimación del canal y la estimación perfecta. A continuación, se realizan procesos similares a los anteriores para la ecualización y decodificación de los bits recibidos.

B.2.15 Función Denoising Estimation 2

Esta función también realiza la estimación del canal mediante un modelo de denoising, pero utiliza un enfoque diferente para tratar el ruido. Después de la predicción, se calcula el MSE y el MAPE entre la estimación del canal y la estimación perfecta. Luego, se realiza una operación de resta entre la estimación del canal y la cuadrícula interpolada original para tratar de eliminar el ruido. Finalmente, se realizan procesos similares a los anteriores para la ecualización y decodificación de los bits recibidos.

B.3 Procesos y servicios

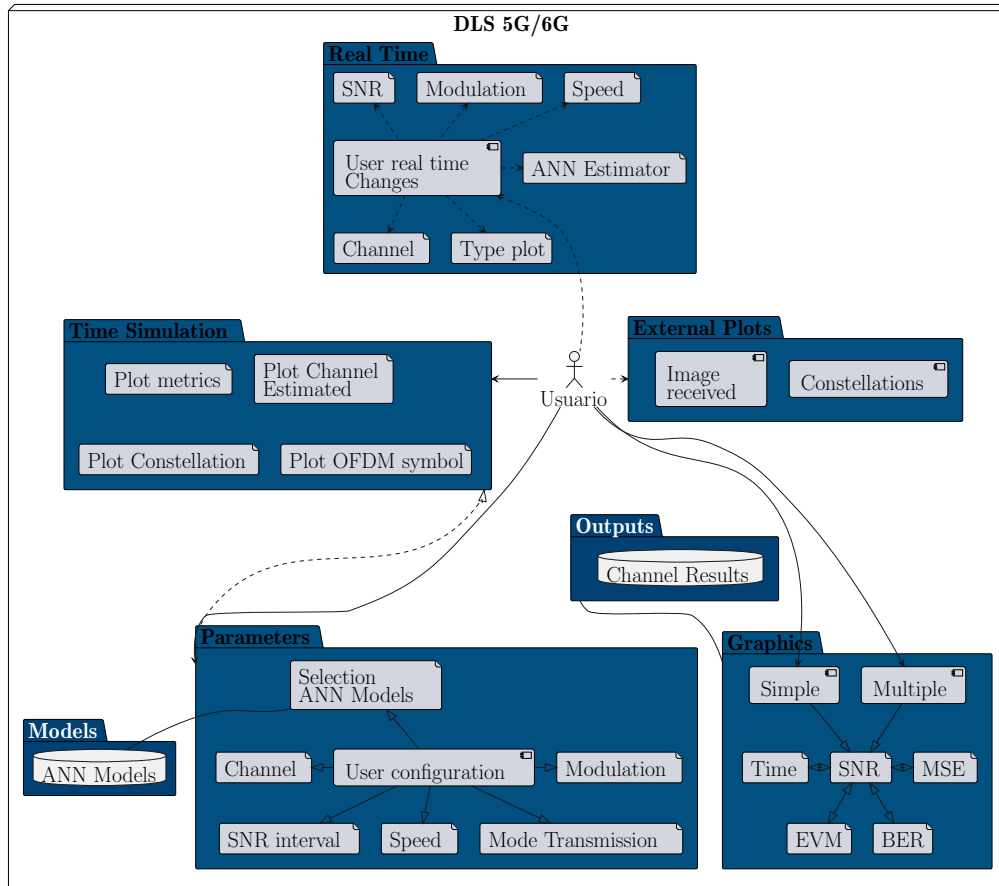


Figura B.12: Diagrama de casos de uso app DLS5G

La figura B.13 presenta el diagrama de flujo del script final así como de la figura B.12 el diagrama de casos de uso, existen 3 formas de utilizar la herramienta desarrollada, la primera es por medio de la ejecución del script final en el entorno de MATLAB, la segunda es utilizar la GUI desarrollada para la simulación, estas 2 funcionan exactamente igual, la ultima forma es por medio de la GUI en la pestaña de **Real Time** este servicio funciona por medio de un ciclo while permitiendo la interacción del usuario sobre el sistema de comunicaciones en tiempo real, para explicar de mejor manera los procesos y servicios se utilizaran los ejemplos de las secciones siguientes.

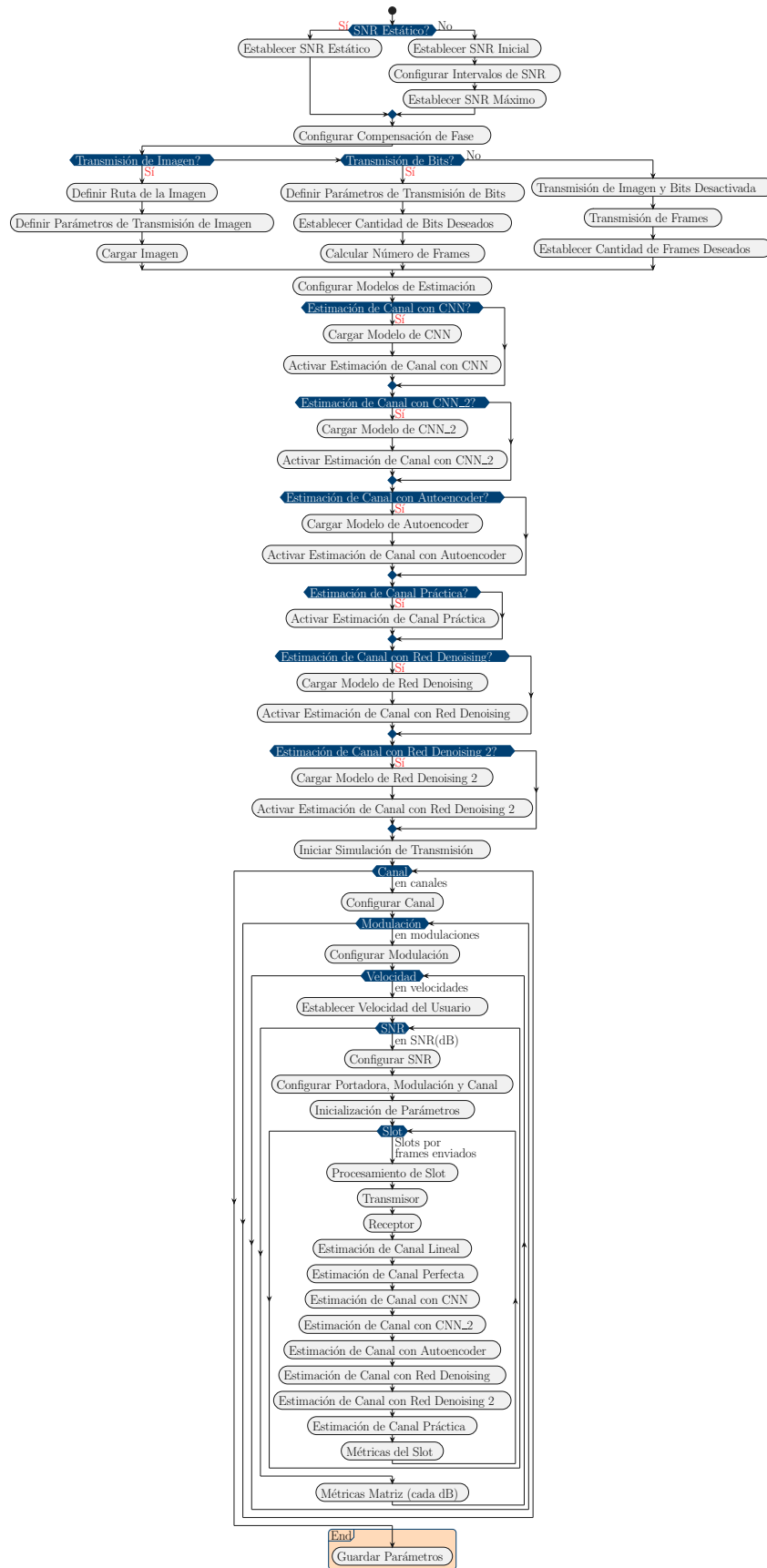


Figura B.13: Diagrama de flujo Script Simulación Final

B.4 GUI - Ejemplos de utilización

En este apartado se presentara GUI desarrollada para la simulación del entorno de comunicaciones y su implementación para la estimación de canales usando modelos de redes neuronales, existen 5 interfaces de usuario, las cuales se mostraran en los siguientes apartados, para explicar de una mejor manera los elementos dispuestos en cada interfaz, han sido listados.

B.4.1 Configuración General

En esta primer interfaz se dispone de un total de 13 elementos principales, lo cuales son :

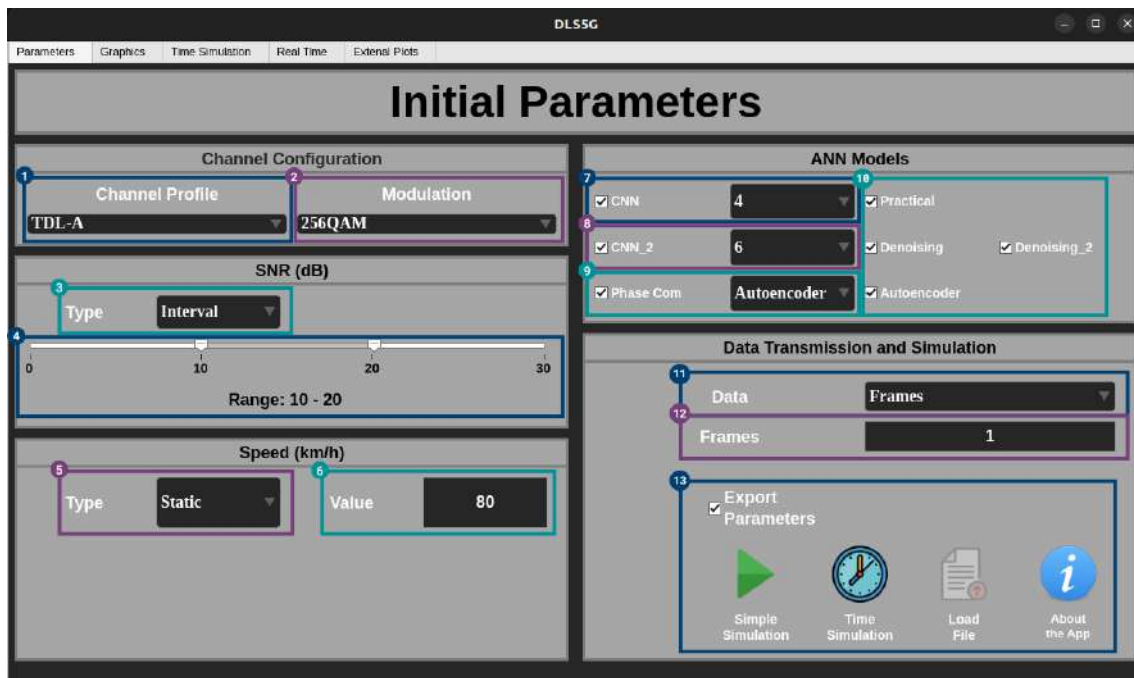


Figura B.14: Interfaz configuración de parámetros

1. Este desplegable permite la selección del tipo de canal, dentro de los disponibles se encuentran **TDL-A**, **TDL-B**, **TDL-C**, **TDL-D**, **TDL-E**.
2. Este desplegable permite la selección del tipo de modulación, dentro de los disponibles se encuentran **QPSK**, **16-QAM**, **64-QAM**, **256-QAM**.
3. Este desplegable tiene 2 posibles valores, *interval* y *static*, en caso de *interval* se activa el slider del numeral 4.

4. Esta herramienta permite seleccionar el intervalo de SNRs de simulación, este slider también se encuentra disponible para la configuración de la simulación en un rango de velocidades.
5. Este elemento cumple la misma función que el del apartado 3 pero para la velocidad.
6. En este elemento se configura la velocidad de usuario estática, un recuadro similar se encuentra disponible de igual manera para la selección de SNR.
7. Por medio de estos 2 elementos se puede configurar la red neuronal basada en un modelo CNN, seleccionada entre las 8 diseñadas.
8. Por medio de estos 2 elementos se puede configurar una segunda red neuronal en un modelo CNN, seleccionada entre las 8 diseñadas.
9. Este elemento permite la activación de la compensación de fase para el modelo seleccionado.
10. Esta zona permite seleccionar la estimación usando modelos como AUtoencoder, Denoising 1, Denoising 2 y Practico.
11. Por medio de este slider es posible configurar el tipo de recursos a transmitir, como se ha mencionado es posible enviar Frames, bits e imágenes, por lo que este elemento permite seleccionarlo, en caso de enviarse una imagen se activara el botón **Load File** disponible en el apartado 10.
12. Este espacio de texto tiene 2 funcionalidades diferentes en función del tipo de transmisión seleccionada, en caso de ser bits se utilizara como entrada para definir la cantidad de bits que se desean enviar, realizará la misma tarea en caso de utilizarse frames como medio de transmisión.
13. En esta zona encontramos 4 botones principales, el primero **Simple simulation** permite realizar una simulación con todos los elementos realizados pero sin ninguna visualización del comportamiento de los resultados, es mas rápida y eficiente para transmisiones grandes o múltiples simulaciones, el siguiente elemento **Time simulation** lleva a la siguiente interfaz y realiza la simulación simple añadiendo el apartado de visualización de métricas en tiempo real, el tercer elemento permite la carga de una imagen para realizar la transmisión en caso de ser necesario, finalmente se presenta un elemento de información sobre los desarrolladores de la aplicación.

B.4.2 Simulación Temporal

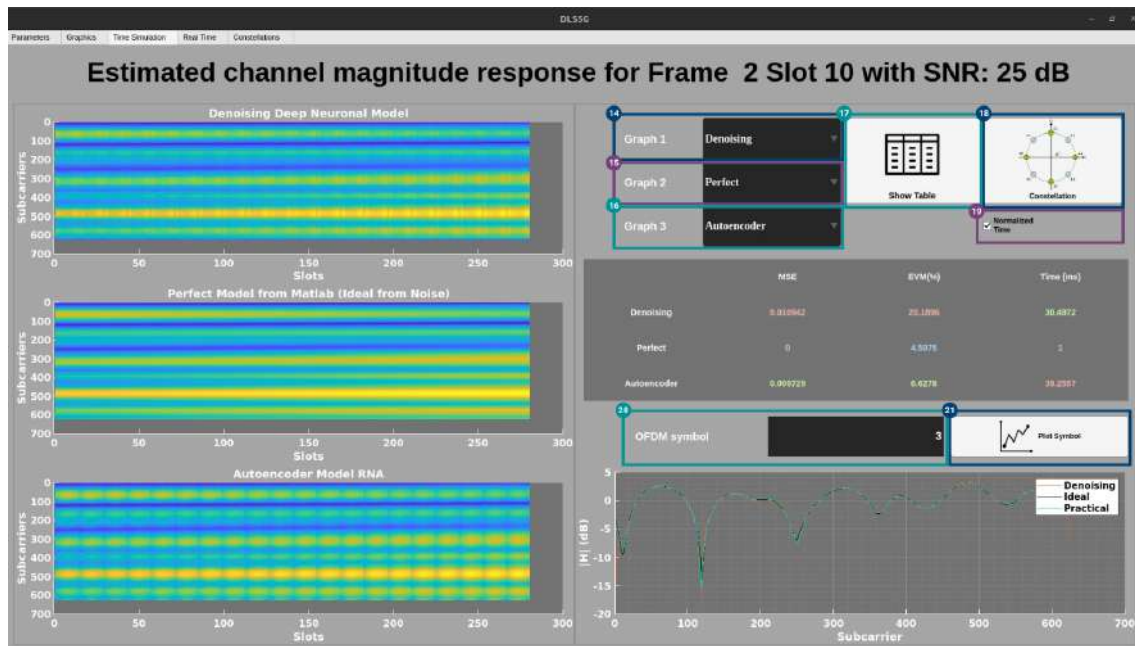


Figura B.15: Interfaz simulación con visualización de métricas

14. Este elemento selecciona los resultados que desean visualizarse en la gráfica 1, permite configurar de que modelo mostrar las métricas en tiempo real, tanto de canal estimado, MSE, EVM y Tiempo de estimación.
15. Este elemento realiza la misma función para la gráfica 2.
16. Este elemento realiza la misma función para la gráfica 3.
17. Por medio de este botón se muestran los resultados de MSE, EVM y Tiempo de estimación en el recuadro inferior, es importante resaltar que el valor que se les asigna es en función al orden comparado con los otros modelos, siendo el mejor resultado representado por el color **Azul**, seguido del color **verde** y finalmente el peor resultado se muestra con el color **rojo**.
18. Este botón genera las constelaciones actualizadas cada frame, de igual manera dirige a la interfaz presentada en la figura B.19.
19. Esta configuración permite mostrar los resultados de tiempo de estimación normalizados, los cuales funcionan como se explico al documento de esta documentación.
20. Se configura el símbolo OFDM que desea ser visualizado en tiempo real.
21. Este botón muestra la respuesta de un único símbolo OFDM.

B.4.3 Simulación en tiempo real

Esta simulación actualiza cada slot los cambios presentados en el sistema de comunicaciones por lo que se puede considerar prácticamente en tiempo real.

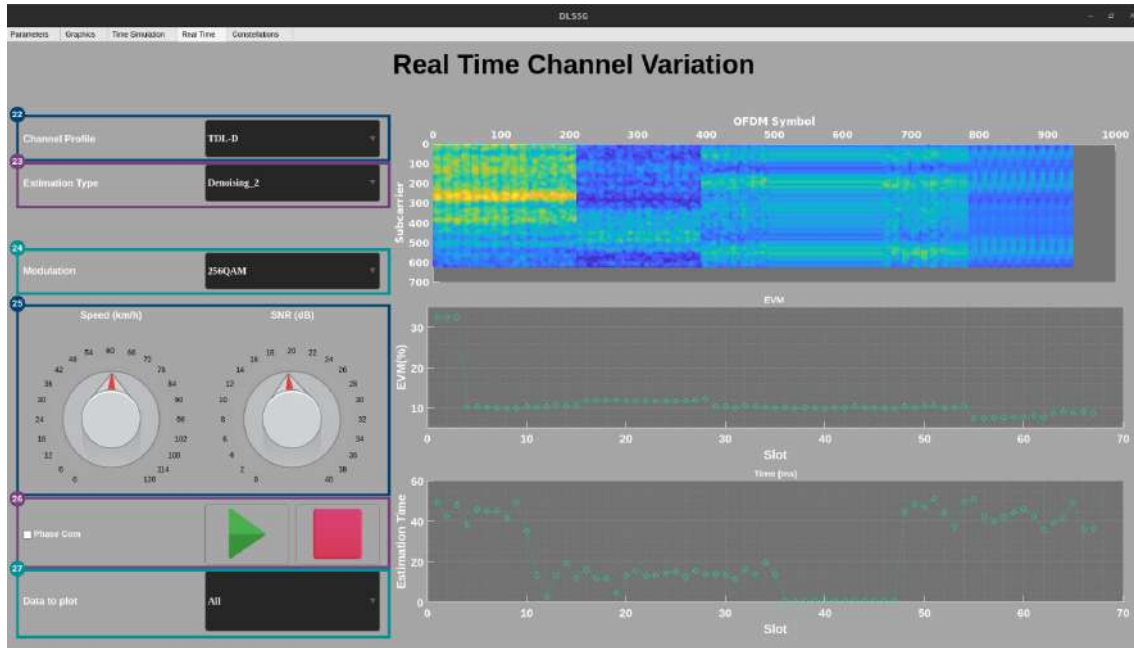


Figura B.16: Interfaz simulación en tiempo real

22. Este desplegable permite la selección del tipo de canal, dentro de los disponibles se encuentran **TDL-A, TDL-B, TDL-C, TDL-D, TDL-E**.
23. Este desplegable permite la selección del tipo de estimador de canal, se tiene un total de 7 opciones para estimar el canal : **Ideal, Lineal, Practico, Autoencoder, Denoising, Denoising 2 y CNN**, teniendo en esta ultima la opción de seleccionar alguno de los 8 modelos diseñados y almacenados en el directorio *Models*.
24. Este desplegable permite la selección del tipo de modulación, dentro de los disponibles se encuentran **QPSK, 16-QAM , 64-QAM, 256-QAM**.
25. Estos elementos permiten definir un valor de velocidad de usuario y de SNR en tiempo real.
26. Se tiene el elemento que activa la compensación de fase para la transmisión actual, y los botones para el comienzo e interrupción del programa.
27. Este ultimo desplegable permite seleccionar entre 3 elementos **All, Last 10 y Last One**, estas opciones corresponden al intervalo de muestra de datos en los plots de la derecha.

En el apartado de gráficos se tienen 2 principales elementos, gráficas múltiples y simples, los pasos para poder graficar un resultado simulado o ya existente en las bases de datos se muestran en las figuras B.17 y B.18.

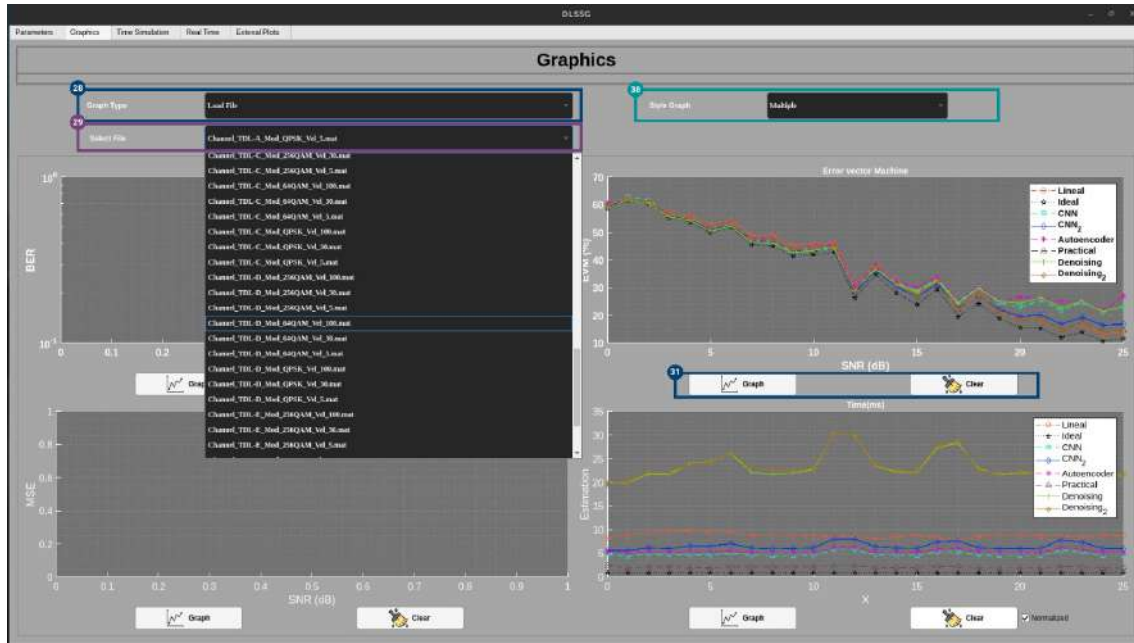


Figura B.17: Interfaz para gráficas múltiples a partir de base de datos

28. Este elemento permite seleccionar entre *Current Simulation* y *Load file*, esta ultima opción carga todos los resultados en .mat disponibles en el directorio **Outputs**, de modo tal que carga todos los archivos disponibles en el desplegable del apartado 29.
29. Este desplegable permite seleccionar el archivo que desea ser cargado para mostrar los resultados en las gráficas.
30. Este elemento permite seleccionar entre dos modos **Unique** y **Multiple**, el cual cambia presentando 4 plots para mostrar las métricas de EVM, MSE, BER, Tiempo de estimación, en caso de modo único, se podrán seleccionar las gráficas para poder verlas mas grande así como otras gráficas personalizadas.
31. Este elemento disponible para cada plot gráfica los datos cargados o permite la limpieza del elemento, es importante mencionar que la función **hold on** se encuentra activada por lo que es posible graficar la cantidad de datos que el usuario desee.
32. Esta sección permite realizar el control para las gráfica simple.
33. Este elemento permite mostrar los resultados normalizados para el tiempo de estimación.

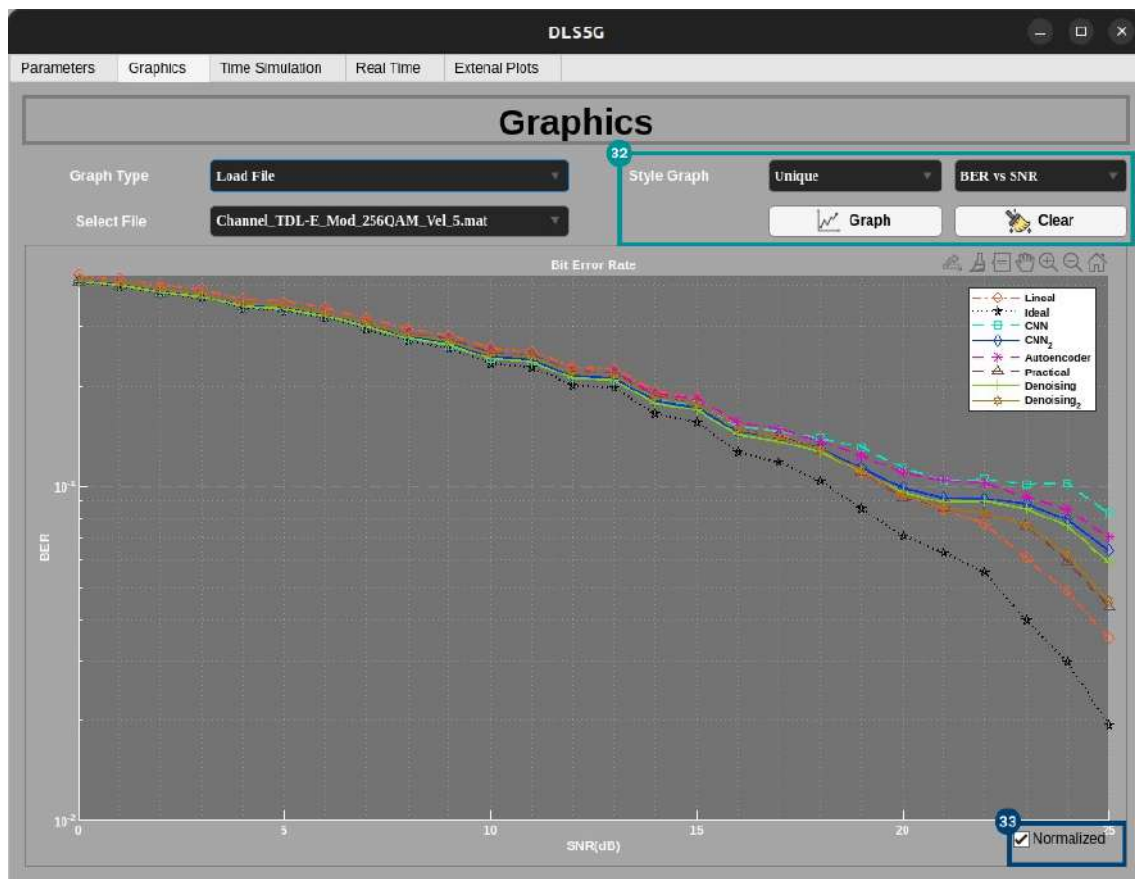


Figura B.18: Interfaz para gráfica simple a partir de base de datos

En la figura B.19 se aprecia la interfaz que permite graficar las constelaciones cada frame.

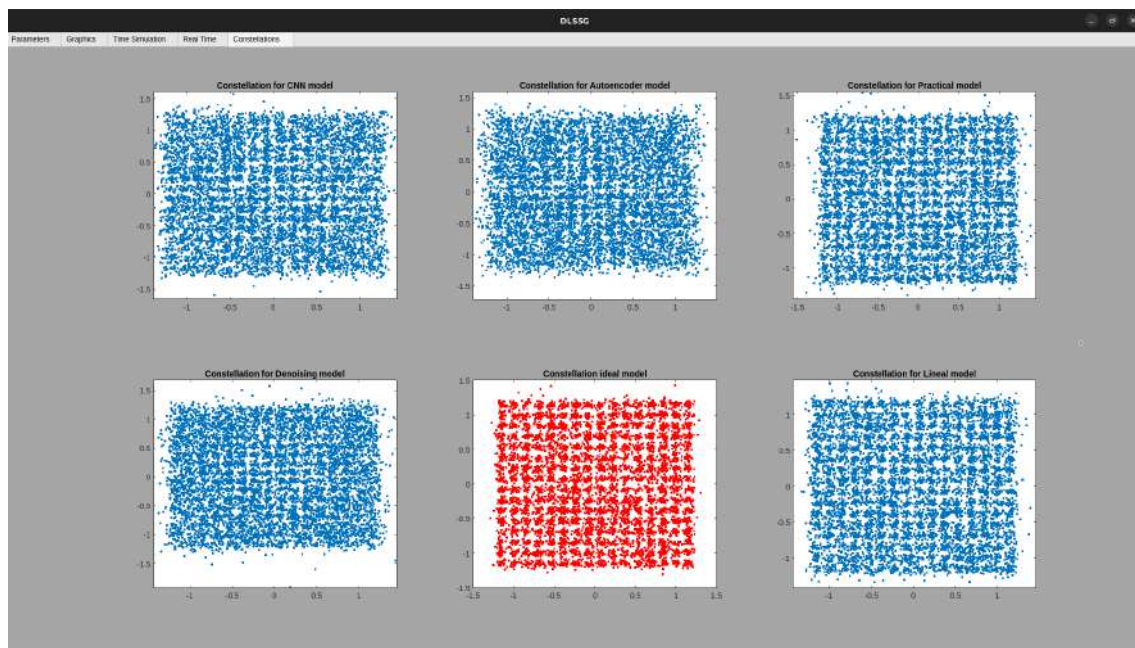


Figura B.19: Interfaz resultados constelaciones

Finalmente en la figura B.19 se aprecia la interfaz que permite graficar las imágenes transmitidas si se ha seleccionado este modo transmisión, la actualización de estas imágenes en caso de múltiples transmisiones se realiza tras cada cambio de SNR.

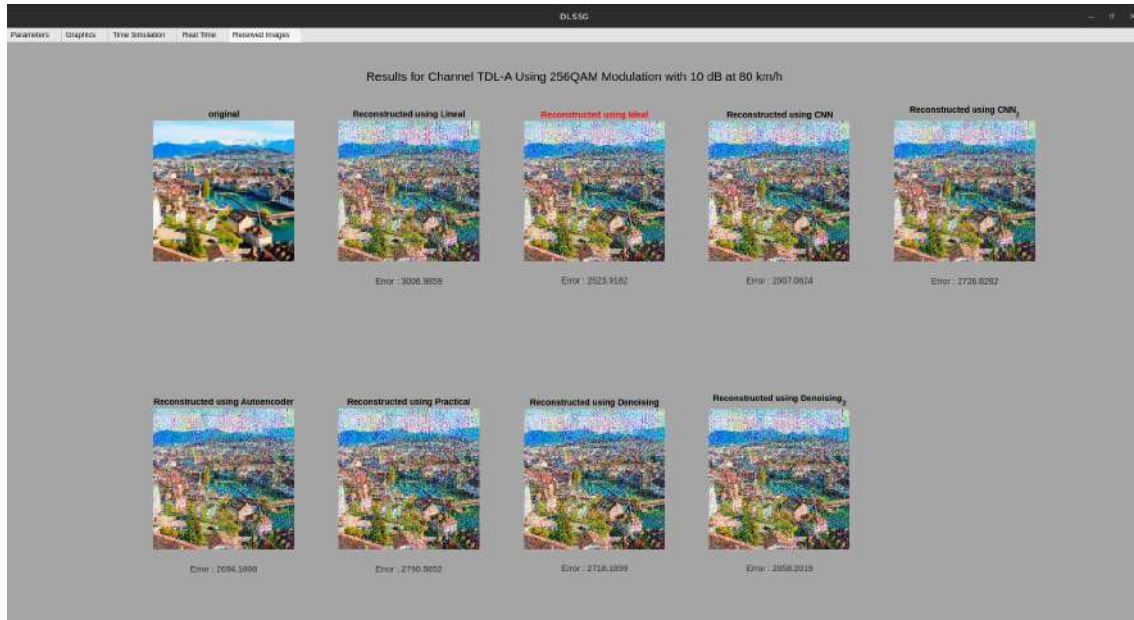


Figura B.20: Interfaz resultados transmisión imágenes

Referencias

- [1] U. Cisco, “Cisco annual internet report (2018–2023) white paper”, *Cisco: San Jose, CA, USA*, vol. 10, n° 1, págs. 1-35, 2020. dirección: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>.
- [2] I. Fogg, “Benchmarking the global 5G user experience–October update. Opensignal”, *Retrieved September*, vol. 11, pág. 2022, 2020. dirección: <https://www.opensignal.com/2020/10/13/benchmarking-the-global-5g-user-experience-october-update>.
- [3] GSMA. “Espectro 5G”. (2019), dirección: <https://www.gsma.com/spectrum/wp-content/uploads/2019/10/5G-Spectrum-Positions-SPA.pdf> (visitado 24-10-2022).
- [4] Google Trends. “Interés a lo largo del tiempo de Machine Learning y Deep Learning”. (2023), dirección: <https://trends.google.com/trends/explore?q=Machine%20Learning,%2Fm%2F0h1fn8h&date=all#TIMESERIES> (visitado 24-10-2022).
- [5] J. Jiménez Motilla y J. Ochovo Pavón, “Evaluación comparativa de redes móviles”, 2017. dirección: https://oa.upm.es/49728/1/PFC_JESUS_JIMENEZ_MOTILLA_JAIME_OCHOVO_PAVON.pdf.
- [6] J. R. B. Gomez, “EVOLUCION DE LAS REDES MÓVILES hasta hoy en día y el impacto de la red móvil de quinta generación”, *Revista Digital de Tecnologías Informáticas y Sistemas*, vol. 3, n° 1, 2019. dirección: <https://www.redtis.org/index.php/Redtis/article/view/36/49>.
- [7] C. García Pérez, “Implementación de herramientas para optimización de canal en comunicaciones 5G”, 2021. dirección: https://oa.upm.es/69446/1/TFG_CARLOS_GARCIA_PEREZ.pdf.
- [8] A. Melgar González et al., “Estimación de canal mediante deep learning en sistemas 5g con mimo masivo”, 2019. dirección: <https://burjcdigital.urjc.es/handle/10115/16548>.
- [9] D. Brake, “5G Technologies and next generation wireless: Implications for policy and competition”, *Information Technology & Innovation Foundation*, págs. 1-22, 2016. dirección: <https://www2.itif.org/2016-5g-next-generation.pdf>.

-
- [10] NGP Capital. “The Truths and Myths of 5G Deployment A Technical Perspective”. Published on October 18, 2019. (2019), dirección: <https://www.ngpcap.com/insights/5g-impact-1>.
- [11] D. S. Ramírez Alfonso y D. R. Colmenares Rodríguez, “Qué es la Tecnología 5G, Implementación de la Red en Colombia y Cambiará Nuestras Vidas”, 2021. dirección: <https://repository.ucc.edu.co/server/api/core/bitstreams/0d07e16f-fb5d-4b27-bf18-00a0f0485545/content>.
- [12] D. H. Morais, *5G and beyond wireless transport technologies: Enabling backhaul, midhaul and fronthaul*. Springer, 2021. dirección: <https://link.springer.com/book/10.1007/978-3-030-74080-1>.
- [13] W. LEI., A. Soong, L. Jianghua et al., *5G system design: an end-to-end perspective*. Springer, 2021. dirección: <https://link.springer.com/book/10.1007/978-3-030-73703-0>.
- [14] J. F. de Valgas, D. Martín-Sacristán y J. Monserrat, “5G New Radio Numerologies and their Impact on V2X Communications”, *Waves, Univesitat Politecnica de Valencia*, págs. 15-22, 2018. dirección: <https://www.iteam.upv.es/waves/2018/02-waves2018.pdf>.
- [15] F. Parca Acevedo, “Análisis para determinar el rendimiento de las formas de onda ortogonal y no ortogonal en redes inalámbricas 5G y posteriores”, 2022. dirección: <https://repository.udistrital.edu.co/bitstream/handle/11349/29416/ParcaAcevedoFelix2022.pdf?sequence=1&isAllowed=y>.
- [16] J. C. Castro Delgado, “Análisis del impacto del intervalo de tiempo de transmisión sobre la latencia en la red de acceso radio en sistemas 5G”, Grupo de Radio e Inalámbrica (GRIAL), 2019. dirección: <http://repositorio.unicauca.edu.co:8080/bitstream/handle/123456789/5618/An%C3%A1lisis%20del%20impacto%20del%20intervalo%20de%20tiempo%20de%20transmisi%C3%B3n%20sobre%20la%20latencia%20en%20la%20red%20de%20acceso%20radio%20de%20sistemas%205G.pdf?sequence=1&isAllowed=y>.
- [17] 3GPP, “5G NR: Physical channels and modulation”, en *TS 38.211 version 16.2.0 Release 16*, 2020. dirección: https://www.etsi.org/deliver/etsi_ts/138200_138299/138211/16.02.00_60/ts_138211v160200p.pdf.
- [18] A. Carreño Mendoza, “Análisis del rendimiento de los canales de la capa física PDSCH y PUSCH mediante el toolbox 5g de MaATLAB.”, B.S. thesis, Escuela Politécnica Nacional, 2021. dirección: <https://bibdigital.epn.edu.ec/bitstream/15000/21760/1/CD%2011243.pdf>.

-
- [19] M. Firdaus e Y. Moegiharto, “Performance of OFDM System against Different Cyclic Prefix Lengths on Multipath Fading Channels”, *Instituto Politécnico de Surabaya, Indonesia*, 2022. dirección: <https://arxiv.org/ftp/arxiv/papers/2207/2207.13045.pdf>.
- [20] F. Rey Micolau y J. Villares Piera, *Canal de comunicaciones*. 2018. dirección: https://openaccess.uoc.edu/bitstream/10609/77305/5/Sistemas%20de%20comunicaci%C3%B3n%20II_M%C3%B3dulo%202_Canal%20de%20comunicaciones.pdf.
- [21] C. Pérez Vega, J. M. Zamanillo Sainz de la Vega y A. Casanueva López, “Sistemas de Telecomunicación”, en Textos Universitarios, 2007, cap. 8. dirección: https://personales.unican.es/perezvr/pdf/ch8st_web.pdf.
- [22] C. Torres Zambrano y C. I. Paéz Rueda, “Análisis de un sistema de comunicaciones afectado por los desvanecimientos plano y lento tipo Rayleigh”, págs. 33-55, 2008. dirección: <http://www.scielo.org.co/pdf/inun/v12n1/v12n1a02.pdf>.
- [23] D. Tse y P. Viswanath, “Fundamentals of Wireless Communication”, en Cambridge University Press, 2004, cap. 2. dirección: <https://web.stanford.edu/~dntse/papers/book121004.pdf>.
- [24] J. A. Guajardo Ibarra, “Modelado del Desvanecimiento Espacial de Pequeña Escala en Entornos Interiores a 28 [GHz]”, 2019. dirección: http://opac.pucv.cl/pucv_txt/txt-9000/UCC9382_01.pdf.
- [25] 3GPP, “Study on channel model for frequencies from 0.5 to 100 GHz”, en *TR 38.901 version 16.1.0 Release 16*, 2020. dirección: https://www.etsi.org/deliver/etsi_tr/138900_138999/138901/16.01.00_60/tr_138901v160100p.pdf.
- [26] G. Monroy Rodríguez, “Uso del efecto Doppler para detección de obstáculos en desplazamiento peatonal”, 2016. dirección: <https://www.cs.cinvestav.mx/TesisGraduados/2016/TesisGuillermoMonroy.pdf>.
- [27] J. O. Smith, *Physical Audio Signal Processing*. DSP Related, 2010. dirección: https://www.dsprelated.com/freebooks/pasp/Tapped_Delay_Line_TDL.html (visitado 01-02-2024).
- [28] C. Prieto del Amo, “Estimación de canal y desplazamiento de frecuencia en sistemas MIMO-OFDM con prefijo cíclico insuficiente”, Tesis doct., Universidad Carlos III de Madrid-Madrid, España, 2015. dirección: <https://core.ac.uk/download/pdf/44310474.pdf>.
- [29] Y. Shen y E. Martinez, “Channel estimation in OFDM systems”, *Freescale semiconductor application note*, págs. 1-15, 2006. dirección: <https://www.nxp.com/docs/en/application-note/AN3059.pdf>.

-
- [30] F. Rey Micolau y J. Villares Piera, *Ecualización de canal*. 2018. dirección: https://openaccess.uoc.edu/bitstream/10609/77305/4/Sistemas%20de%20comunicaci%C3%B3n%20II_M%C3%B3dulo%203_Ecualizaci%C3%B3n%20de%20canal.pdf.
- [31] G. D. Forney, "The Viterbi Algorithm", *Proceedings of the IEEE*, vol. 61, págs. 268-278, 1973. dirección: https://www.isye.gatech.edu/~yxie77/ece587/viterbi_algorithm.pdf.
- [32] P. Denisowski, "Understanding Phase Noise Fundamentals", *Rohde & Schwarz*, 2021. dirección: https://www.allaboutcircuits.com/uploads/articles/RnS-Understanding-phase-noise-fundamentals_wp.pdf.
- [33] S. Gu, H. Long y Q. Li, "Phase Noise Estimation and Compensation Algorithms for 5G Systems", *European Union Digital Library*, 2019. dirección: https://eudl.eu/pdf/10.1007/978-3-030-06161-6_54.
- [34] A. F. Escallón Portilla, V. H. Ruiz Guachetá y J. G. López Perafán, "Evaluación del desempeño físico de un sistema FTTH-GPON para servicios Quad Play después de la incorporación de un módulo RoF", *TecnoLógicas*, vol. 23, n° 47, págs. 23-61, 2020. DOI: <https://doi.org/10.22430/22565337.1391>.
- [35] J. A. Santos Bustos, *Radio cognitiva basada en SDN y SDR para el control de parámetros radioeléctricos en 5G*. 2021. dirección: <https://repository.udistrital.edu.co/handle/11349/26697>.
- [36] 3GPP, "User Equipment (UE) conformance specification; Radio transmission and reception; Part 1: Range 1 Standalone", en *3GPP TS 38.521-1 version 15.1.0 Release 15*, 2019. dirección: https://www.etsi.org/deliver/etsi_ts/138500_138599/13852101/15.01.00_60/ts_13852101v150100p.pdf.
- [37] V. Meghdadi, "BER calculation", 2008, Referenciado del libro *Wireless Communications* de Andreas Goldsmith. dirección: https://www.unilim.fr/pages_perso/vahid/notes/ber_awgn.pdf.
- [38] A. Centeno Franco, "Deep learning", 2019. dirección: <https://idus.us.es/bitstream/handle/11441/90004/Centeno%20Franco%20Alba%20TFG.pdf>.
- [39] Management Solutions. "Algoritmos de Machine Learning", Universidad Politécnica de Madrid, España. (2021), dirección: <https://blogs.upm.es/catedra-idanae/wp-content/uploads/sites/698/2021/04/Idanae-1T21.pdf>.
- [40] IBM. "¿Qué es un árbol de decisión?" (), dirección: <https://www.ibm.com/es-es/topics/decision-trees>.

-
- [41] P. Divya y V. Aiswarya, “Deep Learning: Techniques and Applications”, vol. 8, págs. 125-129, 2023. dirección: <https://www.jetir.org/papers/JETIR2107018.pdf>.
- [42] J. Cano Ávalos, “Introducción al Deep Learning. Aplicación en R”, 2022. dirección: <https://idus.us.es/bitstream/handle/11441/142917/GE%20CANO%20AVALOS%2C%20JAVIER.pdf?sequence=1&isAllowed=y>.
- [43] I. H. Sarker, “Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions”, *SN Computer Science*, vol. 2, n° 6, pág. 420, 2021. dirección: <https://link.springer.com/article/10.1007/s42979-021-00815-1#citeas>.
- [44] G. E. Hinton, “Redes neuronales que aprenden de la experiencia”, *Investigación y ciencia*, n° 194, págs. 104-112, 1992. dirección: <https://dialnet.unirioja.es/servlet/articulo?codigo=223348>.
- [45] T. Kohonen, *Self-organization and associative memory*. Springer Science & Business Media, 2012, vol. 8. dirección: <https://link.springer.com/book/10.1007/978-3-642-88163-3>.
- [46] M. Á. Lafarga Coscojuela, *Biología celular de la Neurona y de la Sinápsis*. Universidad de Cantabria, 1994. dirección: <https://redined.educacion.gob.es/xmlui/handle/11162/11041>.
- [47] S. Bhuyan y S. Mukherjee. “Parts of a Neuron and Their Function”. (2023), dirección: <https://www.sciencefacts.net/parts-of-a-neuron.html>.
- [48] J. Torres. “Deep Learning: Introducción práctica con Keras (Primera Parte)”. (2018), dirección: <https://torres.ai/deep-learning-inteligencia-artificial-keras/>.
- [49] D. J. Matich, “Redes Neuronales: Conceptos básicos y aplicaciones”, *Universidad Tecnológica Nacional, México*, vol. 41, págs. 12-16, 2001. dirección: https://www.frro.utm.edu.ar/repositorio/catedras/quimica/5_anio/orientador1/monograias/matich-redesneuronales.pdf.
- [50] R. Flórez López y J. M. Fernández Fernández, *Las redes neuronales artificiales: fundamentos teóricos y aplicaciones prácticas*. Oleiros (La Coruña), España: Netbiblo, 2008. dirección: <https://investiga.upo.es/documentos/5eb288a529995203e240a5b6>.
- [51] M. C. Dickson, A. S. Bosman y K. M. Malan, “Hybridised Loss Functions for Improved Neural Network Generalisation”, en *Pan-African Artificial Intelligence and Smart Systems*, T. M. N. Ngatched e I. Woungang, eds., Cham: Springer International Publishing, 2022, págs. 169-181. DOI: https://doi.org/10.1007/978-3-030-93314-2_11.

-
- [52] O. A. Montesinos López, A. Montesinos López y J. Crossa, *Multivariate Statistical Machine Learning Methods for Genomic Prediction*. Springer Charm, 2022. DOI: <https://doi.org/10.1007/978-3-030-89010-0>.
- [53] D. Choi, C. J. Shallue, Z. Nado, J. Lee, C. J. Maddison y G. E. Dahl, *On Empirical Comparisons of Optimizers for Deep Learning*, 2020. arXiv: [1910.05446 \[cs.LG\]](https://arxiv.org/pdf/1910.05446.pdf). dirección: <https://arxiv.org/pdf/1910.05446.pdf>.
- [54] S. Ruder, *An overview of gradient descent optimization algorithms*, 2017. arXiv: [1609.04747 \[cs.LG\]](https://arxiv.org/pdf/1609.04747.pdf).
- [55] Medium. “7. Optimization Techniques on Gradient Descent and Learning Rate”. (2020), dirección: <https://medium.com/jun94-devpblog/dl-7-optimization-techniques-on-gradient-descent-and-learning-rate-14b011baa763> (visitado 06-02-2024).
- [56] D. P. Kingma y J. L. Ba, “ADAM: A Method for Stochastic Optimization”, 2015. dirección: <https://arxiv.org/pdf/1412.6980.pdf>.
- [57] R. García, “El perceptrón: una red neuronal artificial para clasificar datos”, *Revista de Investigación en Modelos Matemáticos aplicados a la Gestión y la Economía*, vol. 1, págs. 1-14, 2021. dirección: <https://www.economicas.uba.ar/wp-content/uploads/2016/04/Garcia-Roberto-1.pdf>.
- [58] A. J. Serrano, E. Soria y J. D. Martín, “Redes Neuronales Artificiales”, en Universidad de Valencia, 2009, cap. 3. dirección: http://ocw.uv.es/ingenieria-y-arquitectura/1-2/libro_ocw_libro_de_redes.pdf.
- [59] I. Goodfellow, Y. Bengio y A. Courville, *Deep learning*. MIT press, 2016. dirección: <http://www.deeplearningbook.org>.
- [60] Mathworks. “Redes neuronales convolucionales”. (), dirección: <https://la.mathworks.com/discovery/convolutional-neural-network-matlab.html>.
- [61] C. Quintero, F. Merchán, A. Cornejo y J. S. Galán, “Uso de redes neuronales convolucionales para el reconocimiento automático de imágenes de macroinvertebrados para el biomonitorio participativo”, *KnE Engineering*, págs. 585-596, 2018. DOI: <https://doi.org/10.18502/keg.v3i1.1462>.
- [62] Universidad de Stanford. “CS231n: Deep Learning for Computer Vision”. (), dirección: <https://cs231n.github.io/convolutional-networks/#add>.

-
- [63] A. Krizhevsky, I. Sutskever y G. E. Hinton, “Imagenet classification with deep convolutional neural networks”, *Advances in neural information processing systems*, vol. 60, n° 6, págs. 84-90, 2017. dirección: <https://dl.acm.org/doi/10.1145/3065386>.
- [64] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever y R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting”, *The Journal of Machine Learning Research*, vol. 45, n° 1, págs. 1929-1958, 2014. dirección: <https://jmlr.org/papers/v15/srivastava14a.html>.
- [65] c. Bonilla Carrión, “Redes Convolucionales”, 2020. dirección: <https://idus.us.es/bitstream/handle/11441/115221/TFG%20DGM%E%20Bonilla%20Carri%C3%B3n%2C%20Carmelo.pdf?sequence=1&isAllowed=y>.
- [66] C. Arana, “Redes Neuronales Recurrentes: Análisis de los modelos especializados en datos secuenciales”, 2021. dirección: <https://ucema.edu.ar/publicaciones/download/documentos/797.pdf>.
- [67] J. Pérez Guerrero, “Redes Recurrentes”, 2020. dirección: <https://idus.us.es/bitstream/handle/11441/115230/TFG%20DGM%E%20P%C3%A9rez%20Guerrero%2C%20Jes%C3%BA.pdf?sequence=1&isAllowed=y>.
- [68] A. Keshavarz. “Image Denoising Using Autoencoders (Improved version)”. (2023), dirección: <https://medium.com/@a.keshavarz/image-denoising-using-autoencoders-improved-version-5f8a90019971> (visitado 04-03-2024).
- [69] K. Zhang, W. Zuo, Y. Chen, D. Meng y L. Zhang, “Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising”, *IEEE Transactions on Image Processing*, vol. 26, n° 7, págs. 3142-3155, 2017, ISSN: 1941-0042. DOI: 10.1109/tip.2017.2662206. dirección: <http://dx.doi.org/10.1109/TIP.2017.2662206>.
- [70] J. Tang, Q. Yang y Z. Zhang, “Blind Channel Estimation for MIMO Systems via Variational Inference”, en *ICC 2022 - IEEE International Conference on Communications*, 2022, págs. 1276-1281. DOI: 10.1109/ICC45855.2022.9839066.
- [71] E. Nayebi y B. D. Rao, “Semi-blind Channel Estimation for Multiuser Massive MIMO Systems”, *IEEE Transactions on Signal Processing*, vol. 66, n° 2, págs. 540-553, 2018. DOI: 10.1109/TSP.2017.2771725.
- [72] E. Reyero Ibáñez, “Estimación de canales de comunicación dispersos mediante transformadas discretas del coseno”, 2018. dirección: https://oa.upm.es/53411/1/TFG_EDUARDO_REYERO_IBANEZ.pdf.

-
- [73] S. Sud, "Channel Estimation Using a Chirp Signal and the Fractional Fourier Transform", en *2020 IEEE 3rd 5G World Forum (5GWF)*, 2020, págs. 502-507. DOI: [10.1109/5GWF49715.2020.9221028](https://doi.org/10.1109/5GWF49715.2020.9221028).
- [74] Z. Yang e Y. Liu, "KLT-Based Estimation of Equivalent Channels in 5G MIMO Systems", en *2022 IEEE 8th International Conference on Computer and Communications (ICCC)*, 2022, págs. 1-6. DOI: [10.1109/ICCC56324.2022.10065820](https://doi.org/10.1109/ICCC56324.2022.10065820).
- [75] A. K. Tippiarti y L. Anjaneyulu, "A New Channel Estimation Technique for 5G MIMO Communication Systems", en *2020 IEEE 14th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, 2020, págs. 108-111. DOI: [10.1109/ASID50160.2020.9271705](https://doi.org/10.1109/ASID50160.2020.9271705).
- [76] M. Saideh, M. Berbineau e I. Dayoub, "On the Performance of Sliding Window TD-LMMSE Channel Estimation for 5G Waveforms in High Mobility Scenario", *IEEE Transactions on Vehicular Technology*, vol. 67, n° 9, págs. 8974-8977, 2018. DOI: [10.1109/TVT.2018.2845551](https://doi.org/10.1109/TVT.2018.2845551).
- [77] Z. Albataineh, K. Hayajneh, H. Bany Salameh, C. Dang y A. Dagmseh, "Robust massive MIMO channel estimation for 5G networks using compressive sensing technique", *AEU - International Journal of Electronics and Communications*, vol. 120, págs. 153-197, 2020, ISSN: 1434-8411. DOI: <https://doi.org/10.1016/j.aeue.2020.153197>. dirección: <https://www.sciencedirect.com/science/article/pii/S1434841119329358>.
- [78] Y. Hu, J. Zhao y B. Cheng, "Channel Estimation with an Interpolation Trained Deep Neural Network", *Journal of Computer and Communications*, vol. 9, n° 10, págs. 123-131, 2021. DOI: <https://doi.org/10.4236/jcc.2021.910008>.
- [79] Q. Hu, F. Gao, H. Zhang, S. Jin y G. Y. Li, "Deep Learning for Channel Estimation: Interpretation, Performance, and Comparison", *IEEE Transactions on Wireless Communications*, vol. 20, n° 4, págs. 2398-2412, 2021. DOI: [10.1109/TWC.2020.3042074](https://doi.org/10.1109/TWC.2020.3042074).
- [80] A. V. Stenin y A. A. Kalachikov, "Numerical Evaluation of the Channel Estimation in 5G NR Based on Machine Learning", en *2022 IEEE 23rd International Conference of Young Professionals in Electron Devices and Materials (EDM)*, 2022, págs. 285-288. DOI: [10.1109/EDM55285.2022.9855055](https://doi.org/10.1109/EDM55285.2022.9855055).
- [81] M. Meenalakshmi, S. Chaturvedi y V. K. Dwivedi, "Enhancing channel estimation accuracy in polar-coded MIMOOFDM systems via CNN with 5G channel models", *AEU - International Journal of Electronics and Communications*, vol. 173, págs. 155-166, 2024, ISSN: 1434-8411. DOI: <https://doi.org/10.1016/j.aeue.2023.155016>. dirección: <https://www.sciencedirect.com/science/article/pii/S1434841123004909>.

-
- [82] F. D. Coutinho, H. S. Silva, P. Georgieva y A. S. Oliveira, “5G cascaded channel estimation using convolutional neural networks”, *Digital Signal Processing*, vol. 126, pág. 103 483, 2022, ISSN: 1051-2004. DOI: <https://doi.org/10.1016/j.dsp.2022.103483>. dirección: <https://www.sciencedirect.com/science/article/pii/S1051200422001002>.
 - [83] P. Jiang, C.-K. Wen, S. Jin y G. Y. Li, “Dual CNN-Based Channel Estimation for MIMO-OFDM Systems”, *IEEE Transactions on Communications*, vol. 69, n° 9, págs. 5859-5872, 2021. DOI: [10.1109/TCOMM.2021.3085895](https://doi.org/10.1109/TCOMM.2021.3085895).
 - [84] Y. Wang, J. Chang, Z. Lu, F. Yu, J. Wei e Y. Xu, “Channel estimation of 5G OFDM system based on ConvLSTM network”, en *2022 7th International Conference on Communication, Image and Signal Processing (CCISP)*, 2022, págs. 62-66. DOI: [10.1109/CCISP55629.2022.9974588](https://doi.org/10.1109/CCISP55629.2022.9974588).
 - [85] R. Varshney, C. Gangal, M. Sharique y M. S. Ansari, “Deep Learning based Wireless Channel Prediction: 5G Scenario”, *Procedia Computer Science*, vol. 218, págs. 2626-2635, 2023, International Conference on Machine Learning and Data Engineering, ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2023.01.236>. dirección: <https://www.sciencedirect.com/science/article/pii/S1877050923002363>.
 - [86] R. Shankar, “Bi-directional LSTM based channel estimation in 5G massive MIMO OFDM systems over TDL-C model with Rayleigh fading distribution”, *International Journal of Communication Systems*, vol. 36, n° 16, e5585, 2023. DOI: <https://doi.org/10.1002/dac.5585>.
 - [87] H. A. Le, T. Van Chien, T. H. Nguyen, H. Choo y V. D. Nguyen, “Machine learning-based 5G-and-beyond channel estimation for MIMO-OFDM communication systems”, *Sensors*, vol. 21, n° 14, pág. 4861, 2021. dirección: <http://dx.doi.org/10.3390/s21144861>.
 - [88] V. S. Usatyuk y S. I. Egorov, “2D/3D ResNet Deep Neural Network for 4G and 5G NR Wireless Channel Estimation”, en *2023 25th International Conference on Digital Signal Processing and its Applications (DSPA)*, 2023, págs. 1-4. DOI: [10.1109/DSPA57594.2023.10113403](https://doi.org/10.1109/DSPA57594.2023.10113403).
 - [89] MathWorks. “Deep Learning Data Synthesis for 5G Channel Estimation”. Consultado en Enero 10, 2024. (2020), dirección: <https://la.mathworks.com/help/5g/ug/deep-learning-data-synthesis-for-5g-channel-estimation.html>.
 - [90] MathWorks. “Documentation: Estimate Performance of Deep Learning Network”. Consultado en Enero 17, 2024. (), dirección: <https://la.mathworks.com/help/deep-learning-hdl/ug/estimate-performance-of-deep-learning-network.html>.