Universidad de Cuenca

Facultad de Ingeniería

Carrera de Ingeniería en Telecomunicaciones

Modulación adaptativa para un sistema OFDM, utilizando técnicas de Machine Learning

Trabajo de titulación previo a la obtención del título de Ingeniero en Telecomunicaciones

Autores:

July Katherine Maguana Zhindón

Ruth Alexandra Reino Criollo

Director:

Lizandro Damián Solano Quinde

ORCID:0000-0001-7427-4889

Cuenca, Ecuador

2024-08-30

Resumen

Orthogonal frequency division multiplexing (OFDM) adaptativa es un esquema de modulación que ajusta dinámicamente la modulación del sistema en función de las condiciones del canal en tiempo real, optimizando el uso del ancho de banda y manteniendo un *Bit Error Rate (BER)* óptimo. El presente trabajo de integración curricular analiza el BER de un sistema de OFDM adaptativo convencional y con técnicas de *Machine Learnig* (ML). Para implementar el OFDM adaptativo con ML se genera una base de datos de entrenamiento, mediante el análisis del rendimiento en términos del BER y del Signal to Noise Ratio (SNR) para cinco esquemas de modulación [QPS 8-PSK 16-QAM 32-QAM 64-QAM], estableciendo los límites y parámetros. Se entrenan tres modelos: árbol de decisión, *k-nearest neighbors* (técnica de clasificación) y regresión lineal robusta (técnica de regresión). Se compara el rendimiento de cada modelo, en función del BER y del *throughput*. Los resultados muestran que las técnicas de ML mejoran el rendimiento del sistema OFDM adaptativo, permitiendo una selección precisa del esquema de modulación y una adaptación eficiente a las condiciones del canal. El sistema OFDM adaptativo con ML puede cambiar de modulación, logrando un alto *throughput* y un BER menor a 0.001 con un SNR de 17 dB o más. Entre las técnicas de ML supervisado, los modelos de clasificación como árbol de decisión y *k-nearest neighbors*, superan al de regresión lineal, en precisión de predicciones, logrando un menor BER con un SNR más bajo y un *throughput* alto.

Palabras clave del autor: adaptación eficiente, modelos predictivos, optimización, árbol de decisión, regresión lineal, vecinos más cercanos

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Cuenca ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por la propiedad intelectual y los derechos de autor.

Repositorio Institucional: https://dspace.ucuenca.edu.ec/

Abstract

Adaptive Orthogonal frequency division multiplexing (OFDM) is a modulation scheme that dynamically adjusts the system's modulation based on real-time channel conditions, optimizing bandwidth usage and maintaining an optimal Bit Error Rate (BER). This integrative curricular work analyzes the BER of a conventional adaptive OFDM system and one enhanced with Machine Learning (ML) techniques. To implement adaptive OFDM with ML, a training database is generated by analyzing performance in terms of BER and Signal to Noise Ratio (SNR) for five modulation schemes [QPS, 8-PSK, 16-QAM, 32-QAM, 64-QAM], establishing the boundaries and parameters. Three models are trained: decision tree, k-nearest neighbors (classification technique), and robust linear regression (regression technique). The performance of each model is compared based on BER and throughput. The results show that ML techniques improve the performance of the adaptive OFDM system, enabling precise modulation scheme selection and efficient adaptation to channel conditions. The adaptive OFDM system with ML can switch modulation, achieving high throughput and a BER lower than 0.001 with an SNR of 17 dB or higher. Among supervised ML techniques, classification models like decision tree and k-nearest neighbors outperform linear regression in prediction accuracy, achieving a lower BER with a lower SNR and high throughput.

Autor keywords: efficient adaptation, predictive models, optimization, decision tree, linear regression, k-nearest neighbors

The content of this work corresponds to the right of expression of the authors and does not compromise the institutional thinking of the University of Cuenca, nor does it release its responsibility before third parties. The authors assume responsibility for the intellectual property and copyrights.

Institutional Repository: https://dspace.ucuenca.edu.ec/

Índice de contenido

Resumen **2**

Abstract **3**

Índice de contenido **4**

Índice de figuras **7**

Índice de tablas **10**

Agradecimientos **11**

Dedicatoria **13**

1 Introducción **16**

1.1 Identificación del Problema 16

1.2 Antecedentes 17

1.3 Justificación 19

1.4 Alcance 19

1.5 Objetivos 20

1.5.1 Objetivo general 20

1.5.2 Objetivos específicos 20

2 Marco Teórico **21**

2.1 Sistemas de Comunicación 21

2.2 Modulación OFDM 22

2.2.1 Transmisor OFDM 24

2.2.2 Receptor OFDM 25

2.3 Canal *Additive White Gaussian Noise (AWGN)* 25

2.4 *Machine Learnig* 26

2.4.1 Técnica de Regresión Lineal 28

2.4.2 Técnicas de Clasificación 29
2.4.2.1 Modelo del Árbol de Decisión 29
2.4.2.2 Modelo k-vecinos más cercanos (*K-Nearest-Neighbor - KNN*) de predicción 33

3 Revisión Literaria 37

4 Implementación de *Machine Learning* para OFDM adaptativo 42
4.1 Herramientas de simulación 42
4.2 Implementación de OFDM Adaptativo 42
4.2.1 Parámetros de simulación Wifi 43
4.2.2 Transmisión 44
4.2.2.1 Generación de datos 45
4.2.2.2 Serie a paralelo 45
4.2.2.3 Modulación Adaptativa 45
4.2.2.4 Transformada rapida de Fourier inversa (*Inverse Fast Fourier Transform (IFFT)*) 46
4.2.2.5 Paralelo a Serie 47
4.2.2.6 Prefijo Cíclico 47
4.2.3 Canal *AWGN* 47
4.2.4 Recepción 47
4.2.4.1 Eliminar Prefijo Cíclico 48
4.2.4.2 Serie a Paralelo 48
4.2.4.3 Transformada rapida de Fourier (Fast Fourier Transform (FFT)) 48
4.2.4.4 Estimación del canal 48
4.2.4.5 Demodulador 50
4.2.4.6 Paralelo a serie 50
4.2.4.7 Calculo del Ber 50
4.2.4.8 Retroalimentación (Feedback) 50
4.3 Modelos de *Machine Learning* 51
4.3.1 Proceso de entrenamiento 51
4.3.2 Selección de la Técnica de *Machine Learning* 54
4.3.2.1 Modelo de Regresión Lineal 54
4.3.2.2 Modelo de Árbol de Regresión / Clasificación 59
4.3.2.3 Modelo de KNN 66

4.3.3 Comparación de los tres modelos 71
4.4 Implementación del modelo en OFDM Adaptativo 74
4.5 Escenarios de Prueba 76

5 Resultados y Discusión 77
5.1 Evaluación del rendimiento del sistema OFDM Adaptativo con respecto a OFDM Adaptativo con ML 77
5.1.1 Comportamiento del *Throughput* y cambio de modulación 83

6 Conclusiones y recomendaciones 102
6.1 Conclusiones 102
6.2 Recomendaciones 104
6.3 Trabajos futuros 104

Referencias 106

7 Anexo 110

Índice de figuras

Figura 2.1 Elementos de un Sistema de Comunicación 21
Figura 2.2 Generación de un símbolo OFDM [1] 24
Figura 2.3 Sistema OFDM 25
Figura 2.4 Esquema del Canal *AWGN* 26
Figura 2.5 Estructura de un árbol de decisión 30
Figura 2.6 Diagrama Árbol de Decisión 32
Figura 2.7 Comportamiento del modelo de KNN para la clasificación 33
Figura 2.8 Estructura de la Matriz de Confusión 36

Figura 4.1 OFDM adaptativo convencional 43
Figura 4.2 Transmisor OFDM adaptativo 44
Figura 4.3 Comportamiento del BER vs SNR para las diferentes modulaciones . . 46
Figura 4.4 Modulador Adaptativo, selector 46
Figura 4.5 Bloques en el Receptor OFDM adaptativo 48
Figura 4.6 Modelo de Trabajo 51
Figura 4.7 Gráfica del BER vs SNR en los rangos deseados 52
Figura 4.8 Gráfica de respuesta / Modelo de regresión Lineal 56
Figura 4.9 Gráfica de las predicciones vs valores reales / Modelo de regresión Lineal 57
Figura 4.10 Gráfica de los residuos / Modelo de regresión Linea 58
Figura 4.11 Gráfica del BER vs SNR del modelo de Regresión Lineal 59
Figura 4.12 Estructura del Árbol de Clasificación 60
Figura 4.13 Gráficas de la respuesta de las predicciones reales del modelo *Fine Tree* 61
Figura 4.14 Gráficas de la respuesta de las predicciones reales del modelo *Medium Tree* 62
Figura 4.15 Gráficas de la respuesta de las predicciones reales del modelo *Coarse Tree* 62
Figura 4.16 Gráficas de las predicciones ajustadas con respecto a valores reales modelo *Fine Tree* 63

Figura 4.17 Gráficas de las predicciones ajustadas con respecto a valores reales modelo *Medium Tree* 63
Figura 4.18 Gráficas de las predicciones ajustadas con respecto a valores reales modelo *Coarse Tree* 64
Figura 4.19 Gráficas de los residuos del modelo *Fine Tree* 64
Figura 4.20 Gráficas de los residuos del modelo *Medium Tree* 65
Figura 4.21 Gráficas de los residuos del modelo *Coarse Tree* 65
Figura 4.22 BER vs SNR | Comparación de los tres modelos 66
Figura 4.23 Matriz de Validación | Modelo *Fine KNN* | Métricas TPR y FNR 67
Figura 4.24 Matriz de Validación | Modelo *Fine KNN* | Métricas PPV y FDR 68
Figura 4.25 Matriz de Validación | Modelo *Medium KNN* | Métricas TPR y FNR . . . 68
Figura 4.26 Matriz de Validación | Modelo *Medium KNN* | Métricas PPV y FDR . . . 69
Figura 4.27 Matriz de Validación | Modelo *Coarse KNN* | Métricas TPR y FNR 69
Figura 4.28 Matriz de Validación | Modelo *Coarse KNN* | Métricas PPV y FDR 70
Figura 4.29 BER vs SNR | Comparación de los tres modelos 71
Figura 4.30 BER vs SNR | Comparación de los tres modelos 73
Figura 4.31 Esquema de implementación de *Machine Learning* en el sistema de OFDM 75

Figura 5.1 Comportamiento del BER | Modelo sin ML 78
Figura 5.2 Comportamiento del BER | Modelo de *Robust Linear* 79
Figura 5.3 Comportamiento del BER | Modelo de *Fine Tree* 80
Figura 5.4 Comportamiento del BER | Modelo de *Fine KNN* 81
Figura 5.5 Comportamiento del BER de los 4 modelos 82
Figura 5.6 Comportamiento del índice de modulación | Modelo sin ML 83
Figura 5.7 Comportamiento del *Throughput* | Modelo sin ML 84
Figura 5.8 Comportamiento del índice de modulación | Modelo *Robust Linear* . . . 85
Figura 5.9 Comportamiento del *Throughput* | Modelo *Robust Linear* 85
Figura 5.10 Comportamiento del índice de modulación | Modelo *Fine Tree* 86
Figura 5.11 Comportamiento del *Throughput* | Modelo *Fine Tree* 87
Figura 5.12 Comportamiento del índice de modulación | Modelo *FINE KNN* 88
Figura 5.13 Comportamiento del *Throughput* | Modelo *Fine KNN* 88
Figura 5.14 Comportamiento del índice de modulación de los 4 modelos 89
Figura 5.15 Comportamiento del *Throughput* de los 4 modelos 90
Figura 5.16 Comportamiento del canal | Escenario 2 91

Figura 5.17 Comportamiento del BER | Modelo sin *Machine Learning* 92
Figura 5.18 Comportamiento del BER | Modelo de *Robust Linear* 93
Figura 5.19 Comportamiento del BER | Modelo de *Fine Tree* 94
Figura 5.20 Comportamiento del BER | Modelo de *Fine KNN* 95
Figura 5.21 Comportamiento del BER de los 4 modelos | [0dB - 30 dB] 96
Figura 5.22 Comportamiento del BER de los 4 modelos | [6dB - 26 dB] 97
Figura 5.23 Comportamiento del *Throughput* Promedio con respecto a SNR | Cuatro modelos | [0dB - 30 dB] 98
Figura 5.24 Comportamiento del canal | Escenario 3 99
Figura 5.25 Comportamiento del BER Promedio con respecto a SNR | Cuatro modelos100
Figura 5.26 Comportamiento del *Throughput* Promedio con respecto a SNR | Cuatro modelos 100

Índice de tablas

Tabla 4.1 Parámetros definidos 44
Tabla 4.2 Selección de modulación con respecto al SNR 51
Tabla 4.3 Rangos de SNR para obtener un BER <0.001 52
Tabla 4.4 Comportamiento del BER para las cinco diferentes modulaciones con diferente SNR 53
Tabla 4.5 Selección de modulación con respecto al SNR de subida y modulación . . 54
Tabla 4.6 Rangos de los Predictores 54
Tabla 4.7 Comparación de Métricas | Modelo de Regresión Lineal y Arból de Decisión 72
Tabla 4.8 Comparación de Métricas | de *K-nearest neighbor* 72

Tabla 5.1 Comparación del SNR enviado al canal con el SNR estimado 77

Agradecimientos.

Primero agradezco a **Dios**, por conceder la fuerza y valor de seguir adelante a pesar de todas las situaciones adversas, por guiarme en un buen camino para alcanzar mis metas y objetivos.

A mis padres, **Manuel** y **Mercedes** por brindarme su apoyo incondicional en esta parte muy importante de mi vida profesional, gracias por el esfuerzo y por creer en mi en esta trayectoria.

A mis hermanos, **Jenny**, **Darwin** y **Jhonny** y a mi sobrina **Amy**, por ofrecerme su paciencia y cariño sincero, como también, su apoyo para obtener este importante logro de vida.

A mis compañeros y profesores, que han estado a mi lado desde el inicio en las etapas difíciles y celebrar los momentos especiales. Gracias por compartir cada experiencia, por sus enseñanzas, y por hacer de este viaje algo inolvidable.

A mi compañera de tesis **Ale**, que se ha convertido en una persona importante en mi vida, por todos los momentos que pasamos juntas, gracias Ale por comprenderme y tenerme paciencia, se que ha sido un largo camino el cual hemos recorrido juntas. Se que podemos lograr muchas cosas en la vida. !Te deseo muchos éxitos en tu vida Ale¡

A mi tutor **Ing. Lizandro Solano**, por su asesoramiento y paciencia a lo largo de este proyecto. Su experiencia y orientación han sido claves para que esta tesis se hiciera realidad. Sin su guía, no habría podido lograr este objetivo.

July K. Maguana Zhindón

En primer lugar, le agradezco a **Dios** por darme la fortaleza y sabiduría para superar cada uno de los obstáculos de este largo camino, por permitirme levantarme cada día para seguir luchando por alcanzar esta meta tan grande.

A mis padres, **Ruth** y **Fernando**, quienes han sido mis pilares en este trayecto. Su amor, apoyo incondicional, sus palabras y cada una de sus pequeñas y grandes acciones me han permitido lograr este sueño. Gracias por siempre estar y jamás dejar de creer en mí y en que juntos íbamos a lograrlo.

A mis hermanos, **José** y **Christian**. José, que siempre me apoyó en todo sin importar lo que fuera, cada una de tus palabras fueron muy importantes para mí, gracias. Christian, que con sus palabras y preguntas me motivaban a seguir.

A mis tíos **Alicia** y **Jorge**, que no solo me brindaron palabras de aliento, sino que me ayudaron en todo lo que podían. Llevo siempre en mi corazón su apoyo y les estoy eternamente agradecida por todo. Y a toda mi familia tíos y primas que me han alentado siempre a seguir.

A todos mis amigos que tuve la dicha de conocer en el transcurso de mi camino por la universidad, cada una de las risas, frustraciones y lágrimas que compartimos juntos para lograrlo. Al grupo de Machetes, que se convirtió en una familia, celebrando y apoyándonos siempre para seguir y culminar cada uno de nuestros trabajos. Gracias por hacer de la universidad un lugar donde encuentras a personas que te alegran el día y te hacen valorar cada esfuerzo que tuviste que pasar.

A mi compañera de tesis, **July**, quien me hacía reaccionar cuando pensaba que no podía continuar. Cada llanto, risa y frustración que tuvimos en este trabajo valieron la pena. Mi futura colega, ¡sí pudimos, ÏNGENIERAS¡

Al tutor **Ing. Lizandro** Solano por su tiempo y orientación, que nos permitieron desarrollar y culminar con éxito este trabajo de titulación.

R. Alexandra Reino Criollo

Dedicatoria.

Dedico esta tesis a mis padres por todo su cuidado, por que se por todo lo que han pasado y el esfuerzo que dieron para que yo pueda seguir cumpliendo mis metas.

A mi sobrina Amy que desde su nacimiento me dio ese sentimiento de poder seguir y ser un ejemplo para ella en su futuro, demostrándole que todo es posible si hay esfuerzo y dedicación.

A Ricardo, por estar a mi lado cuando sentía que ya no podía más, eres una de las personas más importantes en mi vida.

July K. Maguana Zhindón

Dedico mi tesis con mucho amor a mis padres, por acompañarme y apoyarme incondicionalmente en cada momento difícil de este largo camino que llega a su fin con el logro de un gran sueño. Cada palabra, aliento, me han permitido culminar este gran camino junto a ustedes.

A mis hermanos José y Christian que siempre me han apoyado a su manera, sin importar la distancia gracias por estar en cada momento importante de este arduo trayecto. ¡¡Y si se pudo!!

A mis tíos, Alicia y Jorge que me han apoyado siempre, y eso lo llevo siempre en mi corazón.

A mis tíos y primos, y a todos aquellos que conforman mi familia, gracias por estar y siempre alentarme a seguir.

A mis bebes (mis mascotas) que estuvieron a mi lado en todo momento, en cada desvelada y madrugada esperándome.

R. Alexandra Reino Criollo

Glosario

AMC Adaptive Modulation and Coding. 42

AWGN Additive White Gaussian Noise. 4, 5, 7, 21, 25, 26, 44, 47, 50, 76

BER Bit Error Rate. 2, 3, 16, 38, 43

CP Prefijo Cíclico. 25

DL Deep Learning. 39, 41

FFT Fast Fourier Transform. 5, 17, 23, 25, 48

FN False Negatives. 35

FP False Positive. 35

IFFT Inverse Fast Fourier Transform. 5, 23, 24, 45–47

ISI Interferencia Intersimbólica. 16, 17, 22

KNN K-Nearest Neighbor. 33, 51, 103

MAE Mean Absolute Error. 29, 72

ML Machine Learning. 26, 39, 51, 76, 104

MSE Mean Squared Error. 29, 71

OFDM Orthogonal frequency division multiplexing. 2, 3, 16, 22

RMSE Root Mean Square Error. 29

SNR Signal to Noise Ratio. 2, 18, 51, 52, 77, 82, 102

TN True Negatives. 35

TP True Positive. 34, 35

VoIP Voz sobre IP. 43

1. Introducción

El sistema de modulación por división de frecuencia ortogonal (Orthogonal frequency division multiplexing (OFDM)), por su alta eficiencia espectral y su capacidad de envío de la señal por múltiples trayectorias, permite mejorar el rendimiento en los sistemas de comunicación. La modulación OFDM adaptativa tiene la capacidad de ajustar dinámicamente sus parámetros en tiempo real según las condiciones del canal de comunicación. Esta es robusta en entornos con interferencias y atenuaciones. Permite una transmisión eficiente y una alta utilización de recursos en el canal de envío de la señal. Para lo cual se han propuesto técnicas que permiten ajustar el esquema de modulación usado por OFDM acorde al estado del canal - OFDM Adaptativo. Estas técnicas pueden ser deterministas o convencionales y basadas en *Machine Learning*.

En este trabajo de integración curricular se evalúa el rendimiento de un sistema OFDM que ajusta el esquema de modulación utilizado mediante el uso de técnicas de *Machine Learning*, el cual es comparado con un esquema OFDM adaptativo convencional, es decir basado en técnicas deterministas. Para esta evaluación se consideran tres modelos supervisados de *Machine Learning*, a través de las que la modulación utilizada por OFDM se adapta a las condiciones variables del canal. Los modelos de *Machine Learning* considerados son; regresión lineal, árbol de decisión y *k-nearest neighbors*, y para cada uno de ellos se evalúa el rendimiento en términos de la Bit Error Rate (BER) y el *throughput*, para distintos escenarios de condiciones del canal.

1.1. Identificación del Problema

La técnica de modulación OFDM es ampliamente utilizada en diversos sistemas de comunicación debido a las ventajas que ofrece. Permite una utilización más eficiente de los recursos del sistema, como el ancho de banda y la potencia de transmisión, optimizando la eficiencia espectral. Mediante el uso de múltiples portadoras ortogonales, OFDM maximiza el uso del ancho de banda y reduce la Interferencia Intersimbólica (ISI). Sin embargo, OFDM utiliza el mismo esquema de modulación para todos las subportadoras. Esto puede afectar su rendimiento en términos de la tasa de error de bits (BER), especialmente cuando las condiciones del canal varían instantáneamente. Al no adaptar la modulación a las condiciones específicas, el rendimiento puede disminuir.

El efecto de utilizar el mismo esquema de modulación para todas las subportadoras en los

sistemas de comunicación actuales, dónde se requieren altas tasas de transmisión, resulta en un mayor uso de ancho de banda y en la presencia de ISI.

Estas limitaciones que presenta el esquema OFDM se deben mayormente a que sus parámetros de modulación no se ajustan dinámicamente con respecto a la variabilidad del canal en el tiempo. En situaciones donde el canal tiene múltiples trayectos y es selectivo en frecuencia, OFDM puede reducir su eficiencia, ya que el esquema de modulación utilizado por las subportadoras no sería óptimo para las condiciones del canal.

En consecuencia, es necesario cambiar a modulaciones de orden más bajo, como BPSK y QPSK, ya que tienen mayor tolerancia al ruido, para mitigar los efectos de un canal con características variantes en el tiempo.

Según lo expuesto, es necesario que el esquema de modulación utilizado por un sistema de comunicación se adapte dinámicamente a las condiciones variables del canal. De esta manera, se requiere un mecanismo que permita seleccionar la técnica de modulación más eficiente, es decir que se ajuste a las condiciones variables del canal. Este mecanismo incorporaría técnicas de inteligencia artificial, las cuales, a partir de una estimación del estado del canal, seleccionan el esquema de modulación óptimo.

1.2. Antecedentes

La técnica de modulación OFDM (Multiplexación por División de Ortogonal de Frecuencia) divide el espectro disponible en múltiples subcanales, donde cada uno transporta información utilizando un esquema de modulación específico [2]. El resultado es una señal que contiene N subcanales de transmisión dispuestos de manera ortogonal, lo que significa que no se interfieren entre sí. Esta característica reduce la interferencia entre símbolos (ISI) y mejora la eficiencia espectral, ya que utiliza eficazmente el ancho de banda disponible. Debido a su capacidad para lograr una alta eficiencia espectral, aumentar el rendimiento del sistema y mitigar la interferencia entre símbolos en el sistema de comunicación, OFDM se ha convertido en una alternativa ampliamente adoptada en los sistemas de comunicación [3]. Una característica fundamental de OFDM es su fácil implementación del proceso de modulación y demodulación mediante las operaciones de la *Fast Fourier Transform (FFT)* y su Inversa (IFFT). Sin embargo, en adición a estas propiedades, presenta limitaciones en canales selectivos en frecuencia y multi trayecto, los cuales inciden en la degradación de la relación señal-ruido del sistema (SNR) [4]. Los sistemas OFDM utilizan un mismo esquema de modulación en todos los subcanales, pero el

efecto del canal sobre cada uno de estos es diferente, afectando directamente en la tasa de error de bits (BER). Por este motivo, se han considerado, como solución, implementaciones de OFDM adaptativas, en las cuales se emplean diferentes esquemas de modulación para los subcanales, según el nivel del Signal to Noise Ratio (SNR) [5]. Czylwik [6] propone la utilización de diferentes esquemas de modulación para las subportadoras individuales, distribuyendo la potencia de la señal de manera óptima entre las subportadoras para maximizar la eficiencia espectral y minimizar el BER. Para esto, el SNR medido en cada subcanal es utilizado para determinar el esquema de modulación y la potencia de la señal óptima para el subcanal. De esta manera, cuando el SNR se degrada, se selecciona un esquema de modulación basado en el SNR que garantice el rendimiento del BER deseable. Esta técnica de OFDM adaptativo permite reducir la tasa de error de bits en condiciones del canal dónde el SNR es bajo [7]. OFDM adaptativo resulta de utilidad en comunicaciones donde el medio de transmisión es variable por lo que el rendimiento de una implementación OFDM convencional, no adaptativa, se ve afectada, como en el caso de comunicaciones a través de línea eléctrica [8], canales acústicos subacuáticos [9], entre otros.

Sin embargo, en adición a los beneficios que ofrece OFDM adaptativa, también presenta limitaciones, tales como su complejidad, lo cual se traduce en un mayor requerimiento de capacidad computacional, y su incapacidad de ajustarse a canales que cambian rápidamente en el tiempo, entre otros [10][11][12].

En la actualidad la tendencia en la investigación de esquemas de modulación adaptativa se ha dirigido al uso de técnicas de inteligencia artificial (IA) [10][13]. Estas técnicas de IA son un conjunto de enfoques y algoritmos que permiten mediante el aprendizaje de los datos, identificar patrones, tomar decisiones o realizar predicciones. La técnica de aprendizaje automático es ampliamente utilizada en la implementación de sistemas de comunicación basados en OFDM adaptativa, mediante el desarrollo de algoritmos para la selección de un esquema de modulación óptimo. Bin Ha et al. [13] proponen un modelo de aprendizaje previo a la comunicación real, para determinar de manera adaptativa el modo de transmisión óptimo según la condición del canal, empleando técnicas de *Deep Learning (DL)*, manejan una estructura de capas denominada red neuronal artificial (ANN). Si bien estos algoritmos de DL son eficientes, suelen ser computacionalmente exigentes y requieren una gran cantidad de muestras de entrenamiento, razón por la que el uso de *Machine Learning (ML)* ha representado una importante solución a este problema de recursos, debido a que los algoritmos de ML requieren menor capacidad computacional que los hacen adecuados para implementar esquemas de procesamiento de señales rápidos, que pueden cumplir con los crecientes requisitos de tráfico de datos [14]. La

aplicación de *Machine Learning* en esquemas de modulación OFDM adaptativas utilizan técnicas de aprendizaje supervisado, no supervisado, o esquemas híbridos, los cuales combinan las dos anteriores [13].

1.3. Justificación

OFDM es un esquema de modulación utilizado en sistemas de comunicación que permite mejorar la eficiencia espectral, capacidad del sistema, mitigación de la interferencia inter simbólica. Sin embargo, a pesar de sus capacidades, su utilización se ve limitado a canales relativamente estáticos. Por lo que es importante notar que las condiciones del canal de comunicaciones varían entre diferentes localizaciones geográficas y con el tiempo. De esta manera la implementación de un esquema de comunicaciones debe considerar estas variaciones en las características espectrales y temporales del medio de transmisión. Una implementación de un esquema de modulación estático resultaría, no solamente, en un uso ineficiente de los recursos de comunicaciones, sino en un incremento en el BER, afectando directamente a la calidad del sistema de transmisión [7]. Por otro lado, una implementación de un esquema de comunicaciones dinámico o adaptativo solventa las limitaciones impuestas por un esquema estático [5]. En este contexto, es necesario que OFDM pueda seleccionar el esquema de modulación de manera adaptativa. Un esquema de modulación adaptativo OFDM, permitiría transmitir datos de una manera más eficiente, al usar adecuadamente los recursos de ancho de banda, teniendo siempre presente las condiciones del canal. Esto requiere un análisis continuo de la calidad de la recepción de datos, con el objetivo de identificar cuando las condiciones del canal ya no son adecuadas para la modulación usada, y de esta manera seleccionar un esquema de modulación óptimo con respecto a las nuevas condiciones del canal, el cual permite hacer un uso más eficiente de los recursos utilizados para la comunicación y reducir la tasa de errores (BER) con base a la relación señal-ruido (SNR).

1.4. Alcance

En el presente trabajo de integración curricular se analiza el rendimiento de un sistema de comunicación basado en una implementación de OFDM adaptativo. Para esto se emplean técnicas de *Machine Learning*, que permiten ajustar dinámicamente el esquema de modulación utilizado por OFDM, de tal forma que responda a las variaciones en el estado del canal. Debido a que se requiere predecir la próxima modulación en base a los parámetros del sistema, se han

considerado las técnicas: i) *linear regression*, ii) *decision tree* y iii) *k-nearest neighbors*. Que ajustan los datos a una función que predice el tipo de modulación o los clasifican basándose en las características de los parámetros.

Para el análisis del rendimiento del esquema de modulación implementado, se obtienen métricas, para diferentes condiciones preestablecidas del canal de comunicaciones. Estas métricas son analizadas y contrastadas entre sí, así también con las correspondientes a un sistema de comunicaciones basado en OFDM que implementa técnicas deterministas o convencionales para ajustar la modulación acorde a las condiciones del canal.

Las métricas utilizadas para el análisis y comparación de rendimiento son la tasa de error de bit (BER) y el *throughput* o tasa de transmisión.

1.5. Objetivos

1.5.1. Objetivo general

Evaluar el rendimiento de un sistema de comunicación basado en OFDM adaptativo implementado a través de técnicas de *Machine Learning*.

1.5.2. Objetivos específicos

- Evaluar el rendimiento de OFDM adaptativo convencional en diferentes condiciones preestablecidas del canal.
- Implementar un método para la selección de la modulación adaptado a las condiciones del canal, aplicando técnicas de *Machine Learning*.
- Entrenar el esquema de modulación implementado para OFDM adaptativa, en diferentes condiciones preestablecidas del canal.
- Comparar el esquema de modulación OFDM adaptativo usando *Machine Learning*, con el esquema OFDM adaptativo convencional.

2. Marco Teórico

En este capítulo se introducen los conceptos teóricos fundamentales, necesarios en el desarrollo del presente trabajo de integración curricular.

2.1. Sistemas de Comunicación

Los sistemas de comunicación están presentes en casi todas las actividades de la vida cotidiana, debido a la necesidad de la comunicación constante. Existen múltiples tipos de sistema de comunicación, como los sistemas de comunicación móvil, redes de datos y satelitales, sistemas de radiodifusión y sistemas de comunicación óptica. Su estructura fundamental se puede resumir en un modelo estándar, cuyos componentes básicos se indican en la Figura 2.1 [15].

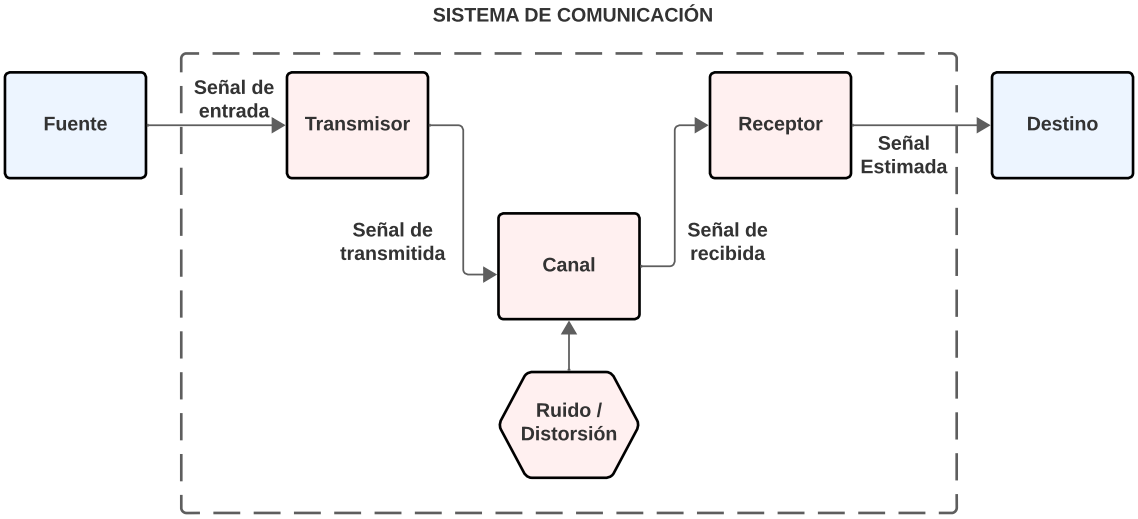

Figura 2.1: Elementos de un Sistema de Comunicación

- **Fuente:** genera el mensaje (Información) a transmitirse.
- **Transmisor:** modifica la señal de banda base para una transmisión eficiente. Incluye varias etapas: la codificación de la fuente en la cual la información se representa de manera eficiente por una secuencia de bits, la codificación de canal que introduce redundancia en la información con el fin de detectar y corregir los errores, la modulación que mapea la secuencia de bits para el medio de transmisión y la amplificación.
- **Canal:** medio utilizado en una red de comunicación, para transmitir la señal de información. Como lo es el canal inalámbrico, el canal con *Additive White Gaussian Noise (AWGN)*, el canal de fibra óptica, el canal satelital, canal de cable coaxial, etc. Cada tipo

de canal tiene un modelo matemático, que representa las características del canal que afectan a la transmisión de datos.

- **Receptor:** recupera la señal recibida y elimina la distorsiones provocadas por el canal. Incluye varias etapas que constan de los procesos inversos realizados en el transmisor: la demodulación convierte la señal recibida en un secuencia de bits, la decodificación del canal donde se utiliza la redundancia agregada para recuperar el mensaje, la decodificación de fuente que reconstruye el mensaje original y la corrección de errores.
- **Destino:** usuario al que se transmite el mensaje.

La señal se atenúa dentro del canal, fenómeno que aumenta en función de la distancia. Las señales se distorsionan por los fenómenos físicos, las interferencias y perturbaciones denominadas ruido, durante todo el trayecto.

Existen dos modos de comunicación, el broadcast o difusión donde interviene un solo transmisor y varios receptores, y la comunicación punto a punto que involucra un enlace entre un solo transmisor y receptor generalmente con un flujo bidireccional [16].

2.2. Modulación OFDM

La multiplexación por división de frecuencia ortogonal (Orthogonal frequency division multiplexing (OFDM)) es un método de modulación digital multiportadoras. Dónde el ancho de banda se divide en N subportadoras. Cada subportadora transporta un bloque de información, obtenido al dividir el flujo de símbolos que ingresan al modulador en N bloques. Al incrementar el tiempo de símbolo N veces, se añade un intervalo de guarda entre símbolos para eliminar el Interferencia Intersimbólica (ISI) en el receptor, provocado por los ecos que se producen en los canales multitrayecto. La características de ortogonalidad que presentan las subportadoras, permite que el espectro sea utilizado con más eficiencia.

Para generar la técnica OFDM se consideran los siguientes parámetros: el esquema de modulación, el intervalo de guarda y el número de subportadoras N. La Ecuación (2.1), muestra que el número de subportadoras N depende del ancho de banda y la duración del símbolo.

$$N = BW * Ts \tag{2.1}$$

OFDM incorpora técnicas de sincronización en tiempo y frecuencia entre el transmisor y re-

ceptor, y para canales móviles incorpora la ecualización del canal. Una de las ventajas de OFDM es que el proceso modulación y demodulación es fácil de implementar, mediante la transformada rápida de Fourier (*Fast Fourier Transform (FFT)*) y su inversa (*Inverse Fast Fourier Transform (IFFT)*). Se adapta a las condiciones del canal y es un esquema robusto para canales multitrayecto, ofrece una mayor eficiencia espectral en comparación con un esquema de una sola portadora, por lo que es utilizado en las comunicaciones inalámbricas. Sin embargo, tiene sensibilidad a los errores de sincronización ocasionados por el espaciamiento entre subportadoras.

Ortogonalidad

En OFDM se ubican las subportadoras de manera que sus bandas laterales se solapen, lo que en la recepción permite que no haya interferencia de la portadora adyacente. Esto se logra a través de la ortogonalidad de las portadoras, que se realiza a través del mapeador.

Las subportadoras se consideran como senos o funciones exponenciales, la Ecuación (2.2) muestra el espaciamiento entre cada una, que es igual al ancho de banda BW dividido para el numero N de subportadoras.

$$\text{Espaciamiento} = \frac{BW}{N} \tag{2.2}$$

La Ecuación (2.3), expone la frecuencia de cada subportadora. Y la Ecuación (2.4) muestra la duración del símbolo, donde el esquema de mapeo utilizado determina el tiempo del símbolo T_s.

$$f_n = f_0 + \frac{n}{Ts_{\text{OFDM}}} = f_0 + \frac{BW}{N}n \quad ,n = 0 \ldots . N-1 \tag{2.3}$$

$$Ts_{OFDM} = N * T_s \tag{2.4}$$

La Figura 2.2 muestra cómo se forma el espectro de OFDM, dónde se ubican cada una de las subportadoras (n índice de las subportadora) y el símbolo OFDM.

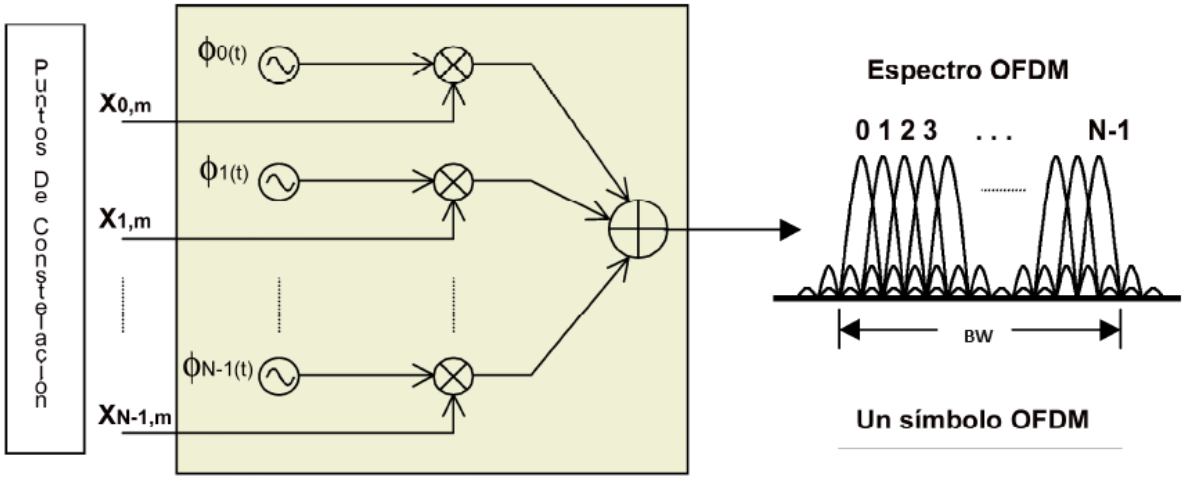

Figura 2.2: Generación de un símbolo OFDM [1]

2.2.1. Transmisor OFDM

En la Figura 2.3 la señal de entrada debe pasar por las diversas etapas que conforman al transmisor hasta ser transmitida. La señal se pasa de serie a paralelo y se compone de m bits de longitud en función del esquema de modulación, a cada uno se asigna a una subportadora mediante la Transformada Inversa de Fourier(*Inverse Fast Fourier Transform - IFFT*), que genera múltiples subportadoras a diferentes frecuencias ortogonales, consecuentemente disminuye la velocidad de bit en la portadora y la interferencia intersímbolo (Inter-symbol Interference - ISI).

La trama de datos en paralelo generada en función de la *IFFT* corresponde a la modulación OFDM, una vez ejecutado este proceso, los datos pasan a una forma serial, donde se agrega un intervalo de guarda (prefijo cíclico) en cada símbolo para eliminar el ISI.

Se incorpora el prefijo cíclico que es una copia de una parte de la señal OFDM, en el intervalo de guarda que es un espacio de tiempo Δ, la Ecuación (2.5) muestra cómo se calcula el tiempo de duración total del pulso enviado cuando se incluye el prefijo cíclico, dónde T es el tiempo de duración el símbolo OFDM sin prefijo adiciona Δ el intervalo de guarda. Esto permite un proceso más sencillo de ecualización.

$$T_s = T + \Delta \tag{2.5}$$

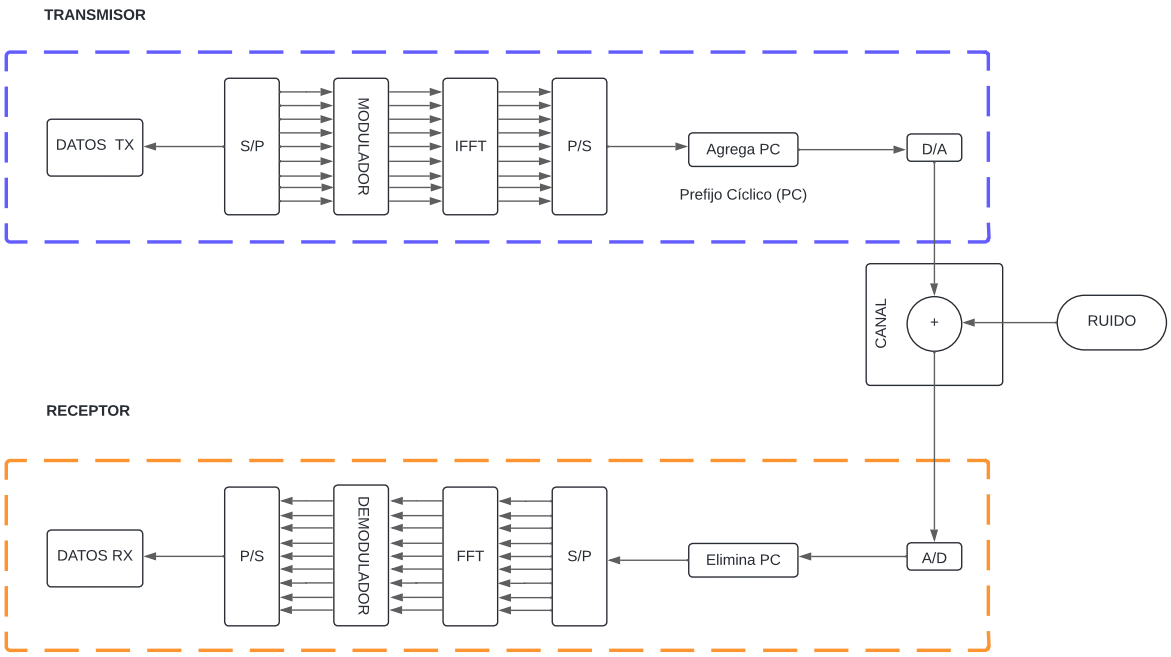

Figura 2.3: Sistema OFDM

2.2.2. Receptor OFDM

La señal OFDM es afectada por el canal *AWGN* y los desvanecimientos de tipo Rayleigh o Rice para canales móviles, antes de ser receptada por las antenas de los equipos receptores.

El receptor realiza el proceso inverso al transmisor que consiste en la extracción de prefijo cíclico seguido de la conversión de serie a paralelo, que ingresa al bloque de la FFT y finalmente al demodulador que obtiene la información transmitida. Posteriormente se realiza un proceso de sincronización, que generan errores en el intervalo de guarda que contiene al Prefijo Cíclico (CP) que se remueve del símbolo OFDM. Para mitigar los efectos que producen los desvanecimientos se realiza un proceso de ecualización, que requiere implementar técnicas de estimación de canal basadas en los símbolos pilotos.

Posteriormente se aplica la *FFT*, los símbolos que son demodulados y pasados a serial para obtener los datos transmitidos como lo muestra la Figura 2.3 [1].

2.3. Canal *AWGN*

Es un modelo de canal de ruido aditivo, que añade ruido blanco Gaussiano a la señal que está siendo transmitida a través del canal. El ruido es aleatorio, razón por la que no se puede determinar con exactitud el comportamiento del canal. Su espectro de frecuencia es uniformemente

continuo, y no se asocia con ningún tipo de desvanecimiento o parámetros del sistema. La Figura 2.4 muestra la representación a nivel de bloques de este tipo de canal [17]. Así mismo, la Ecuación (2.6), representa un modelo de canal AWGN, dónde se suma ruido $n(t)$ a la señal transmitida $s(t)$,

$$r(t) = s(t) + n(t) \tag{2.6}$$

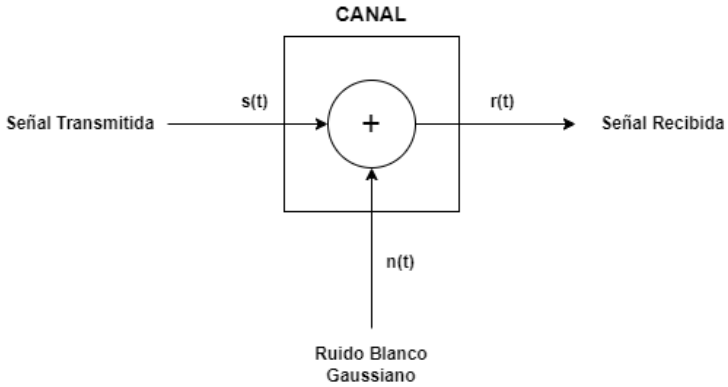

Figura 2.4: Esquema del Canal *AWGN*

donde, $s(t)$ es la señal transmitida, y $n(t)$ es la señal de ruido blanco Gaussiano que se agrega, el cual toma valores con una distribución idéntica e independientes entre sí.

Este modelo de canal, representa un canal de comunicación básico y se utiliza de manera estándar, ya que la señal se distorsiona por el ruido agregado [18].

2.4. *Machine Learnig*

Una de las ramas de la Inteligencia Artificial corresponden al aprendizaje automático como *Machine Learning (ML)*, que se compone de técnicas que permiten a las máquinas (PC) estimar o predecir en base a los datos de entrenamiento.

El proceso de entrenamiento de una técnica de *Machine Learning* consiste en:

- **Recopilación de los datos de entrenamiento:** se los recolecta en una base de datos, los parámetros independientes se determinan en función de las necesidades de la aplicación.
- **Procesamiento de datos:** se depuran y se transforma a un formato adecuado para su análisis.

- **Selección del modelo:** se elige en función de las necesidades del problema y las características de los datos.
- **Entrenamiento y evaluación :** en esta etapa se entrena el modelo con la base de datos establecida, y se analizan las métricas de evaluación, con el fin de verificar que las predicciones sean correctas frente a nuevos datos.
- **Ajuste del modelo:** con los datos de evaluación se modifican los parámetros, para lograr un mejor desempeño.

Maching Learning tiene cuatro métodos diferentes para realizar un aprendizaje automático [19].

- **Aprendizaje supervisado**

  El aprendizaje supervisado trabaja en función de los datos de entrenamiento y las respuestas de salida, estableciendo pares de entrada-salida en el conjunto de entrenamiento mismos que son etiquetados y utilizados para definir una función del modelo. Permite identificar qué relación existe entre los parámetros de entrada y los valores de salida, se manejan dos tipos de técnicas las de clasificación y de regresión en función de los datos.

- **Aprendizaje no supervisado**

  El aprendizaje no supervisado analiza únicamente los parámetros de entrada no etiquetados, este proceso está basado en los datos que se adquieren en tiempo real. Los tipos de técnicas no supervisadas son la agrupación, la estimación de densidad, aprendizaje de características,la búsqueda de reglas de asociación, la detección de anomalías, etc. Permitiendo identificar características, tendencias y estructuras.

- **Aprendizaje semisupervisado**

  El aprendizaje semisupervisado se puede definir como un método híbrido entre los supervisados y no supervisados introducidos anteriormente, el cual trabaja con datos etiquetados y sin etiquetar. Esta técnica proporciona mejores resultados al trabajar con los dos tipos de datos, ya que se entrena con datos etiquetados y aprende a la par con datos no etiquetados.

- **Aprendizaje por refuerzo**

  El aprendizaje por refuerzo es un algoritmo que permite evaluar automáticamente el conjunto de acciones más efectivas para un entorno o sistema en particular permitiendo una mejor automatización, que mejora su eficiencia, es decir, un enfoque impulsado por el

entorno. Es una herramienta para entrenar modelos de IA que puede ayudar a aumentar la automatización u optimizar la eficiencia operativa de diversas aplicaciones, sin embargo, no es recomendable utilizarla para resolver problemas básicos o sencillos, por la cantidad de recursos que utiliza.

2.4.1. Técnica de Regresión Lineal

Es una técnica de *Machine Learning* de tipo supervisado, cuyo funcionamiento se basa en el análisis del conjunto de entrenamiento. La Ecuación (2.7), muestra cómo se modela un conjunto a través de las variables dependientes e independiente, Y y X respectivamente y la relación entre estas [20], dónde β_0 representa el valor esperado cuando las variables independiente X son cero, $\beta_1, \beta_2, \beta_3, \beta_4, .., , \beta_n$, representan los coeficientes asociados a cada variable independiente y ε es el término de error obtenido de la diferencia del valor predicho y el valor observado Y.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots + \beta_n X_n + \varepsilon \tag{2.7}$$

En la técnica de regresión lineal existe la regresión simple y regresión múltiple. Su funcionamiento se basa en calcular los coeficientes de cada una de las variables independientes, para obtener una recta que representa el modelo del conjunto de datos ingresados.

El algoritmo de este método busca minimizar el valor del error cuadrático, estableciendo los coeficientes óptimos de la recta para el modelo. Para evaluar el rendimiento se establecen las siguientes métricas [21]:

Métricas de Evaluación

Se establece la definición de las siguientes variables que componen cada una de las métricas de evaluación.

- **r:** número de observaciones.
- y_i**:** Valor real observado de la i-ésima observación.
- $\hat{y}_i$**:** Valor predicho de la i-ésima observación.
- $\bar{y}$**:** media de los valores observados y_i.

- **Error Cuadrático medio - *Mean Squared Error (MSE)*:** en la Ecuación (2.8) se muestra el cálculo del error cuadrático medio, que es la media entre las diferencias entre las predicciones y los valores reales, un valor bajo y cercano a cero indica un buen rendimiento del modelo.

$$\text{MSE} = \frac{1}{r}\sum_{i=1}^{r}(y_i - \hat{y}_i)^2 \tag{2.8}$$

- **Error Absoluto medio - *Mean Absolute Error (MAE)*:** en la Ecuación (2.9) se calcula el error absoluto medio, media de las diferencia entre el valor absoluto de las predicciones y los valores reales, un valor bajo y cercano a cero indica un menor número de errores.

$$\text{MAE} = \frac{1}{r}\sum_{i=1}^{r}|y_i - \hat{y}_i| \tag{2.9}$$

- **Raíz del Error Cuadrático medio - *Root Mean Square Error (RMSE)*:** se busca que la escala del error sea igual a la escala de los datos ingresados, de la misma manera que MSE un valor bajo y cercano a cero indica un modelo perfecto. En la Ecuación (2.10) se calcula la Raíz del Error Cuadrático medio.

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{r}\sum_{i=1}^{r}(y_i - \hat{y}_i)^2} \tag{2.10}$$

- **Coeficiente de determinación:** es un indicativo de la variabilidad del modelo como se muetras en la Ecuación (2.11), 1 indica una perfecta variabilidad, y 0 indica que el modelo no se ajusta a los datos de entrada.

$$R^2 = 1 - \frac{\sum_{i=1}^{r}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{r}(y_i - \bar{y})^2} \tag{2.11}$$

2.4.2. Técnicas de Clasificación

2.4.2.1. Modelo del Árbol de Decisión

Es un modelo de aprendizaje automático supervisado de clasificación mediante particiones binarias, cuyo objetivo es predecir a qué clase pertenece un caso en función de una o más

características conocidas. Los algoritmos utilizados permiten generar modelos tanto para variables cuantitativas (regresión) como para variables cualitativas (clasificador).

Esta técnica está formada por nodos de decisión que realizan una pregunta de decisión sobre una variable específica. Las respuestas a estas preguntas pueden generar nuevos nodos de decisión, denominados ramas. Utiliza el algoritmo de partición binaria recursiva para estructurar el árbol, realizando divisiones de un subconjunto de datos asociadas a una de las variables, esto separa el conjunto de datos en nuevos subconjuntos.

Cada nuevo dato, del que se conocen los parámetros, recorre las ramas del árbol basándose en las reglas y decisiones generadas, formando un recorrido entre los nodos de decisión y el nodo terminal.

Clasificadores a partir de particiones

Un clasificador es un algoritmo que determina a qué clase debe asignarse los valores de entrada. El método de construcción del clasificador comienza eligiendo una variable que permite la mejor separación de los casos existentes, como se muestra en la Figura 2.5. Los datos se dividen en grupos tan diferentes entre sí como sea posible, pero de manera que los casos que hay dentro de cada subgrupo tengan la máxima semejanza posible entre sí, obteniendo homogeneidad entre los datos [22]. Generando la estructura del árbol con los nodos de decisión que permita al modelo clasificar eficientemente los nuevos valores.

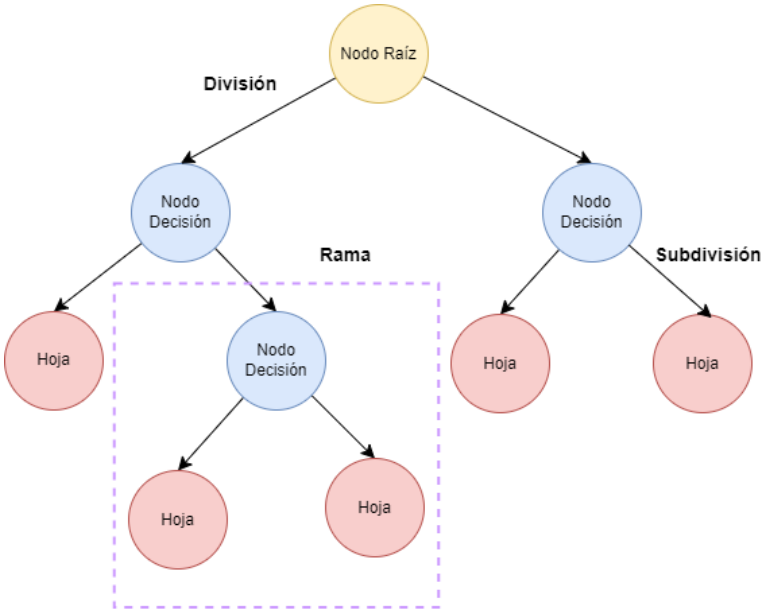

Figura 2.5: Estructura de un árbol de decisión

Para desarrollar el modelo, se requiere de una muestra de la base de datos, que mediante un método inductivo genera las reglas que serán base del modelo.

Error de Entrenamiento

El error de entrenamiento $R(h)$ se mide por el número de errores cometidos en la predicción y permite conocer el ajuste del modelo predictivo a los datos de entrenamiento. Se utiliza un conjunto de prueba que divide el conjunto de aprendizaje en dos subconjuntos C1 y C2. Se entrena con el subconjunto C1 y se verifica la precisión de la clasificación con el subconjunto C2. Se considera que C1 como C2 sean independientes o idénticamente distribuidos, asegurando que la selección de un subconjunto no influya en otro. La Ecuación (2.12) muestra como calcular el error de entrenamiento $R(h)$.

$$R(h) = \frac{1}{N}\sum_{n=1}^{N} 1\left[h\left(x_n\right) \neq j_n\right]. \tag{2.12}$$

donde:

- N: es el número total de ejemplos del cojunto de entrenamiento.
- $h(x_n)$ clasificador aplicado al n-ésimo ejemplo x_n.
- y_n**:** clase verdadera.

Si $1\left[h\left(x_n\right) \neq j_n\right]$**:** toma el valor de 1 la predicción $h(x_n)$ no coincide con la clase verdadera y_n y si el valor es cero la predicción $h(x_n)$ coincide con la clase verdadera y_n.

Clasificador con estructura de árbol binario

Este tipo de clasificador se construye a partir de una sucesión recursiva de divisiones de los subconjuntos formando nuevos subconjuntos en base a las reglas de decisión, partiendo del conjunto inicial X. A los subconjuntos o nodos que no pueden dividirse se denominan conjuntos terminales. De esta manera, un nuevo valor x que ingresa al árbol recorre las divisiones y condiciones hasta llegar al nodo terminal correspondiente a sus características, como se observa en la Figura 2.6, que muestra la estructura general de un árbol binario, destacando los nodos de decisión, los nodos terminales y las diferentes ramificaciones que se forman.

Criterio de evaluación

El criterio de evaluación se aplica al modelo luego de ser entrenado, para conocer si este se

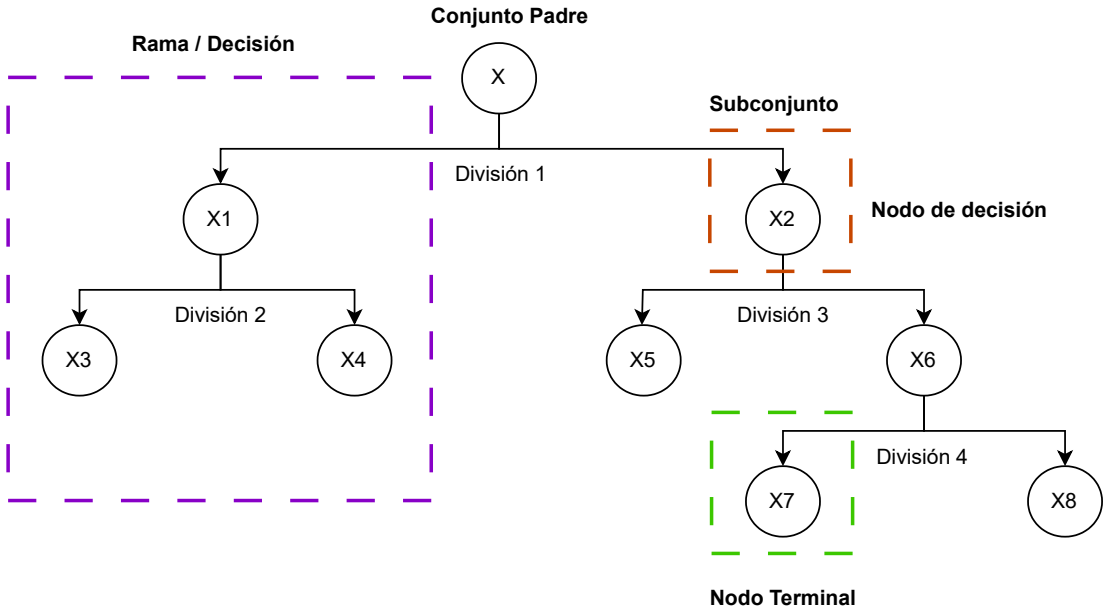

Figura 2.6: Diagrama Árbol de Decisión

ajusta a los datos de entrenamiento y si clasifica eficientemente los nuevos datos en cada uno de las clases. Se basa en la entropía H del conjunto, midiendo la aleatoriedad de los datos con características diferentes. Este desorden indica cuán mezclado está el subconjunto de datos con respecto a otras clases [23].

La entropía del conjunto de datos se define matemáticamente como:

$$H(D) = -\sum_{i=1}^{c} p_i \log_2 (p_i) \tag{2.13}$$

donde:

- **D:** es el conjunto de datos.
- **c:** clases de clasificación.
- p_i**:** número de ejemplos que pertenecen a una clase i.

La entropía es máxima cuando los datos están perfectamente mezclados y es cero cuando todos los ejemplos pertenecen a una sola clase. En los árboles de decisión esta métrica se utiliza para dividir los conjuntos de datos buscando reducir su valor a 0.

2.4.2.2. Modelo k-vecinos más cercanos (*K-Nearest-Neighbor - KNN*) de predicción

Es un modelo de *Machine Learning* de tipo supervisado que asigna la etiqueta de una clase a un valor dependiendo de sus vecinos más próximos (ver la Figura 2.7).

Durante esta etapa el modelo aprende en función de la base de datos ingresada, donde k hace referencia a la cantidad de vecinos, que se consideran para determinar la clasificación de los n grupos que se establecen en la base de datos de entrenamiento [20].

En la etapa de clasificación, el modelo entrenado hace uso de la configuración que se estableció durante el entrenamiento, para clasificar nuevos valores en base a la similitud de los vecinos más próximos.

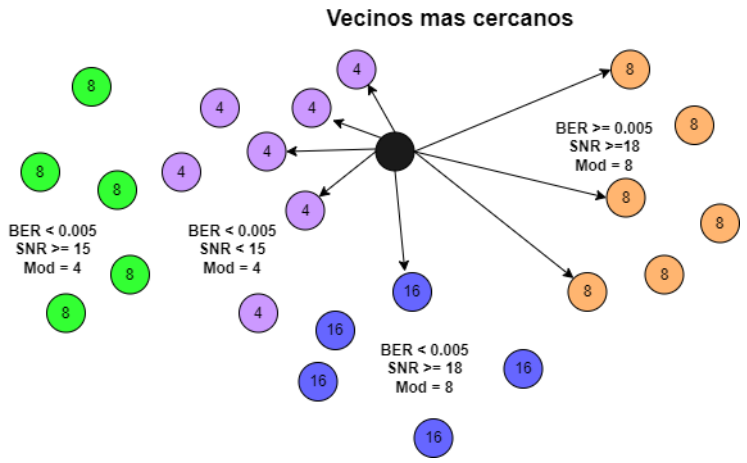

Figura 2.7: Comportamiento del modelo de KNN para la clasificación

La implementación requiere gran capacidad de recursos y procesamiento (CPU), por este motivo se necesita que la base de datos sea reducida y con el menor número de parámetros para la clasificación o estimación [20].

Dependiendo de la aplicación, en ocasiones es necesario considerar más de un vecino, mediante los cuales se establecen los casos de clasificación para los nuevos valores que son evaluados por el algoritmo, que clasifica en función de la distancia Euclidiana entre el nuevo valor y los k vecinos más próximos. Esta clasificación es posible, ya que se fundamenta en que cerca de un punto existen puntos similares, y cuando se tengan nuevos valores se encuentran sus *k-nearest neighbors* mediante su entrenamiento.

El funcionamiento del modelo de *K-Nearest Neighbor (KNN)* se establece de la siguiente manera:

1. Calcula la distancia entre el valor a clasificar y los valores establecidos durante la etapa

de entrenamiento.

2. Selecciona los k vecinos más cercanos en función de la distancia.
3. Realiza un proceso de votación entre los k vecinos para determinar la clase a la que pertenece.

- **Distancia entre el nuevo valor y los k vecinos**

$$d\left(q, x_i\right) = \sum_{f \in F} w_f \delta\left(q_f, x_{if}\right) \tag{2.14}$$

En la Ecuación (2.14) se muestra el cálculo de la distancia, donde F representa al conjunto, w_f son los pesos y q_f x_{if} son los valores de las características para el nuevo valor y el conjunto de datos ya entrenados. δ representa a la función de la distancia euclidiana propia de esta técnica.

- **Proceso de votación**

Durante esta etapa se determina la clase a la que el nuevo valor pertenece, en función de las clases de sus vecinos más próximos. Se realiza luego del cálculo de las distancias con respecto a todos los puntos más cercanos del conjunto de entrenamiento, utilizando la Ecuación (2.15). Una vez identificados los *k-nearest* se realiza este proceso de votación donde cada vecino determina en función de la distancia a la clase que pertenece el nuevo valor mediante un voto (peso), y la clase con mayores votos se asigna al nuevo valor.

$$\text{Vote}\left(y_j\right) = \sum_{c=1}^{k} \frac{1}{d\left(q, x_c\right)^p} 1\left(y_j, y_c\right) \tag{2.15}$$

donde:

- k**:** es el número de vecinos,
- $d(\mathbf{x}, \mathbf{x}_c)$**:** es la distancia Euclidiana respecto a cada punto $\mathbf{x}_c$,
- p**:** permite modificar cómo influye la distancia a los diversos puntos,
- $1(y_j, y_c)$**:** es una función indicadora que devuelve 1 si la etiqueta coincide y 0 en caso contrario.

- **Verdadero Positivo - True Positive (TP):** representa a una clasificación correcta, donde

las predicciones positivas se establecen como positivas.

- **Falso Positivo - False Positive (FP):** representa a una clasificación incorrecta, donde las predicciones negativas se establecen como positivas.
- **Verdadero Negativo - True Negatives (TN):** representa a una clasificación correcta, donde las predicciones negativas se establecen como negativas.
- **Falso Negativo - False Negatives (FN):** representa a una clasificación incorrecta, donde las predicciones positivas se establecen como negativas.

Métricas de evaluación

Una vez que el modelo está entrenado se establecen las siguientes métricas para evaluar su desempeño:

- Precisión (*Accuracy*): determina la relación entre las predicciones correctas sobre el total de predicciones realizadas, en la Ecuación (2.16) se muestra la forma de calcular.

$$Accuracy = \frac{N°PrediccionesCorrectas}{TotaldePredicciones} \tag{2.16}$$

- *Recall*: establece la proporción de predicciones que pertenecen a la clase correcta, en la Ecuación (2.17) se muestra la forma de calcular, mediante el TP y el FN.

$$Recall = \frac{TP}{TP + FN} \tag{2.17}$$

- Matriz de Confusión: evalúa el rendimiento del modelo de clasificación, tiene una estructura donde representa los aciertos y errores del modelo [24]. En las columnas de la matriz se establece las clases de las predicciones y en la filas las clases reales.

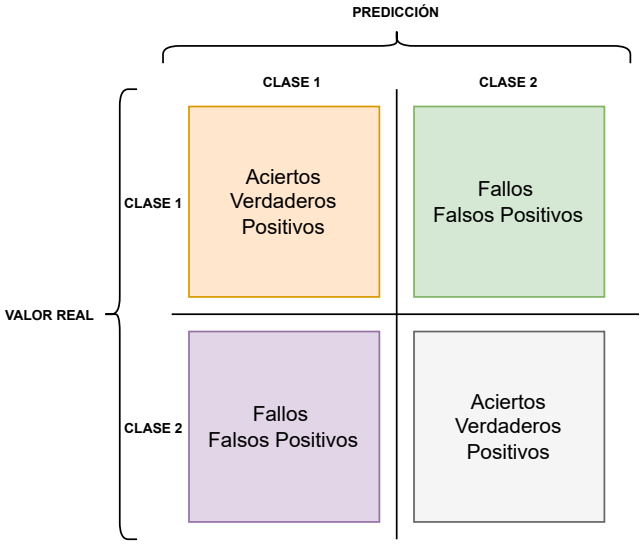

Figura 2.8: Estructura de la Matriz de Confusión

En la Figura 2.8 se muestra la estructura de la matriz de confusión, en las filas se ubican los valores reales y en las columnas las predicciones correspondientes a cada una de las clases. En esta matriz se presenta el porcentaje correspondiente a las cuatro opciones.

3. Revisión Literaria

La modulación OFDM adaptativa tiene la capacidad de ajustar dinámicamente sus parámetros en tiempo real según las condiciones del canal de comunicación. Esta es robusta en entornos con interferencias y atenuaciones. Permite una transmisión eficiente y una alta utilización de recursos en el canal de envío de la señal [25].

La modulación OFDM adaptativa es implementada en sistemas de comunicación acústicos subacuáticos (UA), debido a su alto nivel de ruido por múltiples factores, como el ancho de banda limitado, las variaciones rápidas del canal y las propiedades de la señal. Con base en la señal-ruido (SNR) que se obtiene en la estimación del canal mediante subportadoras piloto, se elige una nueva modulación. En cada bloque de datos transmitido, se incluyen subportadoras piloto uniformemente espaciadas en la banda de frecuencia. La señal recibida en estas subportadoras se utiliza para estimar el canal y, a partir de esta estimación, se calcula el SNR en el dominio de la frecuencia. También utiliza subportadoras nulas, en las que no envía información, para facilitar la estimación del desplazamiento de frecuencia de la portadora, lo que aumenta la velocidad de datos en los sistemas de comunicación UA [9]. Otras técnicas de selección para este tipo de sistemas manejan como parámetros la frecuencia de la portadora y el ancho de banda del canal en conjunto con el SNR [26].

La modulación OFDM adaptativa mejora el BER del sistema y permite implementar modulaciones de orden superior, permitiendo aumentar la velocidad de los datos transmitidos. El esquema *Adaptive dual-mode OFDM with index modulation* (A-DM-OFDM-IM) propone seis valores de umbral para seleccionar el tipo de modulación. A-DM-OFDM-IM utiliza retroalimentación limitada para adaptar la selección de modulación según las condiciones del canal, y una estructura adaptativa que selecciona el tipo de modulación según el SNR promedio. El receptor envía señales piloto conocidas al transmisor, para estimar la respuesta del canal y calcular el SNR en cada subportadora. Esto permite que el transmisor elija el tipo de modulación de acuerdo con las variaciones del canal en tiempo real, considera un canal de retorno imperfecto, lo que significa que puede haber ruido en la comunicación de la información de retroalimentación [5].

Fang et al. [10] propone un esquema de transmisión adaptativa OFDM centrada en la estimación del canal, tanto la estimación y elección de la modulación la realiza el receptor. La técnica de selección establece un umbral del parámetro SINR (relación señal/interferencia más ruido), cuyo valor se establece en base a la forma de asignación de las subportadoras, que puede ser de tres modos: una modulación diferente para cada subportadora, la misma modulación para

todas las subportadoras y modulaciones por grupos de subportadoras.

El parámetro principal para la selección de los esquemas de modulación es el nivel de SNR estimado en la señal recibida, en canales lentos y con desvanecimiento plano los errores se concentran específicamente en un grupo de subportadoras, a las cuales se excluye de la transmisión. Si bien este esquema afecta al rendimiento se mitiga con el uso de modulaciones más robustas, su algoritmo de adaptabilidad se basa en enviar bits pilotos que estiman el canal y el SNR [11].

Se establecen dos tipos de modulación adaptativa, en el primer tipo todas las subportadoras transmiten y en el segundo tipo lo hacen únicamente las subportadoras con mejor rendimiento del BER. En la primera, los datos se transmiten continuamente, y en la segunda, se utiliza el bloqueo de transmisión para evitar la transmisión en subportadoras donde el Bit Error Rate (BER) es alto, a través de la utilización de subportadoras nulas. Se pueden emplear varias técnicas de modulación como son la selección de modulación basada en el SNR, BER o en la eficiencia espectral, esta se calcula con respecto a la calidad del canal para asignar una modulación óptima a cada subportadora. Se emplean esquemas de transmisión adaptativa por subbanda, donde se dividen todas las subportadoras en bloques de subportadoras adyacentes denominados subbandas. Para la selección de las modulaciones se usa un algoritmo que selecciona el modo de modulación óptimo para cada subbanda en función de la estimación del canal. El algoritmo se basa en la comparación del SNR promedio de cada subbanda con los umbrales de conmutación predefinidos para determinar el modo de modulación óptimo para la subbanda correspondiente. De esta manera se genera una mejora en el bloqueo de transmisión, lo cual permite un mejor rendimiento en el BER [27]. Las interferencias del canal influyen directamente en el desempeño del sistema de comunicación OFDM adaptativa, por lo que se han implementado técnicas de adaptabilidad basadas en la potencia en cada subportadora. Si bien presentan beneficios en la minimización del BER tienen un alto costo computacional ya que el número de iteracciones es proporcional al número de subportadoras empleadas [12]. Otros parámetros para la selección del esquema de modulación que se consideran son el ancho de banda, la longitud de la trama y multiplicidad de la frecuencia Doppler [10] [12].

Una de las técnicas más usadas para la modulación adaptativa es la Adaptive Modulation and Coding (AMC), esta técnica adapta y codifica la modulación y la tasa de codificación en cada transmisión en función de la estimación del canal, para la obtención del SNR. Esta selección de modulación y codificación se basa en umbrales de SNR para mantener una tasa de error de bit (BER) objetivo [28][29].

La AMC adapta la potencia de transmisión, el esquema de modulación y la tasa de codificación con respecto a la estimación del canal en el receptor, con lo cual aumenta la capacidad del canal. Al seleccionar la modulación y el esquema de codificación óptimo con respecto a las condiciones del canal, se logra mejorar el rendimiento del sistema bajo modulación OFDM [30].

En los algoritmos adaptativos convencionales, desarrollados por Hughes-Hartogs, Chow y Levin-Campello, el parámetro fundamental para seleccionar el esquema de modulación es el SNR. Para mejorar el SNR, estos algoritmos, con diferentes niveles de complejidad, asignan bits y energía, al tiempo que aprovechan los recursos del espectro, en [31], se describe el procedimiento de cada tipo de algoritmo.

En la actualidad se utiliza la inteligencia artificial como un recurso para la implementación de una modulación adaptativa en OFDM. *Machine Learning (ML)* y *Deep Learning (DL)* son técnicas utilizadas para aprender el comportamiento del canal del sistema de comunicación, para la selección de una modulación adecuada en las diferentes subportadoras con respecto al SNR obtenido en el receptor y enviada mediante un feedback al transmisor [26]. *Machine Learning* es un área de la Inteligencia Artificial que utiliza algoritmos estadísticos para diseñar sistemas inteligentes. DL es una evolución más avanzada de *Machine Learning* que permite que las máquinas tomen decisiones sin intervención humana. La principal diferencia es que, en DL las características se aprenden automáticamente en varios niveles, mientras que en *Machine Learning* se basan en características creadas por ingenieros [32].

El reconocimiento automático de modulación (AMR) consta de dos subsistemas: extracción de características y clasificación. Las características se extraen del preprocesamiento y se categorizan utilizando clasificadores tradicionales, basados en *Machine Learning* y DL. Se extraen características relevantes de las señales moduladas, como la amplitud, la fase, la frecuencia, la energía y el SNR. Estas características se utilizan como entradas para los algoritmos de clasificación, como redes neuronales artificiales (ANN), máquinas de vectores de soporte (SVM), máquinas de aprendizaje extremo (ELM). En DL estas características se aprenden directamente de los datos de entrada, sin necesidad de una extracción manual previa como lo es en *Machine Learning*. El método que se utiliza para la modulación adaptativa es aplicar diferentes niveles de modulación como QPSK, 16-QAM, 64-QAM, a grupos de subportadoras. La selección del nivel de modulación se basa en consideraciones de tasa de transmisión de datos, calidad del canal, SNR, robustez frente a interferencias, y eficiencia espectral. La elección del nivel de modulación óptimo dependerá de un equilibrio entre estos factores para lograr un rendimiento de transmisión óptimo [32].

La identificación de modulación de señal (SMI) es implementada por la extracción de datos mediante aprendizaje automatizado, ya que los métodos tradicionales, no basados en inteligencia artificial, llegan a ser muy limitados y la implementación de OFDM es poco práctica. Debido a estos problemas, se opta por el uso de Deep Learning (DL) que usa un grupo de neuronas que aprenden automáticamente con respecto a los datos recibidos, estas usan redes neuronales convolucionales en las cuales se tiene dos conjuntos de datos con diferentes modulaciones y verificación de la robustez del método con respecto al SNR. Las redes neuronales convolucionales (CNN) pueden aprender a extraer características de amplitud y fase de las señales de modulación, lo que les permite capturar patrones complejos del comportamiento. Estas características son usadas para clasificar la señal en una de las categorías de modulación predefinidas. El método utilizado para la selección de la modulación correcta con respecto al canal son las CNN. El método de aprendizaje usado por DL es que, en cada ciclo, se seleccione aleatoriamente algunas neuronas en la capa neuronal, se ocultan temporalmente y después se realiza el proceso de entrenamiento y optimización de la red neuronal [4].

Machine Learning es importante para la modulación adaptativa, debido a que se adecua en distintas situaciones y comportamientos del sistema de comunicación para una mejora en el rendimiento de transmisión. Este modelo de aprendizaje automático tiene como objetivo optimizar la eficiencia espectral y la confiabilidad en sistemas MIMO-OFDM mediante técnicas de selección de antenas (TAS) y adaptación de potencia (PA), lo que lo hace adecuado para su aplicación en sistemas de comunicación, el rendimiento del modelo es superior al de la modulación adaptativa basada en reglas con niveles SNR. Se entrenan clasificadores supervisados, como máquinas de vectores de soporte (SVM), árboles de decisión o redes neuronales, utilizando conjuntos de datos etiquetados que contienen información sobre las características de la señal recibida y la modulación correspondiente. Donde esta información es utilizada para describir las propiedades del sistema. Estas características comprenden parámetros extraídos de la señal recibida, la matriz del canal, la modulación y la asignación de potencia, siendo fundamentales para el análisis y la toma de decisiones de modulación. Una vez entrenados, estos clasificadores pueden predecir la mejor modulación para una señal dada en función de sus características [33].

Dentro de la IA se presentan las superficies inteligentes (SI), las cuales buscan mejorar la eficiencia energética, la cobertura y rendimiento del espectro de las comunicaciones inalámbricas, mediante la distribución discreta de las tramas de entrenamiento [34]. En el área de DL, se utilizan redes neuronales como técnicas de aprendizaje automático para predecir el

índice de modulación basándose en las métricas obtenidas en la estimación del canal [35]. Estas redes son utilizadas tanto en la estimación del canal con en la detección de señales. La IA ha permitido realizar esta estimación con solo una fase de entrenamiento, lo que reduce los recursos en la transmisión. Esto ha posibilitado el estudio y la propuesta de sistemas OFDM sin prefijo cíclico (cyclic prefix - CP) y con mayor número de señales piloto [36]. Las redes neuronales se consideran adecuadas debido a que requieren menos datos y tiempos de entrenamiento más cortos, ofreciendo baja complejidad, gran robustez y mejor rendimiento en términos de tasa de error de bit (BER).

Para el proceso de entrenamiento, se establece un conjunto de clases, las cuales contienen diversas métricas obtenidas de la estimación del canal [37]. Este tipo de clasificadores selecciona el esquema de modulación basándose en las entradas de la relación señal-ruido (SNR) en las subportadoras, y ajustándose dinámicamente según las restricciones del BER [38].

En el caso de MIMO-OFDM, mediante técnicas de *Machine Learning* se establece el esquema de modulación y codificación basándose en las condiciones del canal. Según Alvario y Heath [38], la modulación adaptativa se entiende como un problema de clasificación, ya que implica la selección en los parámetros del transmisor. Se emplea el método de *Machine Learning* denominado *k-nearest neighbors* para la adaptación de los enlaces, donde no es necesario conocer los datos exactos, sino comprender los datos de entrenamiento. Esto es fundamental en casos donde no existe un patrón predefinido a seguir. Se busca maximizar el rendimiento manteniendo el BER óptimo, es importante la memoria de procesamiento para la seleccionan de los vecinos más próximos al BER objetivo.

Entre las principales ventajas de la incorporación de la IA en los sistemas OFDM se encuentra el incremento en la velocidad de transmisión de los datos y la estimación más precisa del canal. Sin embargo, a pesar de ser un recurso valioso presenta ciertas desventajas. Dado que el procesamiento de datos es intensivo, se requiere un mayor recurso de memoria. Esto ha llevado a la búsqueda de técnicas para mejorar esta situación, ya que las tablas de consulta tienden a ser de gran tamaño. Este problema es especialmente evidente en métodos de *Machine Learning* como el de *k-nearest-neighbors*, donde los recursos de memoria necesarios crecen al aumentar el tamaño del conjunto de entrenamiento [37][38].

La implementación de *DL*, requiere de hardware con una capacidad y procesamiento alto, como unidades de procesamiento gráfico (GPU) de alto rendimiento, a diferencia de *Machine Learning*, que pueden ejecutarse en plataformas más ligeros, lo cual hace que *Machine Learning* sea más aplicable en entornos prácticos.

4. Implementación de *Machine Learning* para OFDM adaptativo

4.1. Herramientas de simulación

La implementación de técnicas que permitan ajustar dinámicamente el esquema de modulación utilizado por OFDM se hace utilizando la plataforma de programación Matlab, el cual da soporte en el diseño y simulación de sistemás de comunicación, con una precisión cercana a la realidad, por su conjunto de herramientas y funciones que modelan los comportamientos y características de los sistemas reales.

4.2. Implementación de OFDM Adaptativo

El modelo que se toma como base para la implementación de OFDM adaptativo está basado en Adaptive Modulation and Coding (AMC). En la Figura 4.1 se muestra la AMC en un sistema OFDM, el cual ajusta la modulación y la velocidad de codificación en cada transmisión según las condiciones del canal, la adaptación se realiza mediante una retroalimentación desde el receptor por un canal perfecto, donde se envía el índice de modulación adaptativo. En este contexto, se ajusta tanto el transmisor como el receptor a una modulación aceptable, con base en la tasa de bits erróneos (BER) objetivo. Para esto se consideran diferentes tipos de modulación hasta alcanzar el BER deseado, mediante este método se alcanza un BER de 0.001 para un SNR mayor a 22dB [39].

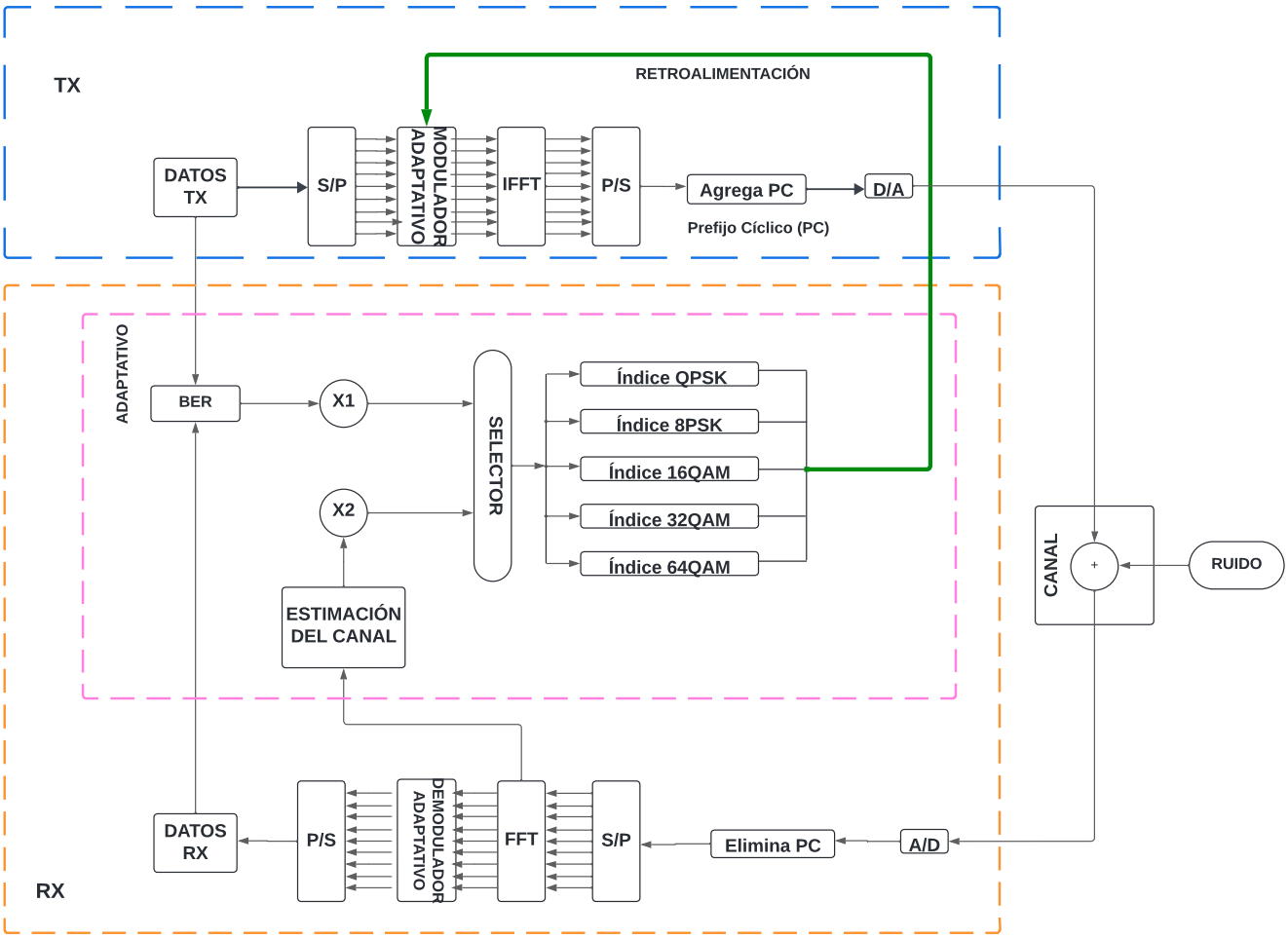

Figura 4.1: OFDM adaptativo convencional

Un BER de 0.001 en valores bajos de SNR, típicamente inferiores a 20 dB en redes Wi-Fi, es importante para aplicaciones que necesitan un rendimiento alto. Un rendimiento alto se considera como la capacidad de que un sistema es eficiente, confiable y posee una alta velocidad de datos, baja latencia y alta capacidad de datos. Los datos que necesitan tener mayor fiabilidad en el receptor son aplicaciones de Voz sobre IP (VoIP), en esta aplicación es necesario tener un Bit Error Rate (BER) menor a 0.001, caso contrario esta señal puede degradar y generar sonidos no deseados como ecos o corte en la voz [40].

4.2.1. Parámetros de simulación Wifi

Los parámetros de OFDM se centran en el entorno de IEEE 802.11a, como se muestra en la Tabla 4.1. Los cuales serán usados para la implementación del sistema OFDM.

Tabla 4.1: Parámetros definidos

Parámetros	Valores
Número de subportadoras (N)	64
Número de subportadoras Pilotos $(\mathrm{N}_{\text{pilotos}} = \mathrm{N}/16)$	4
Número de subportadoras con datos $(N - N_{pilotos})$	60
Intervalo de guarda	0.8μ s
Posición de la subportadoras Piloto	-21, -7, 7 , 21.
Ancho de Banda	20 MHz
Canal	*Additive White Gaussian Noise (AWGN)*
Modulaciones Digitales	QPSK, 8PSK, 16QAM. 32QAM, 64QAM

- **Subportadoras piloto:** transmiten los símbolos piloto, que son valores conocidos utilizados para medir la respuesta del canal.
- **Subportadoras de datos:** transmiten los datos de información.
- **Intervalo de guarda:** espacio de tiempo $\triangle$ incorporado al tiempo del símbolo, utilizado para eliminar el ISI y neutralizar los efectos de sincronización.
- **Posición de la piloto:** ubicación de las subportadoras piloto dentro del conjunto de subportadoras totales.
- **Ancho de banda:** rango de frecuencias utilizado para la transmisión de datos.

4.2.2. Transmisión

Para el transmisor del sistema OFDM adaptativo se usa el modelo de la Figura 4.2, se envía un total de 100.000 datos que de dividen en bloques para ser enviados por 60 subportadoras de datos, y se establecen 4 subportadoras pilotos, dando un total de 64 subportadoras en el sistema y se tiene un intervalo de guarda de 0.8μ s entre subportadoras, estos datos son enviados en un ancho de banda de 20 MHz.

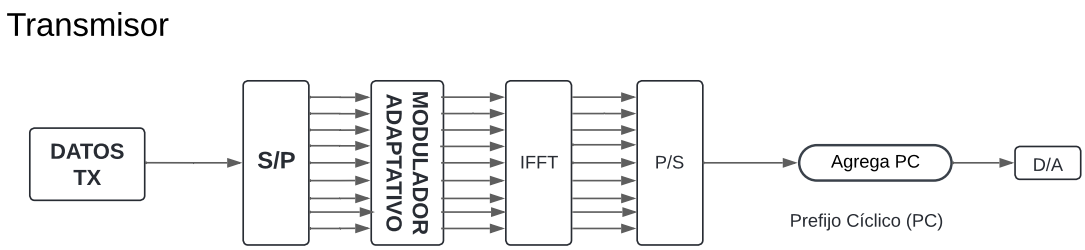

Figura 4.2: Transmisor OFDM adaptativo

4.2.2.1. Generación de datos

Para generar los datos que se envían por el sistema OFDM adaptativo se genera una secuencia aleatoria de bit con una longitud de 100000 bits.

4.2.2.2. Serie a paralelo

Para aplicar la *Inverse Fast Fourier Transform (IFFT)* se necesita tener los datos en paralelo, por lo que el bloque convierte el flujo de símbolos en serie que llega del modulador en paralelo, asignando cada símbolo modulado a las subportadoras.

La cantidad de datos por subportadoras varía con respecto a la modulación seleccionada debido a los bits por símbolos de cada modulación. En la Ecuación (4.1), se calcula la cantidad de datos por subportadoras, esto depende de la cantidad de subportadoras con datos en la que N es la cantidad de datos, $N_{pilotos}$ es el número de subportadoras pilotos y el índice de modulación usada K.

$$\text{Datos portadora} = (N - N_{pilotos}) * K \tag{4.1}$$

4.2.2.3. Modulación Adaptativa

La modulación usada en el transmisor se selecciona con respecto al índice de modulación obtenido de la retroalimentación del receptor. Tomando en cuenta que se tendrá cinco diferentes tipos de modulación para la selección de la modulación siguiente, se toma en cuenta el valor del BER de cada modulación en los diferentes SNR en el sistema de OFDM, Figura 4.3.

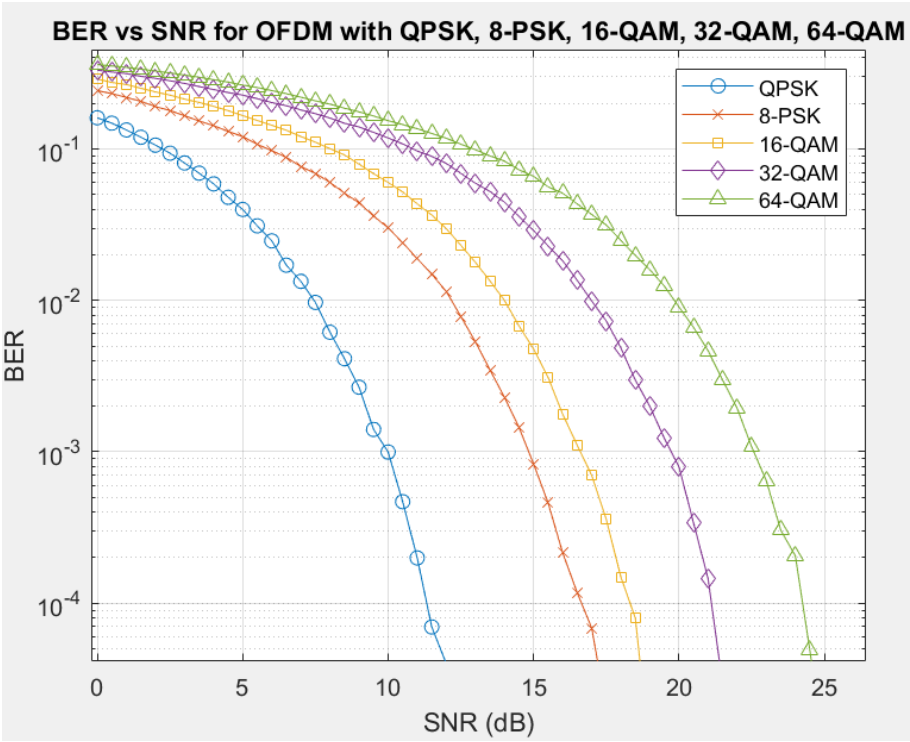

Figura 4.3: Comportamiento del BER vs SNR para las diferentes modulaciones

Mediante el índice de modulación obtenida en la retroalimentación por un canal ideal, del receptor se selecciona una de las cinco modulaciones disponibles, la que se utilizará para la transmisión de información, Figura 4.4

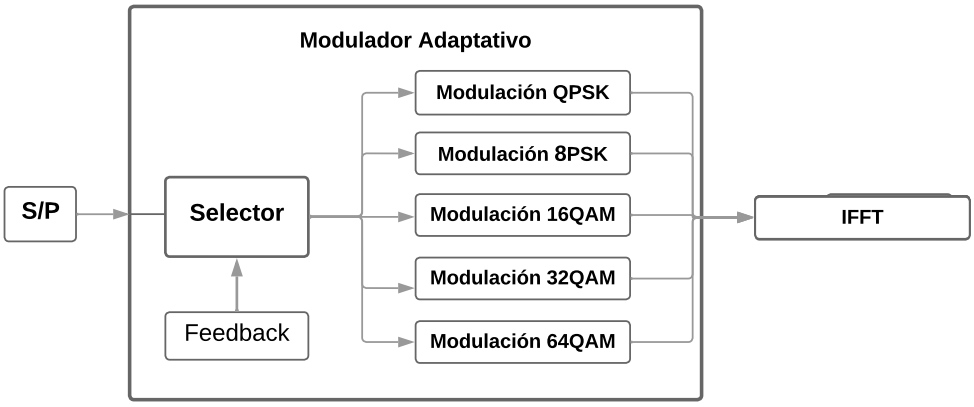

Figura 4.4: Modulador Adaptativo, selector

4.2.2.4. Transformada rapida de Fourier inversa (*IFFT*)

OFDM requiere una señal que sea la suma de señales y con estas, en el dominio de la frecuencia, mantengan constantes las frecuencias adyacentes y la separación entre señales. La Ecuación (4.2), muestra el cálculo de la *IFFT*, que toma los valores paralelos previamente

mapeados y los convierte en muestras en el dominio del tiempo.

La *IFFT* toma un número N muestras en el tiempo y entrega como resultado N muestras en el dominio de la frecuencia.

$$x(n) = \frac{1}{N} \sum_{N-1}^{k=0} X(k) e^{j\left(\frac{2\pi}{N}\right)*k*n} \tag{4.2}$$

La *IFFT* en el transmisor envía datos de forma paralela, por lo que se necesita un bloque que convierta la transmisión nuevamente al tipo serie.

4.2.2.5. Paralelo a Serie

Para el envío de la señal procesada por el sistema OFDM es necesario tener datos en serie. Su función es tomar la información insertada en las subportadoras durante el proceso de asignación de subportadoras y agruparlas en un solo flujo que se envía en el dominio de la frecuencia.

4.2.2.6. Prefijo Cíclico

Prefijo cíclico (CP) es importante para evitar la ISI y completar el símbolo OFDM. La duración del CP puede ser de 1/4, 1/8 hasta 1/32 de la duración del símbolo. La duración del prefijo cíclico no debe pasar el 1/4 de la duración del símbolo OFDM [41]. Por lo que se tiene un prefijo cíclico de 16 bits.

4.2.3. Canal *AWGN*

Una vez colocado los prefijos cíclicos se procede a añadir el ruido, por lo que la señal del transmisor pasa por un canal *AWGN*. Las variaciones del canal se dan con respecto a los diferentes escenarios que se seleccionan en función del SNR.

4.2.4. Recepción

En la Figura 4.5 se tiene los bloques usados para el proceso de recepción de datos y demodulación, como también la selección de la siguiente modulación con respecto al BER y SNR.

Receptor

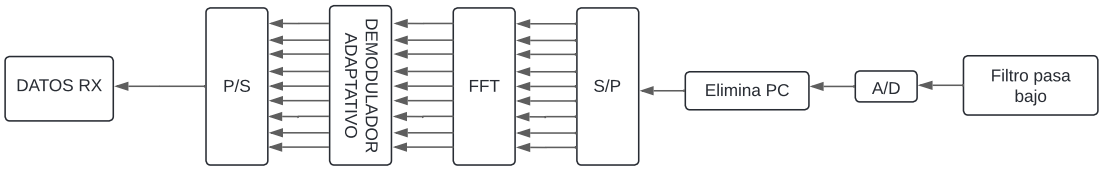

Figura 4.5: Bloques en el Receptor OFDM adaptativo

4.2.4.1. Eliminar Prefijo Cíclico

Para la corrección de ISI se añadió el Prefijo Cíclico en cada envío de señal, por lo que se debe eliminar y obtener solo los datos deseados y enviados por el transmisor.

4.2.4.2. Serie a Paralelo

Para poder generar la Fast Fourier Transform (FFT) es necesario tener los datos en paralelo, por lo que pasa de serie a paralelo para que la entrada al *FFT* sea correcta.

4.2.4.3. Transformada rapida de Fourier (FFT)

En la Ecuación (4.3) se realiza una transformación del dominio de la frecuencia al dominio del tiempo mediante la *FFT*, esto es necesario para poder realizar la estimación del canal y separación de los diferentes espectros con respecto a las subportadoras. Al igual que para la demodulación correcta de los datos.

$$X[k] = \sum_{n=0}^{N-1} x[n] * e^{-j\left(\frac{2\pi}{N}\right)*k*n} \tag{4.3}$$

4.2.4.4. Estimación del canal

Para obtener la estimación de la respuesta del canal h_{est} es necesario añadir subportadoras pilotos en la señal enviada por el transmisor, esto se distribuye en las diferentes subportadoras las cuales son evaluadas en el receptor, en este se conocen la amplitud de las subportadoras pilotos como las posiciones. Para la estimación del canal, se tiene la Ecuación (4.4) donde se

obtiene la relación de las subportadoras pilotos en el transmisor con el receptor.

$$h_{\text{est}} = \frac{1}{N_p} \sum_{i=1}^{N_p} \frac{y_p[i]}{p[i]} \tag{4.4}$$

donde:

- h_{est} relación de amplitudes entre las subportadoras pilotos transmitidas y recibidas.
- N_p número de subportadoras pilotos enviadas.
- $y_p[i]$ es la amplitud de la subportadora piloto en la posición i en el receptor.
- $p[i]$ es la amplitud de la piloto en la posición i en el transmisor.

En la Ecuación (4.5) se corrige la señal recibida con el canal estimado, para estimar el canal y encontrar el SNR.

$$y_{\text{corr}}[i] = \frac{y_{\text{data}}[i]}{h_{\text{est}}} \tag{4.5}$$

donde:

- $y_{corr}[i]$ es el dato corregido en la posición i de las subportadoras.
- $y_{data}[i]$ es el dato recibido en la posición i de las subportadoras.
- h_{est} es la estimación del canal.

Para calcular el SNR en el canal, es necesario calcular tanto la potencia de la señal enviada como la potencia del ruido. Ecuación (4.6) calcula la potencia de la señal $P_{\text{señal}}$.

$$P_{\text{señal}} = \frac{1}{N_p} \sum_{i=1}^{N_p} |y_p[i]|^2 \tag{4.6}$$

Ecuación (4.7) muestra el cálculo de la potencia del ruido $P_{\text{ruido_est}}$, y la Ecuación (4.8) muestra el cálculo el SNR en el canal.

$$P_{\text{ruido_est}} = \frac{1}{N_p} \sum_{i=1}^{N_p} |y_p[i] - h_{\text{est}} \cdot p[i]|^2 \tag{4.7}$$

$$\text{SNR}_{\text{est}} = 10 \log_{10} \left(\frac{P_{\text{señal}}}{P_{\text{ruido_est}}} \right) \tag{4.8}$$

4.2.4.5. Demodulador

Para la demodulación de la señal recibida se conoce la modulación que fue enviada en el transmisor debido al Feedback del receptor, estas igual al transmisor son las cinco opciones: QPSK, 8PSK, 16QAM, 32QAM y 64QAM.

4.2.4.6. Paralelo a serie

Para el análisis de los datos en el receptor y el cálculo del BER se procede a obtener el vector en serie de todos los datos enviados por OFDM con canal *AWGN*.

4.2.4.7. Calculo del Ber

En la parte del receptor se tiene los datos enviados desde el transmisor, por lo que en la Ecuación (4.9) se calcula el número de bits diferentes en el receptor con respecto al enviado del transmisor.

$$\text{BER} = \frac{N_{err}}{N_{bits}} \tag{4.9}$$

Donde:

- N_{err} número total de bits erróneos
- N_{bits} número total de bits transmitidos

4.2.4.8. Retroalimentación (Feedback)

La retroalimentación que se envía al transmisor es el índice de modulación con la que se va a mapear los datos del transmisor, este índice de modulación es enviado cada 1000 símbolos, para esto se tiene en cuenta que el BER deseado es de 0.001 por lo que al analizar el comportamiento de las diferentes modulaciones en la Figura 4.3, para tener el BER deseado y maximizar la cantidad de datos enviados al adaptar la modulación, para esto se tiene las condiciones de cada modulación con respecto al SNR del canal en la Tabla 4.2 [28].

Tabla 4.2: Selección de modulación con respecto al SNR

Modulación	SNR [dB] para alcanzar el BER=0.001
QPSK	<15
8PSK	15 - 18
16QAM	18 - 21
32QAM	21 - 26
64QAM	>26

La retroalimentación del receptor se da mediante un canal perfecto, no se considera para este proceso el canal de transmisión de OFDM.

4.3. Modelos de *Machine Learning*

Se definen tres modelos de aprendizaje automático: el modelo de regresión lineal, utilizado como técnica de Machine Learning (ML) en [42], la cual calcula los coeficientes de la función que modela los datos de entrenamiento, y dos modelos de clasificación: el Árbol de Decisión, elegido por su capacidad de proporcionar respuestas exactas de clasificación y K-Nearest Neighbor (KNN), seleccionado por su eficiencia en la clasificación basada en la proximidad de los datos.

Se utiliza la misma base datos para los tres modelos, en la cual se consideran diversas condiciones de canal (Signal to Noise Ratio (SNR)). La comparación de los resultados obtenidos de la implementación en el sistema, y los valores de las métricas de evaluación, ayudan a seleccionar el modelo más adecuado para el sistema. La Figura 4.6 muestra el esquema del flujo de trabajo utilizado para entrenar cada uno de los modelos y seleccionar el modelo más apto para el sistema.

Figura 4.6: Modelo de Trabajo

4.3.1. Proceso de entrenamiento

Para el proceso de entrenamiento, se generó una base de datos con tres parámetros del sistema: Bit Rate Error (BER), la modulación actual y la estimación del canal (SNR).

- **BER (Bit Error Rate) X1:** valor calculado en la transmisión en tiempo real, en nuestro modelo para el análisis se considera que se tiene los datos enviados en el Rx para el cálculo del BER.
- **Modulación actual X2:** modulación que tiene el sistema en tiempo real.
- **Estimación del SNR X3:**valor del SNR calculado a través de la estimación del canal.

El análisis del comportamiento de las cinco modulaciones es importante para definir los parámetros, por lo que se verifica entre que valores de SNR se tiene un BER apropiado para cada modulación, el BER deseado para el adaptativo con Machine Learnig es de 0.001 a un SNR menor a 20. En la Figura 4.7 se encuentra el comportamiento de cada modulación en el rango de BER de 0.001 y 0.0001.

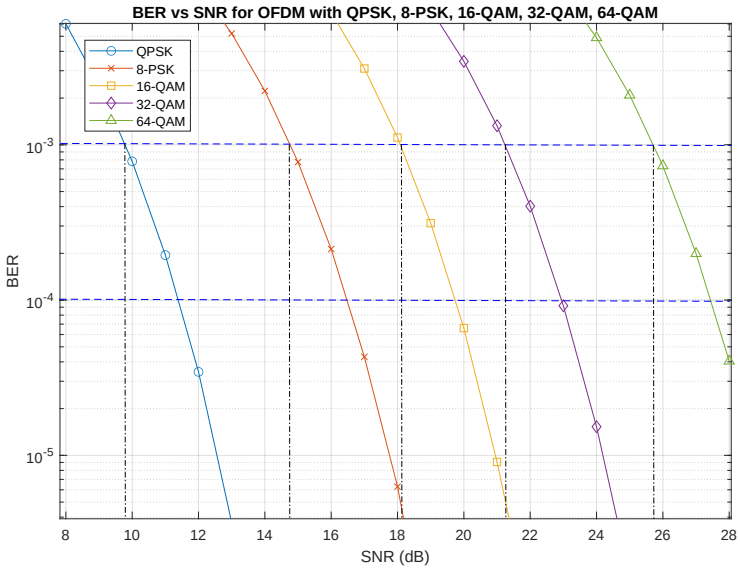

Figura 4.7: Gráfica del BER vs SNR en los rangos deseados

En la Tabla 4.3 se tiene los rangos de los SNR de cada modulación, para obtener el BER deseado, estos rangos se definen al analizar el comportamiento de las modulaciones en la Figura 4.7.

Tabla 4.3: Rangos de SNR para obtener un BER <0.001

Modulación	Rango de SNR [dB]
QPSK	>9
8PSK	>15
16QAM	>18
32QAM	>21
64QAM	>26

En la Tabla 4.4 se puede observar a los valores de BER bajo el valor deseado, estos están resaltados por los bloques de color azul, debido a que para cada modulación con un SNR se tiene un valor de BER diferente, se analizó el comportamiento de cada valor para cada modulación y se calcula el valor óptimo para el cambio de modulación ya sea a un índice de modulación menor o mayor en ese instante de tiempo, con este análisis se obtuvo que el límite para el cambio de modulación esta con un BER de 0.00005.

Tabla 4.4: Comportamiento del BER para las cinco diferentes modulaciones con diferente SNR

SNR_dB	QPSK	8-PSK	16-QAM	32-QAM	64-QAM
4	5.646741e-02	1.405628e-01	1.891346e-01	2.353709e-01	2.368796e-01
6	2.302562e-02	9.602745e-02	1.466954e-01	2.038355e-01	2.112800e-01
8	6.002210e-03	5.830477e-02	1.080938e-01	1.714371e-01	1.839845e-01
10	7.818990e-04	2.902304e-02	7.261984e-02	1.375689e-01	1.545561e-01
12	6.373990e-05	1.037977e-02	4.135985e-02	1.021759e-01	1.245273e-01
14	1.800000e-07	2.221319e-03	2.421863e-02	6.679741e-02	1.154818e-01
16	9.000000e-08	2.140400e-04	7.120522e-03	3.576966e-02	8.470973e-02
18	4.500000e-08	6.020000e-06	1.113705e-03	1.401346e-02	5.648228e-02
20	2.250000e-08	3.010000e-06	6.615990e-05	3.428595e-03	3.177008e-02
22	1.125000e-08	1.505000e-06	6.133320e-07	3.964790e-04	1.713054e-02
24	5.625000e-09	7.525000e-07	3.066660e-07	1.408000e-05	4.901052e-03
26	2.812500e-09	3.762500e-07	1.533330e-07	7.999990e-08	7.298070e-04
28	1.406250e-09	1.881250e-07	7.666650e-08	3.999995e-08	4.088980e-05

Para la selección de la siguiente modulación, se compara el BER con respecto a la modulación actual, en caso de que el canal estimado se encuentre cerca de los límites de decisión se considera cambiar la modulación para que el BER puede alcanzar un valor mayor al deseado como en la Tabla 4.4.

Para tener un cambio de modulación se debe tener en cuenta la modulación actual, esto es necesario en caso que el canal llegue a ser muy ruidoso, por lo que uno de los parámetros a considerar es la modulación actual, como también el BER obtenido y el SNR estimado, esto se visualiza en las Tablas 4.5.

Tabla 4.5: Selección de modulación con respecto al SNR de subida y modulación

Modulación actual	**Rangos de SNR para alcanzar el BER=**0.001	**Modulación siguiente con un BER**< 0.00005	**Modulación siguiente con un BER** ≥ 0.00005
QPSK	<15	8PSK	QPSK
8PSK	15 - 18	16QAM	QPSK
16QAM	18 - 21	32QAM	8PSK
32QAM	21 - 26	64QAM	16QAM
64QAM	>26	64QAM	32QAM

Se genera la base de datos, considerando los intervalos de cada variable y se establecen los escenarios para que la siguiente modulación sea seleccionada para obtener el BER objetivo. El proceso de entrenamiento se genera de manera automática en la APP de Matlab herramienta utilizada para implementar las técnicas de *Machine Learning*, ingresando los datos de entrenamiento.

El conjunto de parámetros establecido entrena los modelos planteados. Se definen los rangos, los cuales se presentan en la Tabla 4.6 y que son utilizados para manejar el sistema.

Tabla 4.6: Rangos de los Predictores

Predictor	**Rango**
X1 (BER)	[0.0005 - 0.0001]
X2 (Modulación)	QPSK, 8PSK, 16-QAM, 32-QAM, 64-QAM
X3 (SNR)	0 dB - 35 dB

Una vez establecidos los rangos y basándose en los valores teóricos, se desarrolla una estructura de reglas para generar los datos de entrenamiento. Mediante un script en Matlab se implementa el Algoritmo 1. Este proceso genera un total de 100,000 datos que conforman la base con la cual se lleva a cabo el entrenamiento automatizado que busca el mejor modelo.

4.3.2. Selección de la Técnica de *Machine Learning*

4.3.2.1. Modelo de Regresión Lineal

La implementación de la técnica de *Machine Learning* se realizó mediante la aplicación de Regression Learner del App de Matlab. El modelo se entrenó con la base de datos creada con

Algorithm 1 Algoritmo para generar la base de datos, basado en AMR

```
if BER < 0.0005 then
    if modulación = 4 then
        if SNR >= 15 then
            modulación = 8
        else if SNR <15 then modulación = 4
        end if
    else if modulación = 8 then
        if SNR >= 18 then
            modulación = 16
        else if SNR <18 then modulación = 4
        end if
        ...
    else if modulación = 64 then
        if SNR >= 26 then
            modulación = 64
        else if SNR <26 then modulación = 32
        end if
    else if BER >= 0.0005 then
        if modulación = 4 then modulación = 4
        else if modulación = 8 then
            if SNR <18 then modulación = 4
            else if SNR >= 18 then modulación = 8
            end if
            ...
        else if modulación = 64 then
            if SNR <26 then modulación = 32
            else if SNR >= 26 then modulación = 64
            end if
        end if
    end if
```

el script de Matlab desarrollado, considerando el modelo más robusto, que durante su entrenamiento no reduce las dimensiones de los datos, ni ajusta los parámetros de configuración.

La Figura 4.8 presenta la gráfica de las predicciones frente a los valores reales. Las lineas azules presenta los valores reales de los índices de modulación [QPSK (4) 8PSK (8) 16QAM (16) 32QAM (32) 64QAM (64)] y la gráfica amarilla los valores predichos, si bien se ajustan a las tendencias no coinciden en su totalidad con los valores reales.

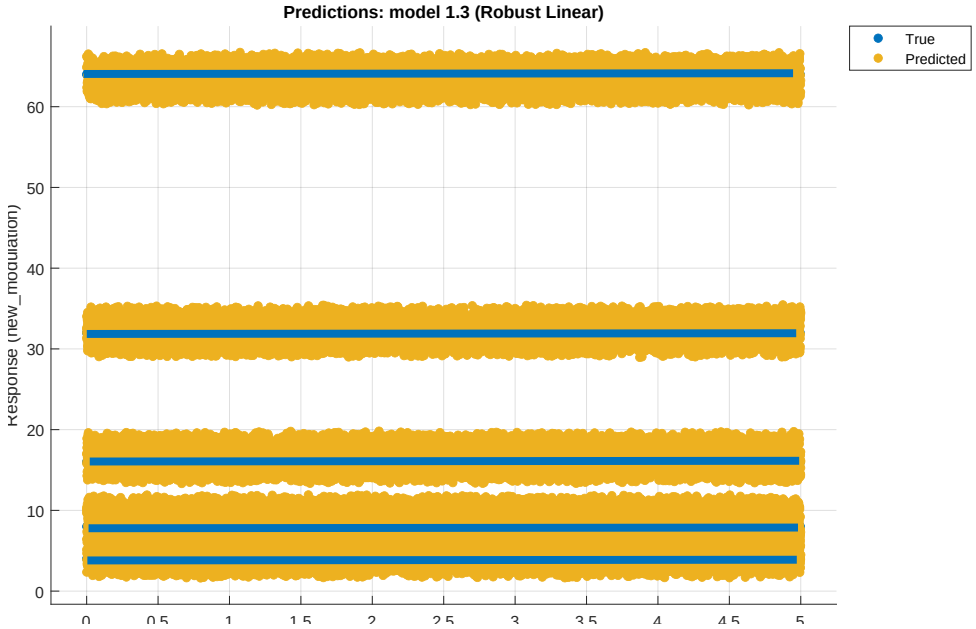

Figura 4.8: Gráfica de respuesta / Modelo de regresión Lineal

La Figura 4.9 presenta la gráfica de las predicciones ajustadas con respecto a los valores reales. La recta representa una predicción perfecta, para tener predicciones exactas los puntos deben alinearse a esta. Sin embargo, se observa una dispersión alrededor de la misma, indicando que el modelo tiene en ciertos casos errores de predicción. Si bien se captura la tendencia, las predicciones no son completamente precisas.

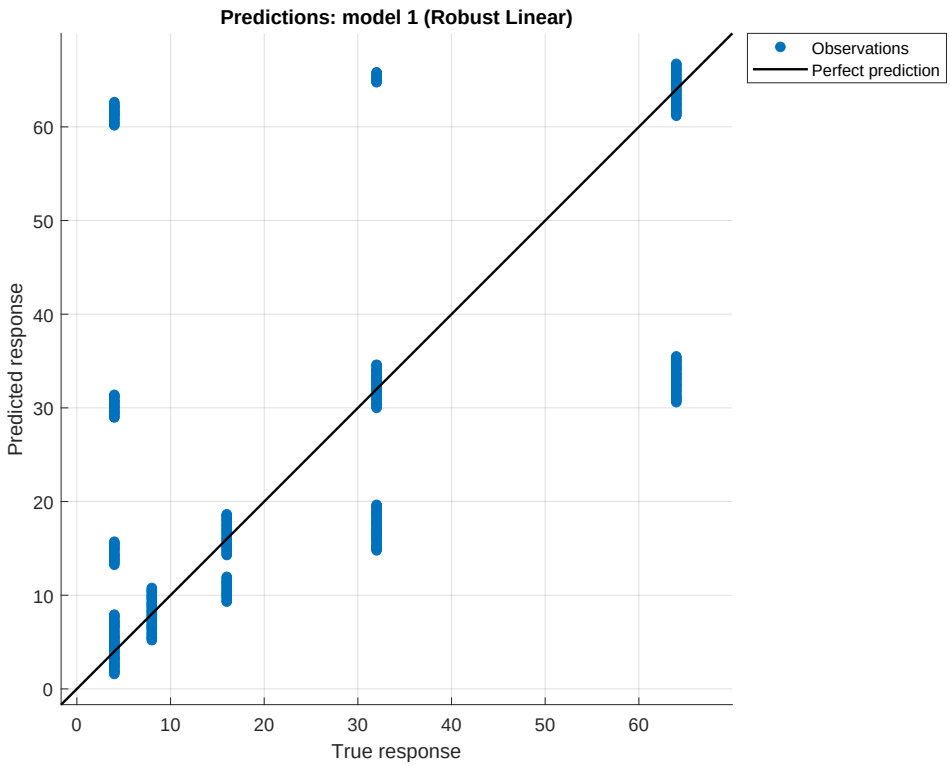

Figura 4.9: Gráfica de las predicciones vs valores reales / Modelo de regresión Lineal

La Figura 4.11 presenta la gráfica de las diferencias entre las predicciones y los datos reales. Se visualizan residuos tanto positivos como negativos, indicando que el modelo en ocasiones sobrestima o subestima los valores de respuesta. Los puntos más lejanos de la línea cero indican que le modelo presenta errores de predicción en los valores correspondiente a los niveles inferiores de modulación, dónde incluso puede generar valores negativos para el sistema OFDM.

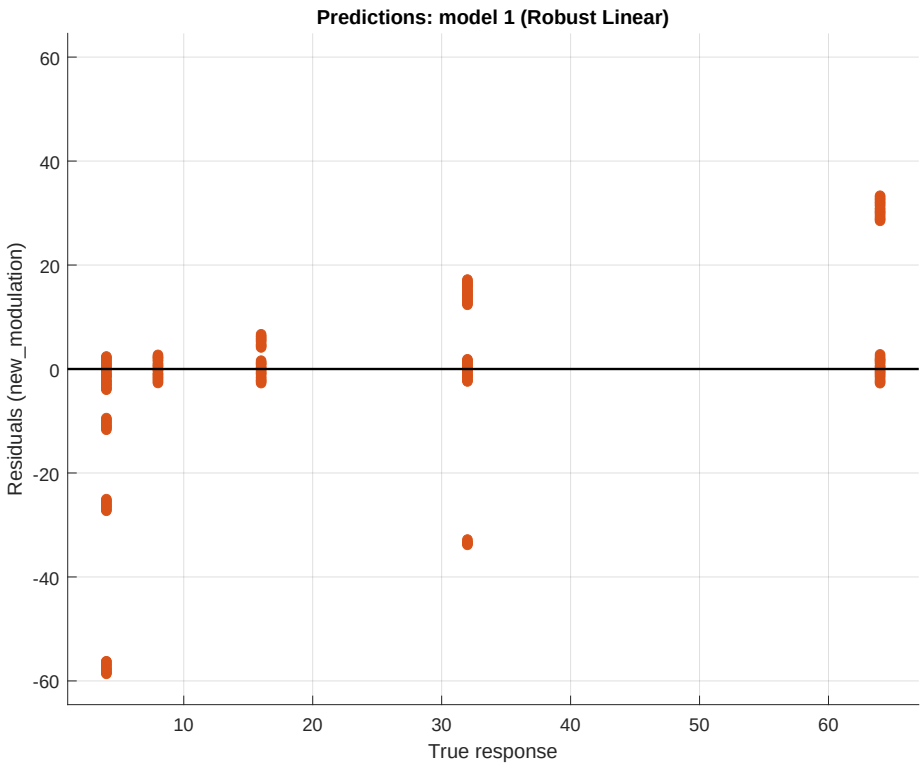

Figura 4.10: Gráfica de los residuos / Modelo de regresión Linea

Los valores predichos con este modelo no son precisos y exactos, en ciertos puntos los valores predichos son negativos, lo que requiere redondear los valores para tener un valor exacto del índice de modulación. Se calcula la distancia entre las cinco modulaciones [QPSK (4) 8PSK (8) 16QAM (16) 32QAM (32) 64QAM (64)] y se selecciona el valor más cercano a la predicción realizada. En la Figura 4.11 se muestra la gráfica del BER vs SNR que disminuye al aumentar el valor del SNR, que muestra que tiene un resultado favorable ya implementado en el sistema. Se elige el modelo de *Robust Linear* ya que en las pruebas sus predicciones son valores más cercanos a los reales y se pueden redondear y tener un dato más exacto.

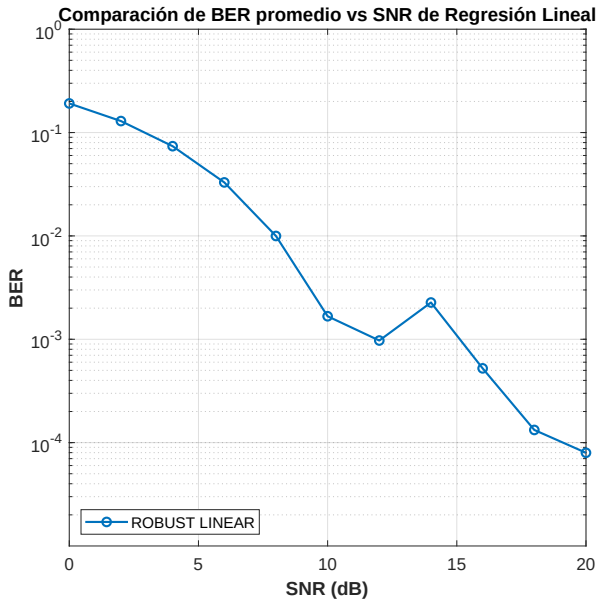

Figura 4.11: Gráfica del BER vs SNR del modelo de Regresión Lineal

4.3.2.2. Modelo de Árbol de Regresión / Clasificación

Estos modelos manejan la misma estructura de árbol que se muestra en la Figura 4.12, que se compone del nodo raíz que representa al conjunto de datos, seguido de los nodos de decisión con los cuales se realizan la divisiones de los datos, se observa que utiliza un parámetro para la decisión [BER, SNR, Índice de Modulación actual] y los nodos terminales (hojas) que representan las predicciones o el grupo de clasificación final. Se entrenan cada uno de los modelos y mediante la comparación de sus datos de validación, y la gráfica de BER vs SNR se determina que modelo es óptimo para la implementación en el sistema OFDM general.

El modelo del árbol de decisión se crea en base a la siguiente estructura, que se conforma de las decisiones que la herramienta App Matlab plantea para la configuración del algoritmo (Anexo 1), estas reglas son generadas por el análisis que realiza la técnica sobre los datos, mientras que las reglas propuestas son utilizadas para generar la base de datos.

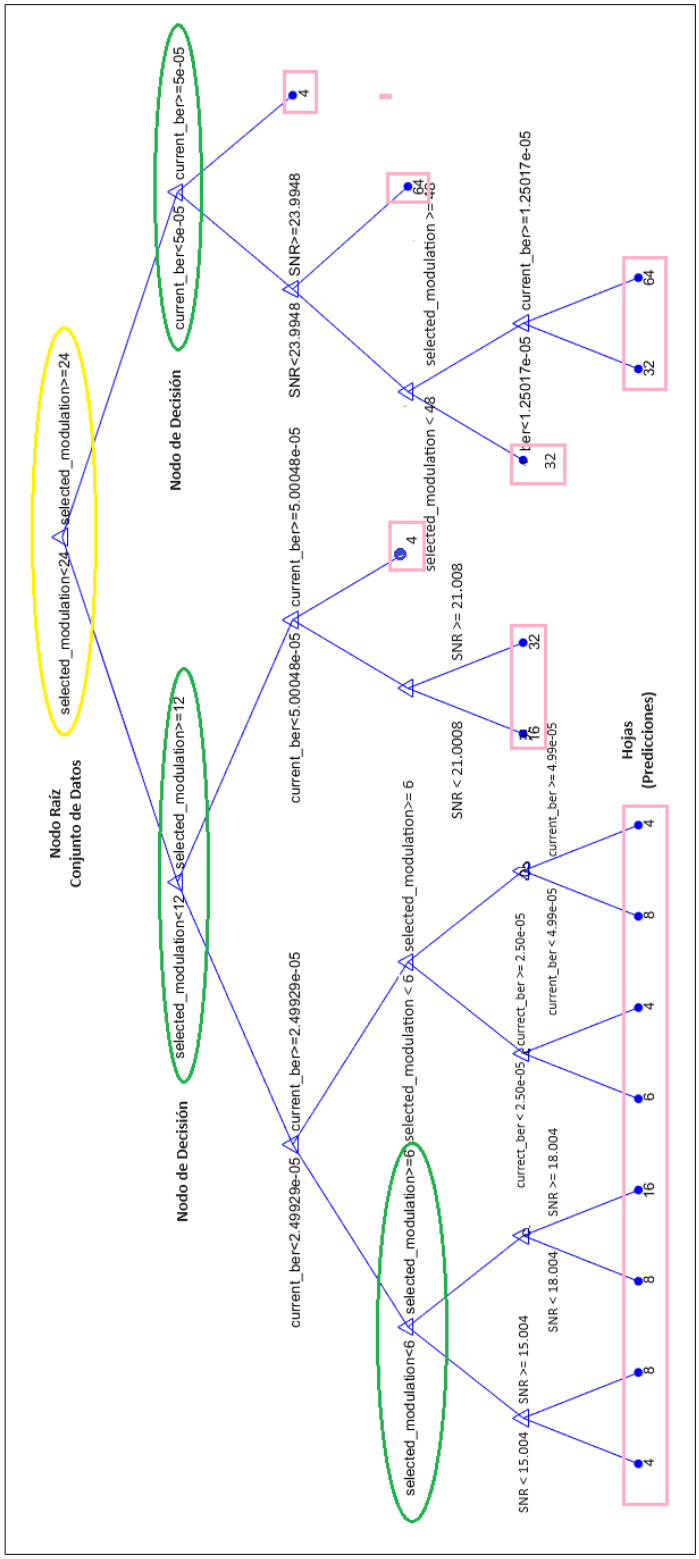

Figura 4.12: Estructura del Árbol de Clasificación

Tipos de Árbol de Regresión

- ***Fine Tree:*** se caracteriza por el tamaño de hoja de 4 que representa el valor mínimo, esto le permite tener más nodos terminales, caracterizado por capturar detalles finos y patrones complejos en los datos de entrenamiento, maneja una alta precisión en el conjunto de entrenamiento, llegando a presentar sobre ajuste de los datos.
- ***Medium Tree:*** tiene un tamaño de hoja de 12 que representa un valor intermedio, su estructura es moderada, busca detalles en el análisis de los datos y simplicidad a la vez, genera un estructura más simplificada que el modelo de *Fine Tree*.
- ***Coarse Tree:*** tiene un tamaño de hoja de 36 que representa un valor grande, por lo que maneja un menor número de nodos terminales, y su estructura es más simple y generalizada, maneja una menor precisión en comparación que los modelos previos.

Las Figuras 4.13, 4.14 y 4.15 muestran las gráficas de las predicciones de los tres modelos, la Figura 4.13 muestra la gráfica del modelo *Fine Tree*, donde las predicciones se alinean con los valores reales de la modulación [QPSK (4) 8PSK (8) 16QAM (16) 32QAM (32) 64QAM (64)]. En los niveles de 4 y 8 se observa una cierta variabilidad en los datos. En la Figura 4.14 y 4.15 se muestra la gráfica del modelo *Medium Tree* y *Coarse Tree* respectivamente, siguen la tendencia del primer modelo, pero con una menor variabilidad de los datos en los índices de las modulación más robustas QPSK (4) y 8PSK (8).

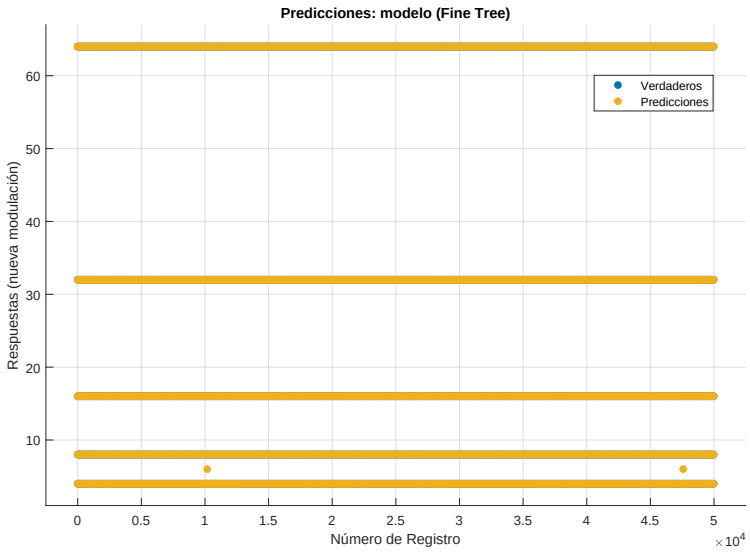

Figura 4.13: Gráficas de la respuesta de las predicciones reales del modelo *Fine Tree*

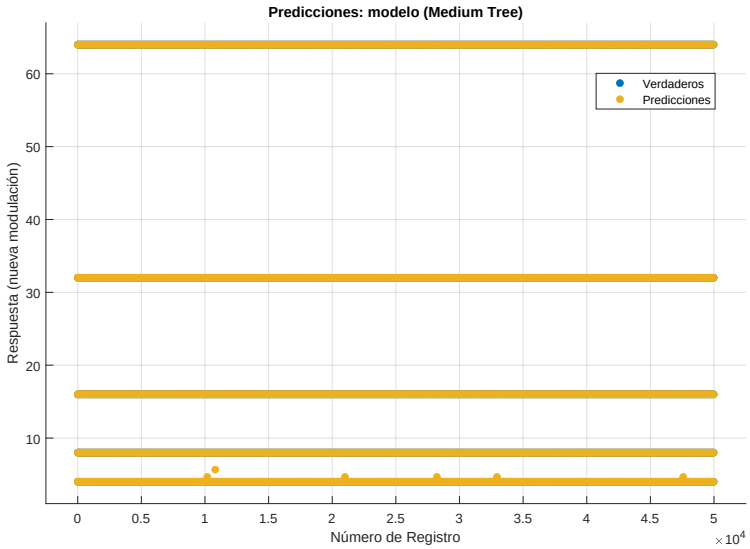

Figura 4.14: Gráficas de la respuesta de las predicciones reales del modelo *Medium Tree*

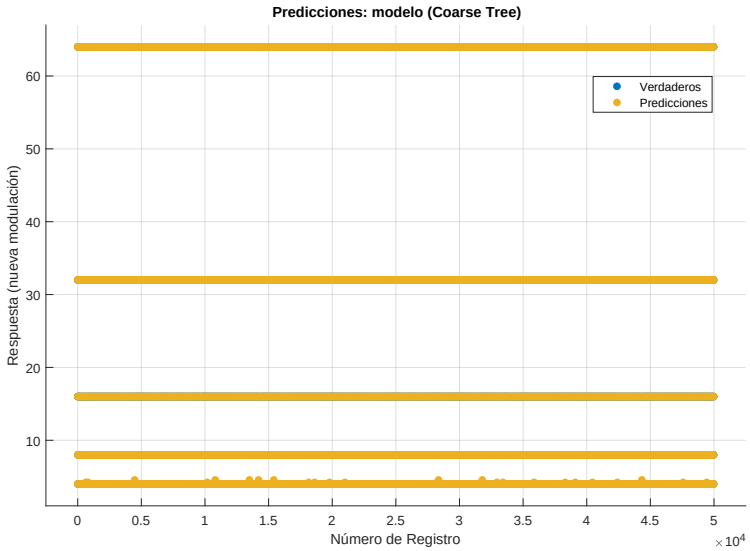

Figura 4.15: Gráficas de la respuesta de las predicciones reales del modelo *Coarse Tree*

Las Figuras 4.16, 4.17 y 4.18 muestran las gráficas de las predicciones ajustadas con respecto a valores reales. Se observa que tanto los valores reales (linea azul) como los predichos (linea amarilla), coinciden por lo que no se puede observar la linea azul. Todas las gráficas tienen resultados semejantes, con variabilidad en las modulaciones más robustas QPSK y 8PSK, por la dispersión de los datos que se observan alrededor de la recta amarilla en los índices de la nueva modulación [4 8].

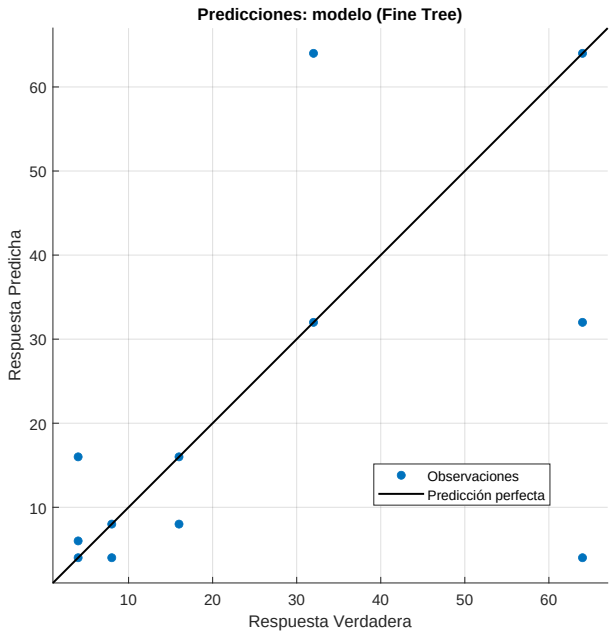

Figura 4.16: Gráficas de las predicciones ajustadas con respecto a valores reales modelo *Fine Tree*

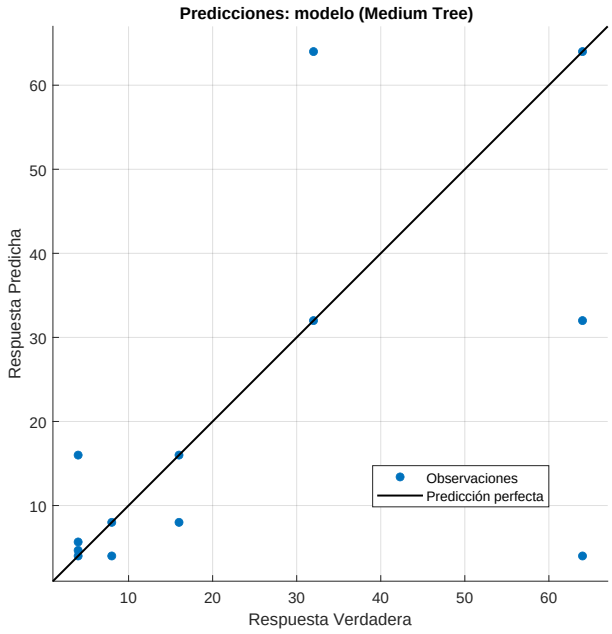

Figura 4.17: Gráficas de las predicciones ajustadas con respecto a valores reales modelo *Medium Tree*

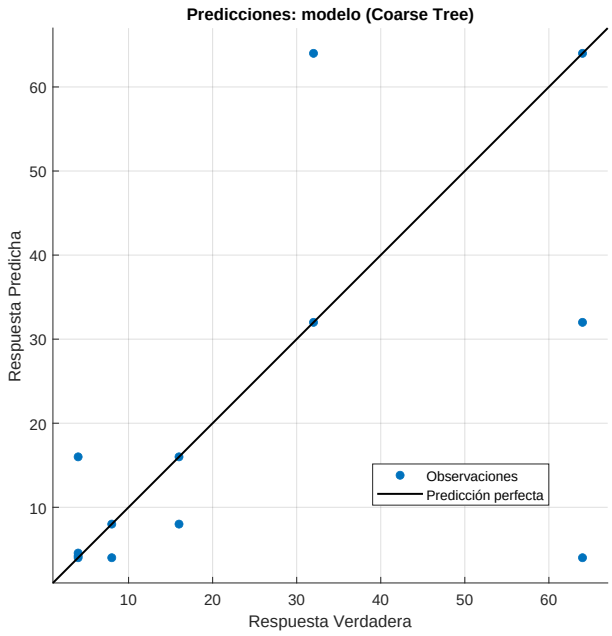

Figura 4.18: Gráficas de las predicciones ajustadas con respecto a valores reales modelo *Coarse Tree*

Las Figuras 4.19, 4.20 y 4.21 muestran las gráficas de los residuos. En los tres modelos, los resultados son similares: las predicciones se alinean bien con los valores reales, aunque se observa una dispersión en los valores más bajos, especialmente en el nivel 4.

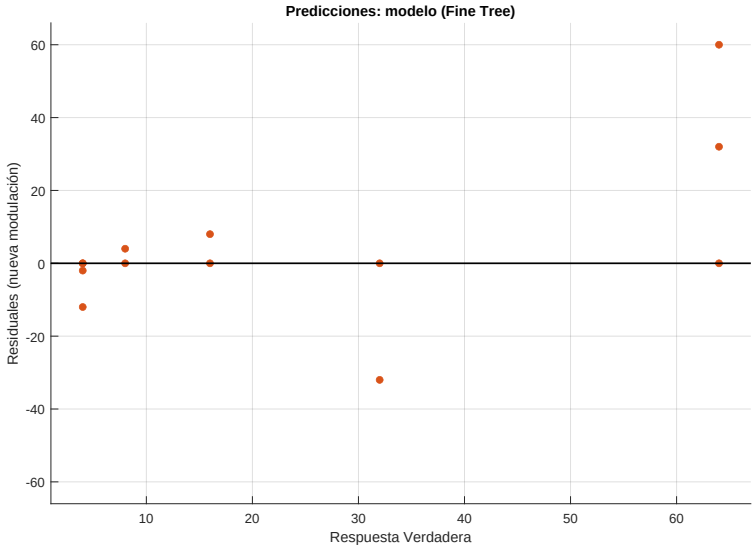

Figura 4.19: Gráficas de los residuos del modelo *Fine Tree*

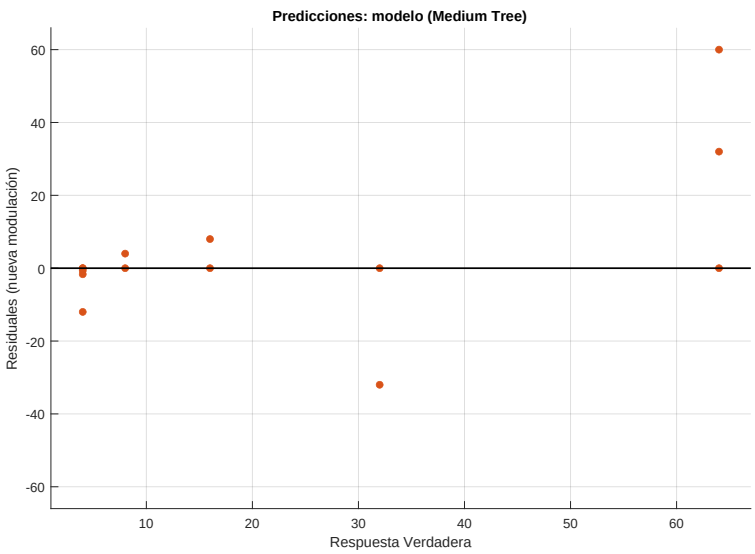

Figura 4.20: Gráficas de los residuos del modelo *Medium Tree*

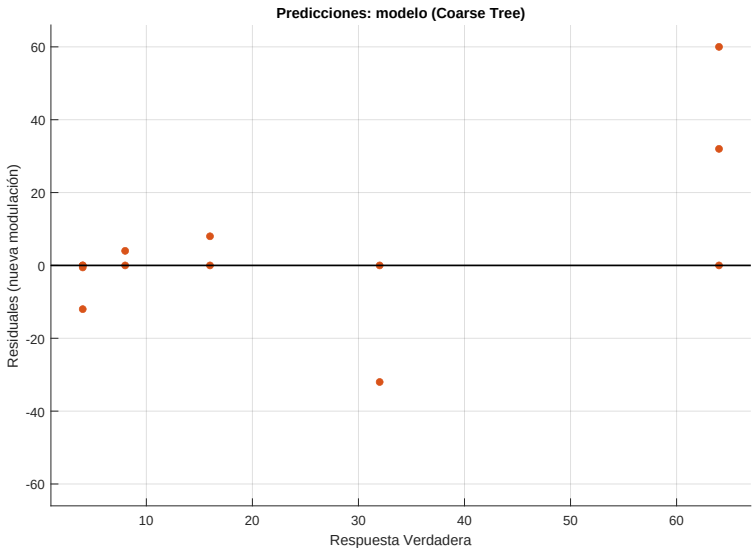

Figura 4.21: Gráficas de los residuos del modelo *Coarse Tree*

En la Figura 4.22 muestra la gráfica del BER vs SNR para cada Árbol de Regresión. Las curvas disminuyen con el aumento del SNR en cada caso, indicando un mejor rendimiento y calidad de la señal en los tres modelos. Los tres modelos tienen el mismo comportamiento, lo que indican que se ajustan perfectamente al modelo del sistema OFDM. El análisis de las métricas de evaluación nos muestra que los modelos son similares y presentan predicciones semejantes al valor real con un mínimo error, con estos resultados, se selecciona el modelo *Fine Tree*, ya que considera detalles más finos entres clases.

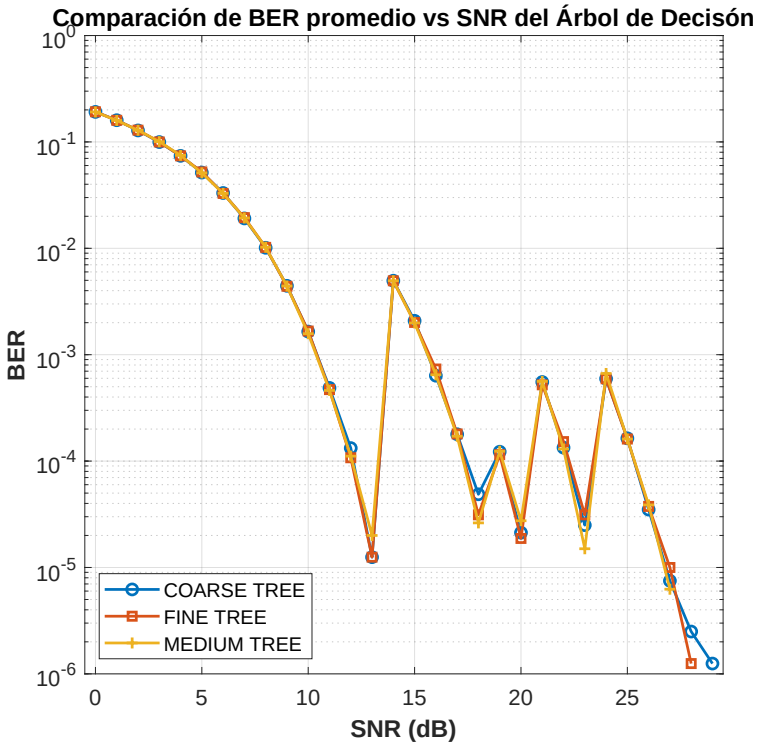

Figura 4.22: BER vs SNR | Comparación de los tres modelos

4.3.2.3. Modelo de KNN

La aplicación de Matlab se utilizó para entrena el modelo KNN con la base de datos creada. Se presentaron diferentes tipos de clasificadores que tienen una excelente precisión en dimensiones bajas (número de parámetros y vecinos), con un alto uso de memoria y de fácil interpretación. La flexibilidad de cada clasificador se maneja en función del número de vecinos que se configure, se evalúan los siguientes modelos de esta técnica.

- **Fine KNN:** El número de vecinos se establece en 1, con un alto grado de precisión por las características detalladas entre clases,llega a presentar sensibilidad al ruido y sobreajuste a los datos de entrenamiento.
- **Medium KNN:** El número de vecinos se establece en 10, con un grado medio de precisión por las características medias entre clases. Modelo más robusto. Tiene menor velocidad de precisión que el modelo *Fine KNN*, con un costo de validación mayor.
- **Coarse KNN:** El número de vecinos se establece en 100, mayor estabilidad en las predicciones, al promediar un mayor número de vecinos. Presenta una menor velocidad de procesamiento en comparación con modelos anteriores.

Se comparan los modelos mediante la matriz de confusión de validación, dónde se establece los porcentajes de observaciones, en conjunto con el porcentaje de:

- **True Positive Rates (TPR):** proporción de los verdaderos positivos (TP) sobre el total de positivos reales (P).
- **False Negatives Rates (FNR):**proporción de los falsos negativos (FN) sobre el total de positivos reales (P).
- **Positive Predictive Values (PPV):**es la proporción de verdaderos positivos (TP) sobre el total de positivos predecidos (PP)
- **False Discovery Rates (FDR):** proporción de falsos positivos (FP) sobre el total de positivos predichos (PP).

Modelo *Fine KNN*

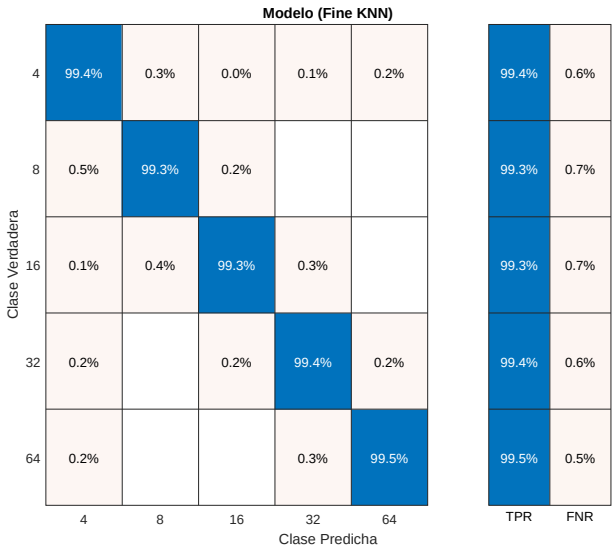

Figura 4.23: Matriz de Validación | Modelo *Fine KNN* | Métricas TPR y FNR

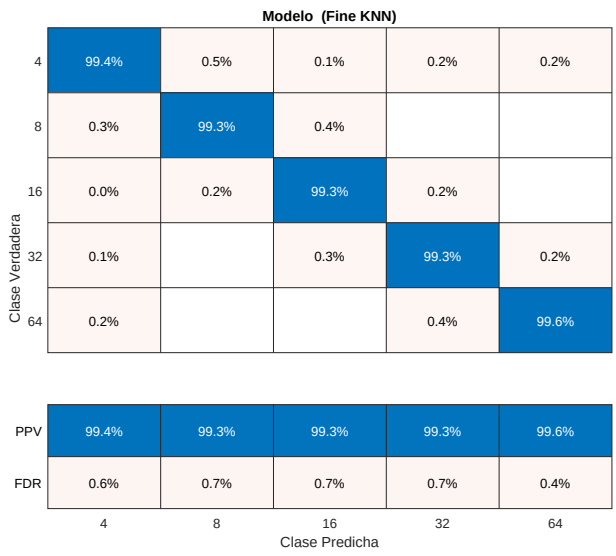

Figura 4.24: Matriz de Validación | Modelo *Fine KNN* | Métricas PPV y FDR

Modelo *Medium KNN*

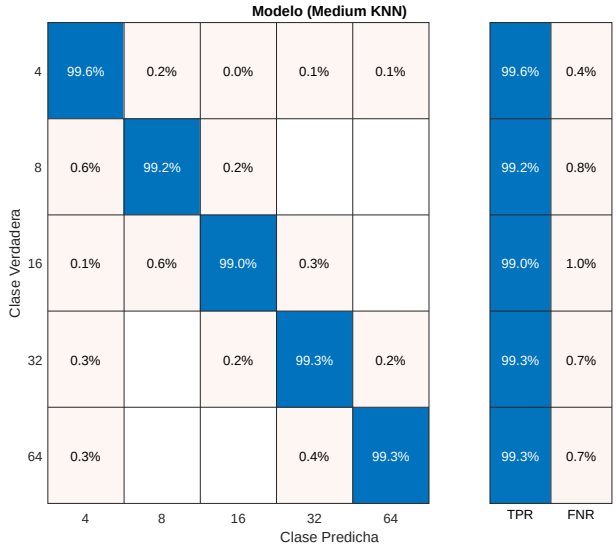

Figura 4.25: Matriz de Validación | Modelo *Medium KNN* | Métricas TPR y FNR

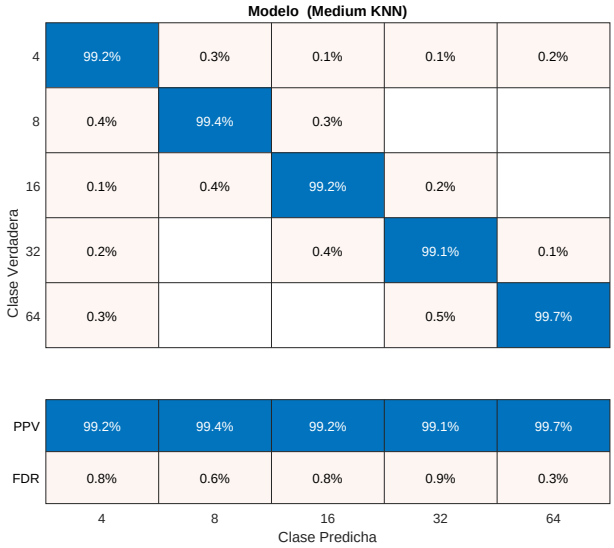

Figura 4.26: Matriz de Validación | Modelo *Medium KNN* | Métricas PPV y FDR

Modelo *Coarse KNN*

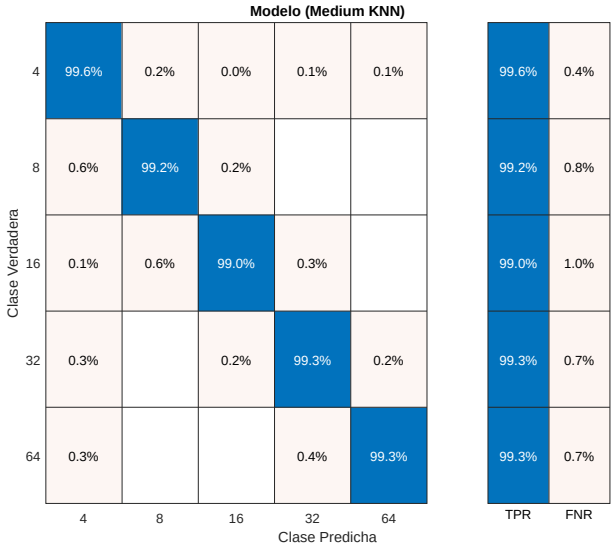

Figura 4.27: Matriz de Validación | Modelo *Coarse KNN* | Métricas TPR y FNR

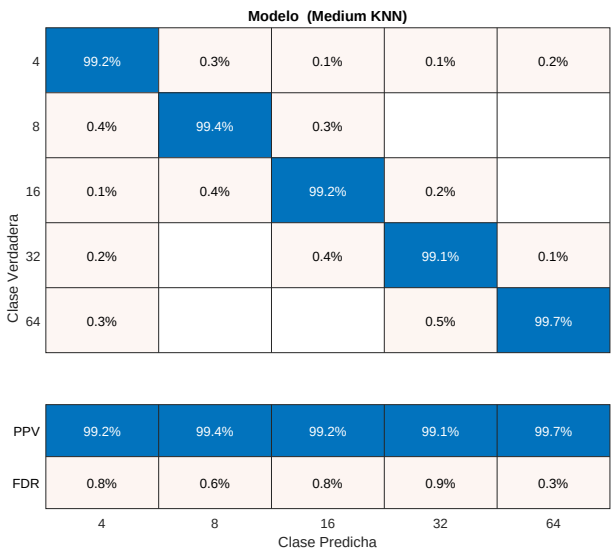

Figura 4.28: Matriz de Validación | Modelo *Coarse KNN* | Métricas PPV y FDR

En las Figuras 4.23, 4.24, 4.25, 4.26, 4.27, 4.28 muestran las matrices de validación de los tres modelos: *Fine KNN, Medium KNN* y *Coarse KNN*. En conjunto muestran un alto rendimiento de clasificación con un alto porcentaje de validación 99.7 %, 99.6 % y 99.6 % respectivamente. En todos los modelos las tasas de verdaderos positivos (TPR) y el valor positivo (PPV) son altas para cada una de las clases [4 8 16 32 64] superando el 99 %, mientras que para los falsos negativos (FNR) y las tasas de descubrimientos falsos (FDR) se mantiene por debajo del 0.6 %, 0.5 % y 0.8 % respectivamente. Esto nos indica que, para el tipo de predicción requerido, se convierten en modelos precisos y fiables, con una alta capacidad de clasificación y minimización de errores.

En la Figura 4.29 se muestra la gráfica del BER vs SNR para cada modelo. Las curvas disminuyen con el aumento del SNR en cada caso, indicando un mejor rendimiento y calidad de la señal. Sin embrago, el modelo *Fine KNN* tiene la mejor curva en términos del BER, siendo mejor que los modelos de *Medium KNN* y *Coarse KNN*, y en conjunto con las matrices de validación similares en las métricas de evaluación para cada modelo, se selecciona el modelo de *Fine KNN* para la implementación dentro del sistema general por el rendimiento y características que presenta.

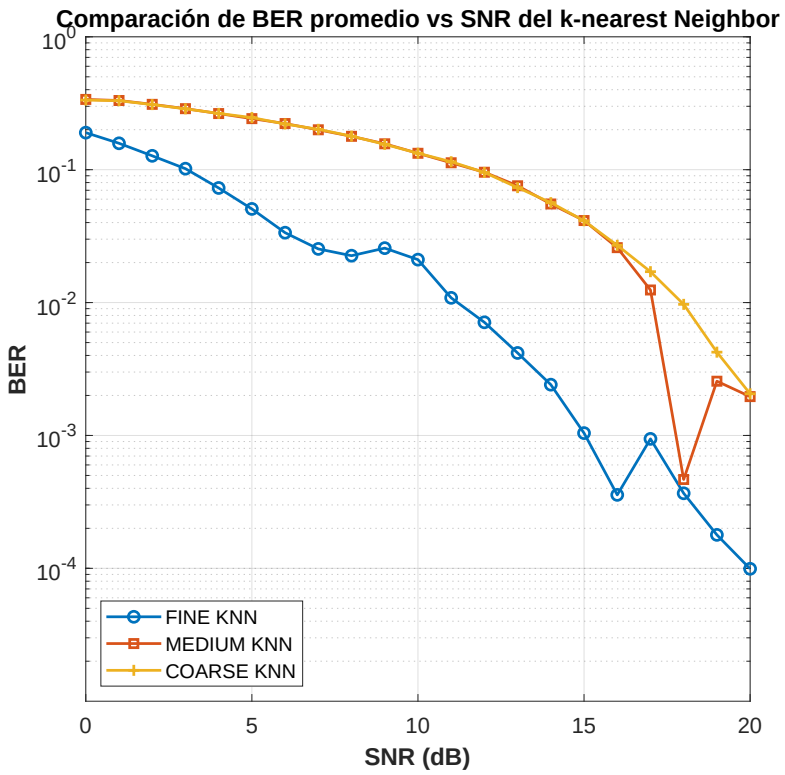

Figura 4.29: BER vs SNR | Comparación de los tres modelos

4.3.3. Comparación de los tres modelos

Una vez entrenados los tres modelos seleccionados, se evalúa el porcentaje de precisión, basándonos en las métricas de evaluación obtenidas del entrenamiento de los modelos. En la Tabla 4.7 se muestra los valores del porcentaje de precisión y otros datos de evaluación de los modelos de regresión lineal y árbol de decisión, y en la Tabla 4.8 se muestran las métricas para el modelo de KNN. Obtenidos a través de los *Training Results* determinados sobre el conjunto de validación, las siguientes métricas de análisis son:

- **RSEM (Root Mean Squared Error – Error Cuadrático medio de la Raiz):** Mide la diferencia promedio entre las predicciones y los valores reales. Un valor bajo es indicativo de un mejor ajuste del modelo a los datos.
- **R-Squared:** Mide el ajuste de las variables de respuesta que puede explicarse por las predicciones del modelo. Un valor cercano a 1 indica un buen ajuste.
- **Mean Squared Error (MSE) (Error Cuadrático Medio):** Utilizada para evaluar la precisión del modelo, un valor bajo indica un mejor desempeño del modelo.

- **Mean Absolute Error (MAE) (Error Absoluto Medio):** media de las predicciones incorrectas, un valor bajo indica una buena precisión.
- **Prediction Speed:** Indica la cantidad de observaciones que el modelo puede predecir por segundo. Esta métrica es crucial para aplicaciones en tiempo real donde la rapidez de la predicción es fundamental.

Tabla 4.7: Comparación de Métricas | Modelo de Regresión Lineal y Arból de Decisión

Métrica	Regresión Lineal	Árbol de Decisión
RSME	16.562	0.06572
R-Squared	0.51	1
MSE	274.3	0.00432
MAE	9.031	0.0048
Prediction Speed	930000 obs/sec	1100000 obs/sec

- **Accuracy (Validation):** especifica la proporción de predicciones correctas entre el total de predicciones realizadas.

Tabla 4.8: Comparación de Métricas | de *K-nearest neighbor*

Métrica	*K-nearest neighbor (Fine KNN)*
Accuracy (Validation)	99.6 %
Total cost (Validation)	412
Prediction spedd	1000000 obs/sec
Training time	6.76 sec

Las Tablas 4.7 y 4.8 presenta las métricas utilizadas para evaluar el rendimiento de los tres modelos para la misma base de datos. El modelo de regresión lineal robusta muestra un valor alto de RSME, lo que indica un ajuste menos preciso del modelo a los datos. La métrica de *R-Squared* indica que el modelo no captura adecuadamente la variabilidad en los datos, y de la misma manera los valores elevados de MSE y MAE reflejan un peor rendimiento y ajuste. En contraste, el modelo del árbol de decisión (*Fine Tree*) tiene valores significativamente bajos RSME, MSE y MAE, lo que indica un mejor ajuste y rendimiento. Además, el valor de *R-Square* es de 1 lo que sugiere un ajuste perfecto. En la Tabla 4.8 se establece la métricas de modelo *K-nearest neighbor* lo que nos indica un rendimiento alto con una precisión del 99.6 %. Generando predicciones con mayor exactitud lo que se requiere en nuestro sistema OFDM.

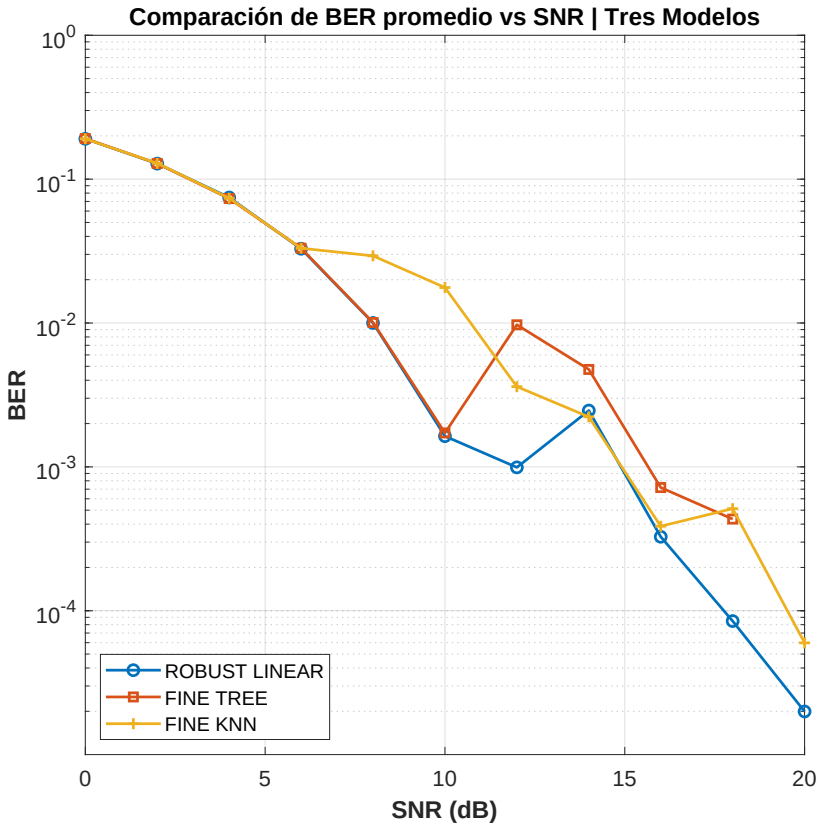

Figura 4.30: BER vs SNR | Comparación de los tres modelos

La Figura 4.30 muestra la gráfica de BER vs SNR y expone el desempeño de los tres modelos seleccionados de las diferentes técnicas. Como se observa las curvas disminuyen conforme aumenta el valor del SNR. El modelo de *Robust Linear* tiene un mejor resultado ya que los valores del BER son menores en comparación con los otros modelos. Mientras que el modelo de *Fine Tree* en similar al modelo de *Robust Linear* en el primer rango de SNR y tiene valores mayores de BER en valores altos de SNR. El modelo de *Fine KNN* tiene los valores intermedios BER entre los modelos de *Fine Tree* y *Robust Linear*. Lo que nos indica la gráfica es que el modelo más adecuado por su resultado es el de *Robust Linear*.

Para seleccionar el modelo se han definido los siguientes parámetros:

- Precisión y exactitud en las predicciones de los esquemás de modulación.
- Métricas de evaluación.
- Análisis de la gráfica BER vs SNR.

***ROBUST LINEAR*:** las predicciones de este modelo no son exactas, no genera valores enteros como [4 8 16 32 64] que representan a los índices de modulación e incluso se dan salidas con valores negativos como [-0.23, -1.2], lo que requiere de un procesamiento de redondeo para

asignar a una de las clases de modulación. Sus métricas de evaluación son bajas, indicando que el modelo no es el más adecuado. Sin embargo, en la gráfica BER vs SNR presenta los valores de BER más bajos en comparación con otros modelos.

***FINE TREE* y *FINE KNN*:** estos modelos generan predicciones exactas y sus métricas de evaluación muestran una alta precisión y exactitud. En la gráfica BER vs SNR, el modelo de *Fine Tree* tiene valores similares al modelo de regresión lineal y en el rango [10 dB a 20 dB] de SNR presenta valores más altos, por otra parte, el modelo de *FINE KNN* tiene valores intermedios de BER.

Si bien los modelos óptimos no tienen los mejores resultados en la gráfica de BER vs SNR, predicen mejor los cambios de modulación. A diferencia del modelo de *Robust Linear* cuyos valores no son precisos. Por este análisis, se determina implementar los tres modelos en el sistema OFDM para la adaptabilidad de la modulación y evaluar su rendimiento.

4.4. Implementación del modelo en OFDM Adaptativo

Para la implementación del sistema OFDM adaptativo, se presenta el esquema de la Figura 4.31, que detalla cómo se integra la técnica de *Machine Learning* para determinar la modulación más efectiva según las condiciones del canal. En el esquema se tiene tres bloques principales: Transmisión (TX), Recepción (RX) y el bloque adaptativo.

Bloque Adaptativo: este bloque es la base del sistema propuesto de implementación, utiliza los modelos de *Machine Learning* seleccionados para ajustar la modulación en tiempo real, basándose en los tres parámetros:

- **BER (Bit Error Rate) – X1:** se obtiene comparando los datos de TX y RX y aplicando los conceptos relacionados.
- **Modulación actual – X2:** representa la técnica actual que se está manejando en el sistema, se actualiza dinámicamente al cambiar la modulación en el bloque del RX.
- **Estimación del canal (SNR) – X3:** a partir de la estimación del canal, se refleja el SNR estimado en el canal.

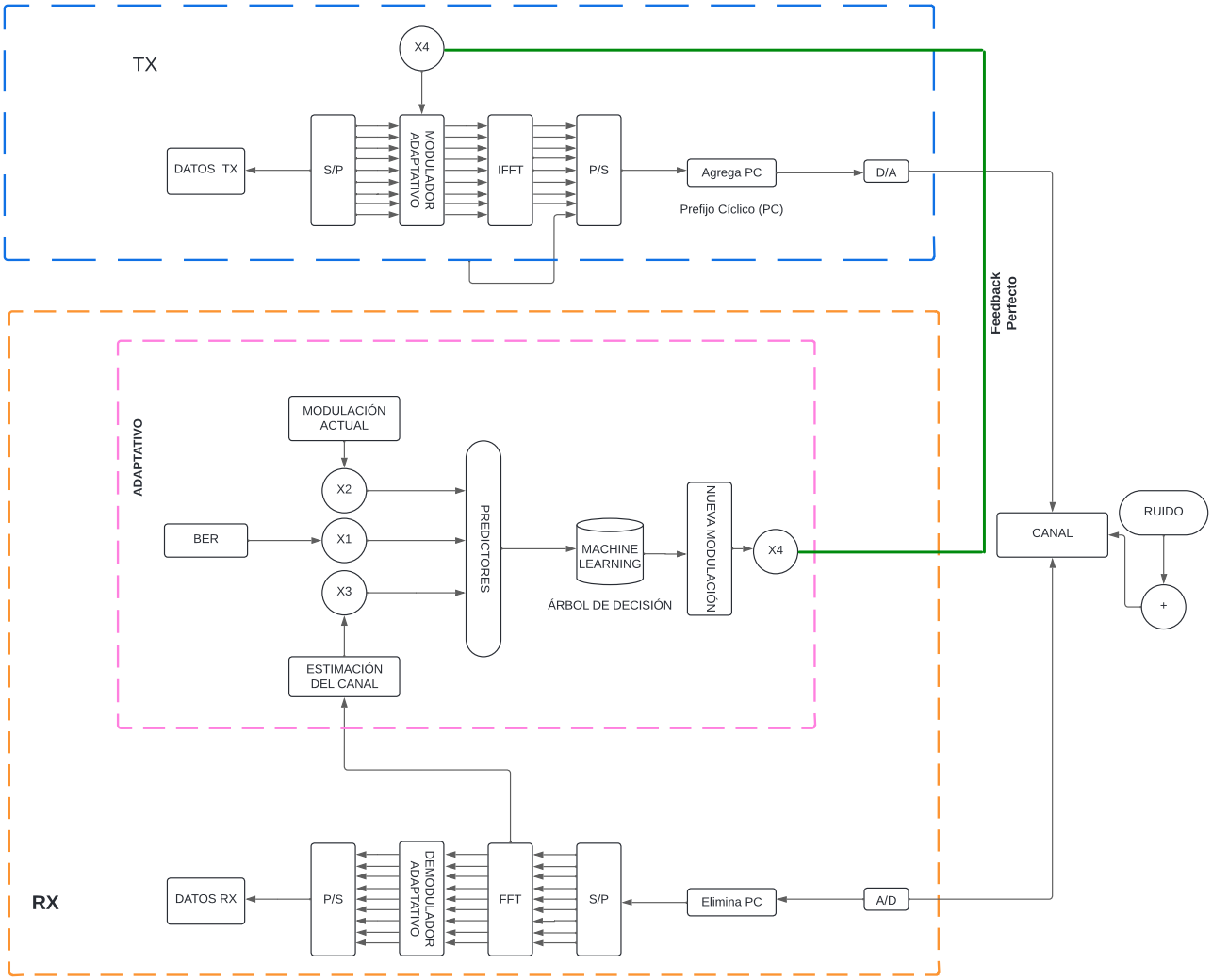

Figura 4.31: Esquema de implementación de *Machine Learning* en el sistema de OFDM

El proceso de ajuste de la modulación se lleva a cabo mediante la obtención de los predictores calculados por el sistema, que se utilizan como entradas de los modelos. Se emplean los valores de entrada [X1 (BER) X2 (Modulación Actual) X3 (Estimación del Canal)], para determina el mejor índice de modulación que minimiza el BER y optimiza la calidad de la transmisión. La nueva modulación se retro-alimenta al transmisor, ajustando de esta manera el índice de modulación en función de las condiciones del canal.

El feedback continuo entre los bloques de recepción (RX) y transmisión (TX) permite que el sistema OFDM se adapte dinámicamente a las condiciones cambiantes del canal. Logrando mantener un rendimiento óptimo en entornos con alta variabilidad como el canal AWGN que se establece para este modelo.

4.5. Escenarios de Prueba

Para el análisis del comportamiento del OFDM adaptativo convencional y con *ML* se han planteado 3 diferentes escenarios mediante el uso del modelo de canal *AWGN*, que consisten en variar el SNR del canal.

- **Escenario 1:** se emplea un canal *AWGN*, que varía el parámetro SNR, este valor va aumentando progresivamente dentro del rango [0 dB - 30 dB], los parámetros del sistema OFDM se presentan en la Tabla 4.3.
- **Escenario 2:** se emplea un canal *AWGN*, en el que se varía el parámetro del SNR, dentro de un rango de [0 dB - 30 dB] con una variación de $\pm$ 2 [dB].
- **Escenario 3:** en este se emplea un canal en el cual el parámetro SNR es totalmente aleatorio, el rango del parámetro esta entre [0 dB - 30 dB], y se considera un caso extremo para analizar el comportamiento del modelo convencional, y de las técnicas de *Machine Learning*.

5. Resultados y Discusión

Los resultados de las pruebas son una parte fundamental de la fase experimental. Se plantearon tres escenarios en los cuales se varían las condiciones del canal mediante la variación del Signal to Noise Ratio (SNR), para evaluar el comportamiento del sistema OFDM adaptativo aplicando dos enfoques, el convencional y el basado en técnicas de *Machine Learning*.

Los parámetros que se necesitan analizar para la selección de la siguiente modulación, es la estimación del canal, debido a los rangos de cada modulación, en la Tabla 4.3, se tienen los resultados de la estimación del SNR del canal con respecto al SNR enviado en el canal mediante la potencia del ruido y la de la señal de entrada.

En la Tabla 5.1 se tiene que para los valores de SNR bajos, la estimación del SNR del canal en la Ecuación (4.8), es cinco veces más del SNR enviado en el canal, como también en los valores de SNR altos tiene un error de hasta 3 [dB].

Tabla 5.1: Comparación del SNR enviado al canal con el SNR estimado

SNR enviado	SNR estimado
0	4.643632
3	6.576318
6	8.647419
9	11.452227
12	14.203211
15	17.139850
18	19.292382
21	21.049272
24	22.311355
27	23.382388

5.1. Evaluación del rendimiento del sistema OFDM Adaptativo con respecto a OFDM Adaptativo con ML

Escenario 1

Para la evaluación se considera los resultados del BER y del *Throughput* con respecto a la variación del SNR. Los valores del BER y *Throughput* son comparados tanto para el sistema OFDM convencional y empleando técnicas de ML. Tomando en cuenta que el valor del BER objetivo es por debajo de 0.001.

Se presentan las gráficas del comportamiento del BER, del *Throughput* y del cambio de modulación de los cuatro casos planteados, al final de la sección se realiza una discusión con base en la comparación del sistema OFDM Adaptativo con respecto a OFDM Adaptativo con ML.

Comportamiento del BER

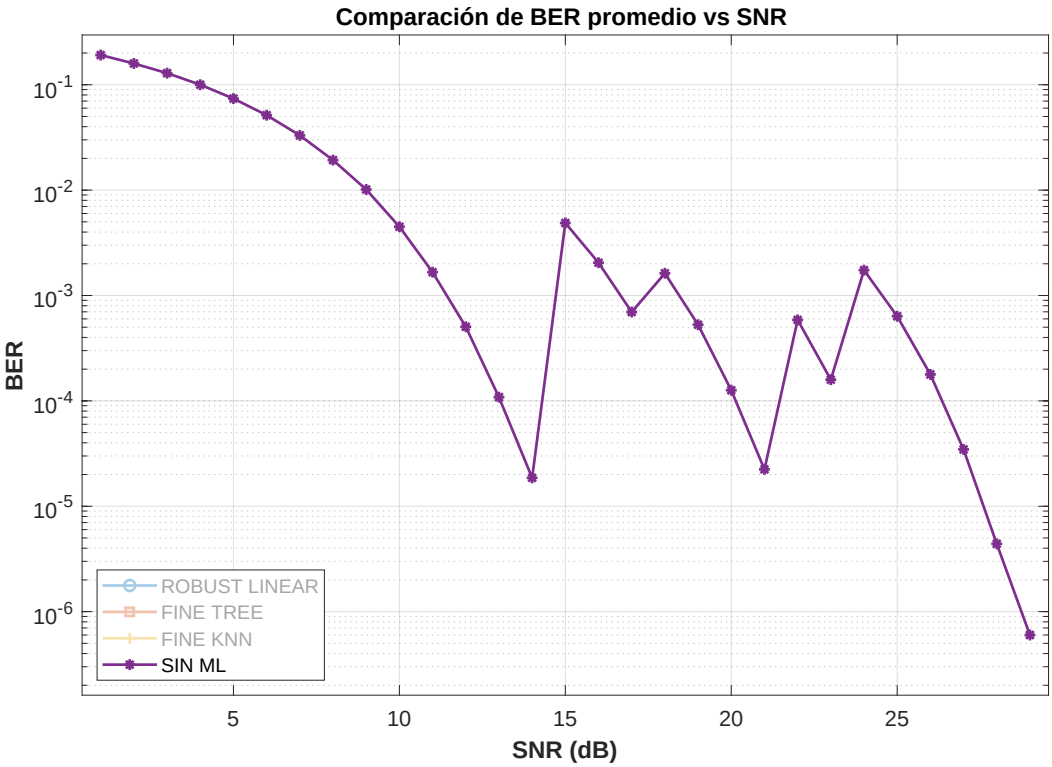

Figura 5.1: Comportamiento del BER | Modelo sin ML

Este modelo es determinista pues se determina en función de las condiciones iniciales y las reglas del modelo planteadas. En la Figura 5.1 se observa el comportamiento del BER con respecto a SNR del sistema OFDM sin ML. Para obtener un BER por debajo de 0.001 se debe llegar a un SNR de 25 [dB]. Dónde se estabiliza y no vuelve a tener picos como en los valores de 15, 18 y 24 [dB], donde el BER aumenta de manera drástica con valores superiores a 0.001.

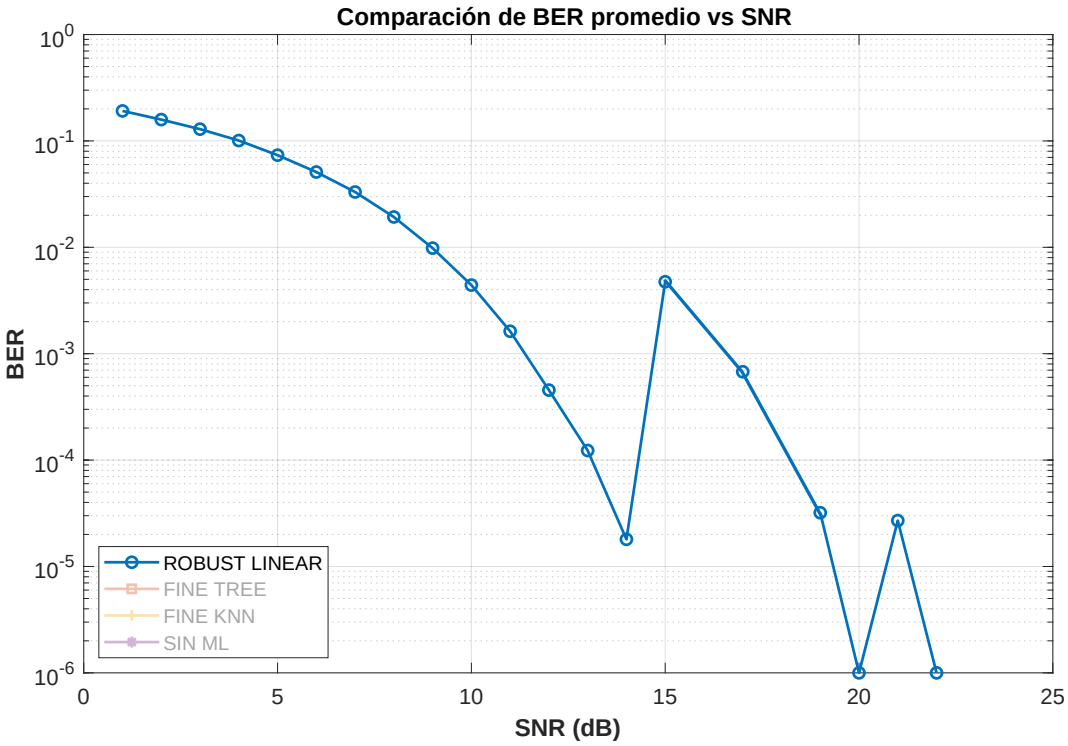

Figura 5.2: Comportamiento del BER | Modelo de *Robust Linear*

En la Figura 5.2 se observa el comportamiento del BER con respecto a SNR del sistema OFDM aplicando el modelo de *Robust Linear*. Para obtener un BER por debajo de 0.001 debe llegar a un SNR de 17 [dB]. Dónde no vuelve a tener picos que superen el BER objetivo, su comportamiento es óptimo para el sistema.

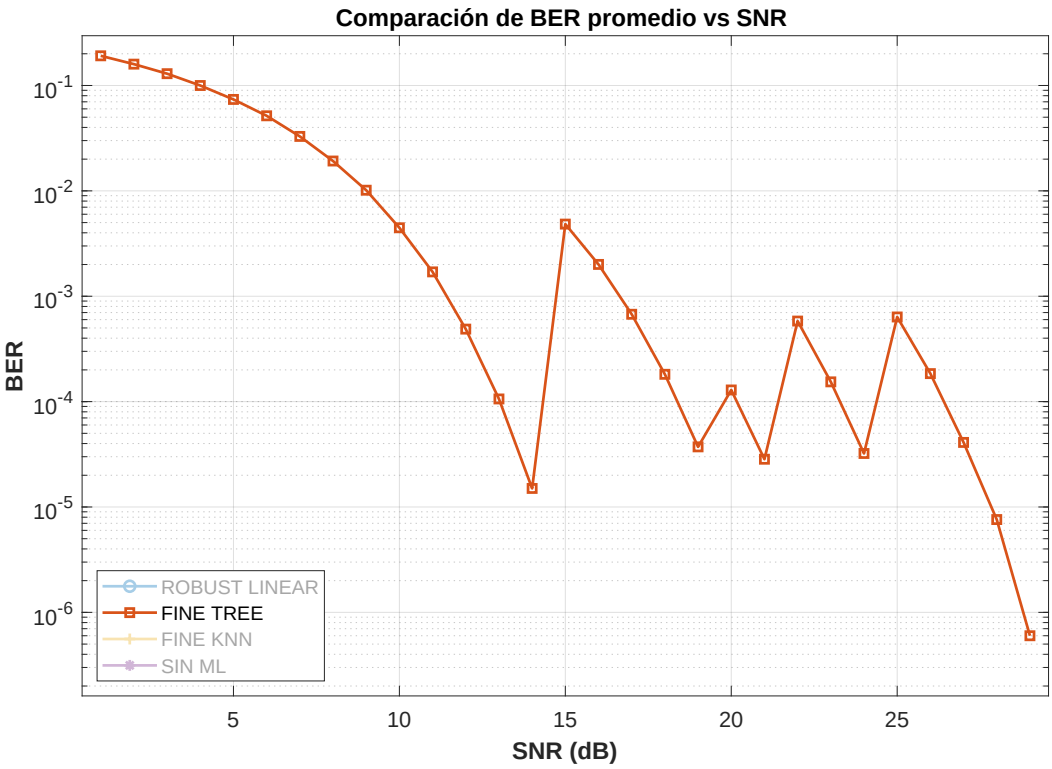

Figura 5.3: Comportamiento del BER | Modelo de *Fine Tree*

En la Figura 5.3 se observa el comportamiento del BER con respecto a SNR del sistema OFDM aplicando el modelo de *Fine Tree*. Para obtener un BER por debajo de 0.001 debe llegar a un SNR de 17 [dB]. Se muestra que al aumentar el valor del SNR se generan picos, pero estos no sobrepasan el valor objetivo que es 0.001, siendo un modelo óptimo para el sistema.

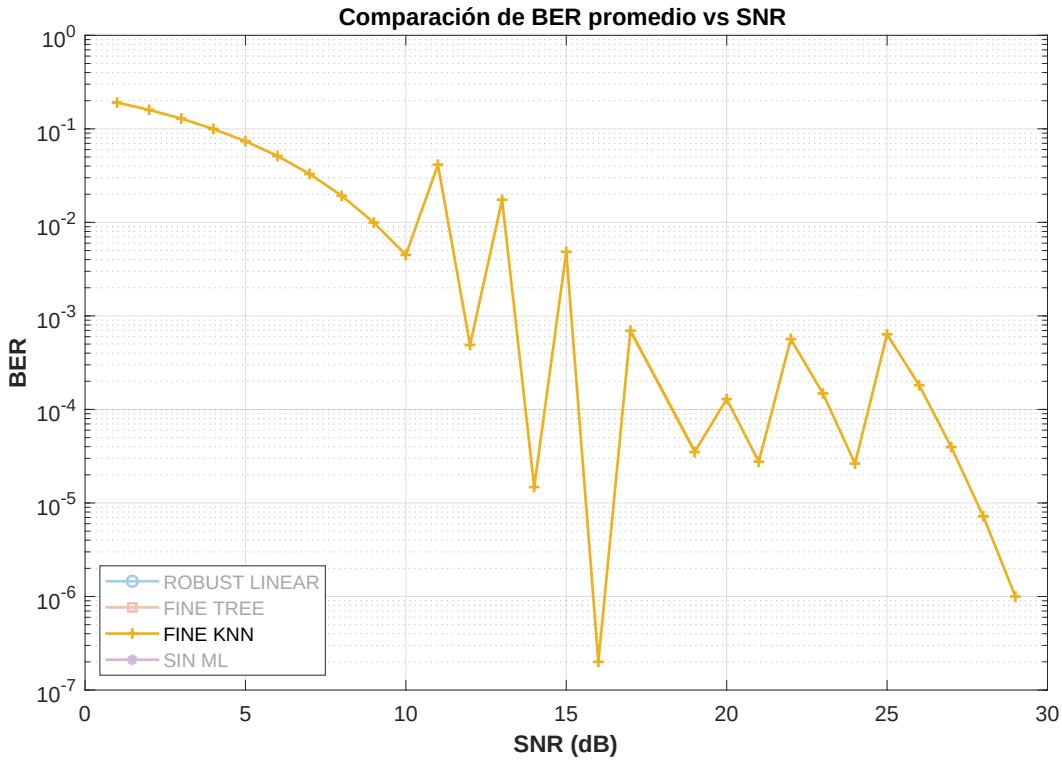

Figura 5.4: Comportamiento del BER | Modelo de *Fine KNN*

En la Figura 5.4 se observa el comportamiento del BER con respecto al SNR del sistema OFDM aplicando el modelo de *Fine KNN*. Para obtener un BER por debajo de 0.001 debe llegar a un SNR de 16 [dB]. Se muestra que al seguir aumentando el valor del SNR se generan picos, pero estos no sobrepasan el valor objetivo que es 0.001, siendo un modelo óptimo para el sistema. Sin embargo, en valores menores a 16 [dB] se presentan picos con un BER relativamente alto para que el sistema se mantenga estable en este rango.

Comparación del sistema OFDM Adaptativo con respecto a OFDM Adaptativo con ML

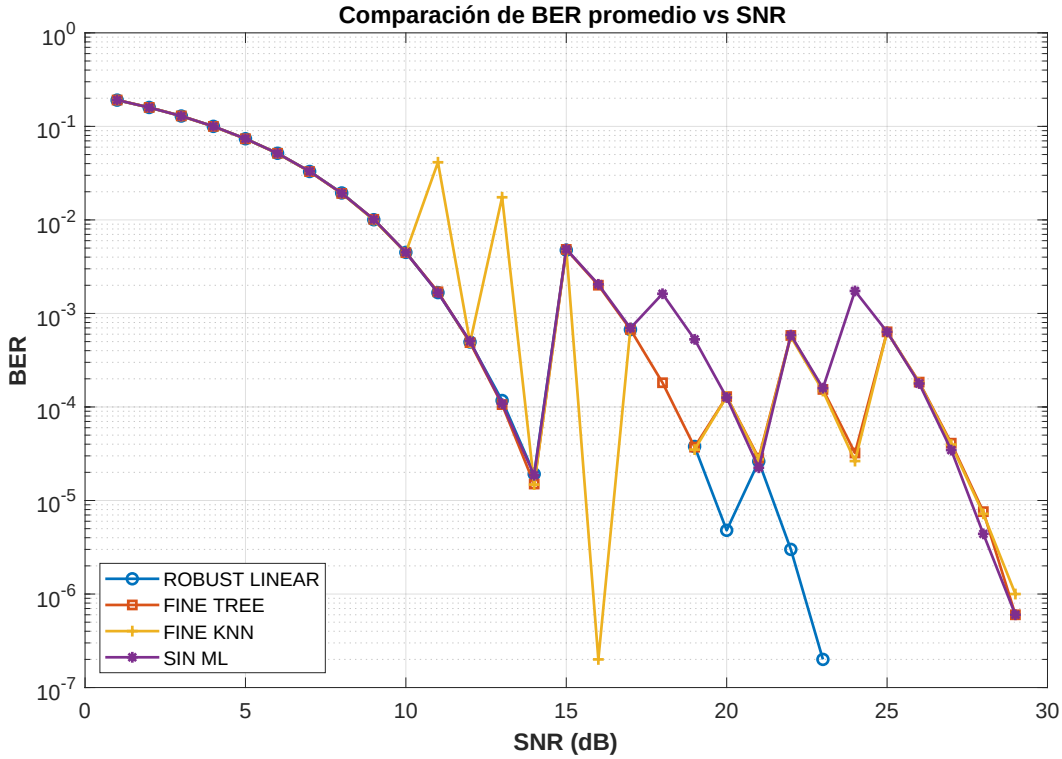

Figura 5.5: Comportamiento del BER de los 4 modelos

En la Figura 5.5 se observa el comportamiento del BER con respecto a SNR, de los cuatro casos descritos anteriormente. Se observa que el modelo de *Robust Linear* presenta un mejor rendimiento ya que genera un solo pico en 21 [dB] esto se debe a que solo maneja 3 tipos modulación y no selecciona modulaciones de mayor índice. Por otra parte el mejor modelo es de *Fine Tree* ya que a partir de los 17 [dB] se mantiene por debajo del BER objetivo al igual que el modelo de *Fine KNN*, con la diferencia que en este último modelo en el rango de valores menores a 15 [dB] presenta picos de una BER alto. El comportamiento del OFDM convencional sin ML tiene varios picos de BER que superan el valor de 0.001 y requiere de un SNR mayor que de las técnicas de ML para estabilizarse en UN valor por debajo del BER objetivo.

En este contexto el sistema OFDM adaptativo con ML tiene un mejor rendimiento que el convencional, ya que a un menor SNR alcanza el valor objetivo. Y de los tres modelos aplicados el *Fine Tree* es el mejor. Si bien el modelo de *Robust Linear* es estable no cambia a una modulación superior a los 16 [dB]. En base a los resultados obtenidos cada modelo presenta mejores resultados que otros en ciertos puntos como el modelo de FINE KNN para un SNR de 16 [dB], considerando que las condiciones del canal pueden variar, se puede tener un sistema que conmute entre modelos de Machine Learning logrando con esto un valor de BER bajo en todo

momento.

5.1.1. Comportamiento del *Throughput* y cambio de modulación

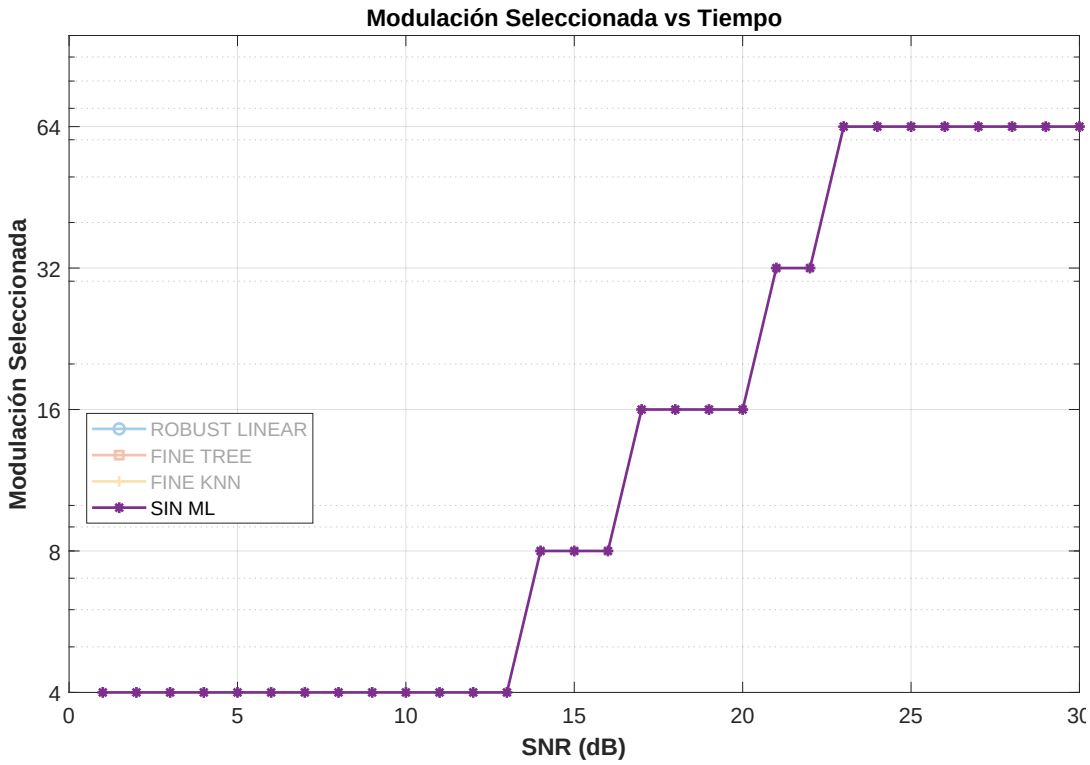

Figura 5.6: Comportamiento del índice de modulación | Modelo sin ML

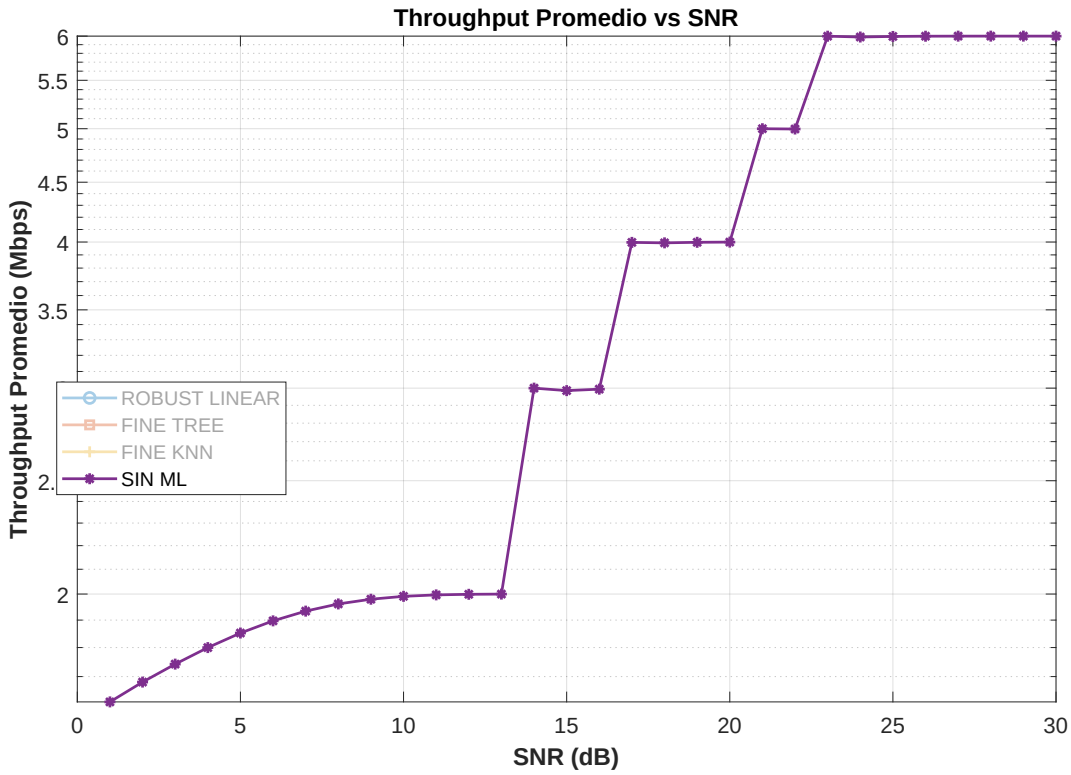

Figura 5.7: Comportamiento del *Throughput* | Modelo sin ML

En las Figuras 5.6 y 5.7 se muestra la adaptación de la modulación y el cambio del *Throughput* con respecto al valor del SNR. Al aumentar progresivamente este valor la modulación se va adaptando tomando valores más altos debido a las condiciones del canal, en este contexto el *Throughput* aumenta con respecto a la selección de modulación, ya que se incrementa la cantidad de bits por símbolos que se envían. Para un SNR de 23 [dB] se alcanza el valor más alto de *Throughput*, ya que se elige la modulación más alta (64 QAM).

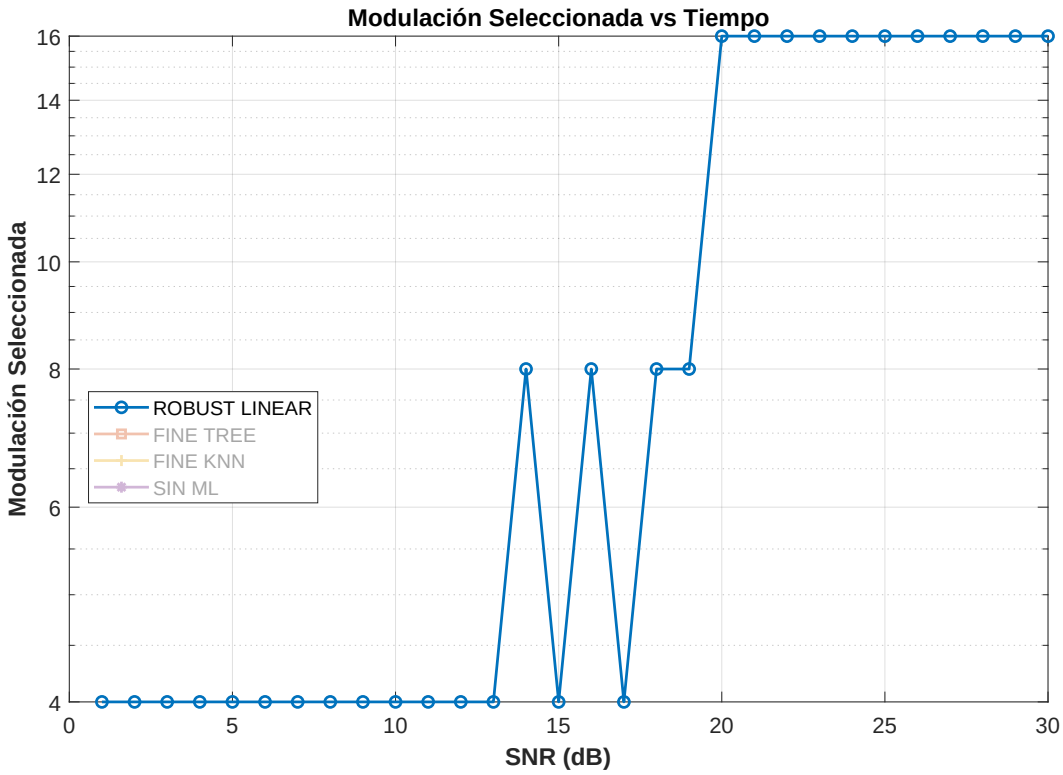

Figura 5.8: Comportamiento del índice de modulación | Modelo *Robust Linear*

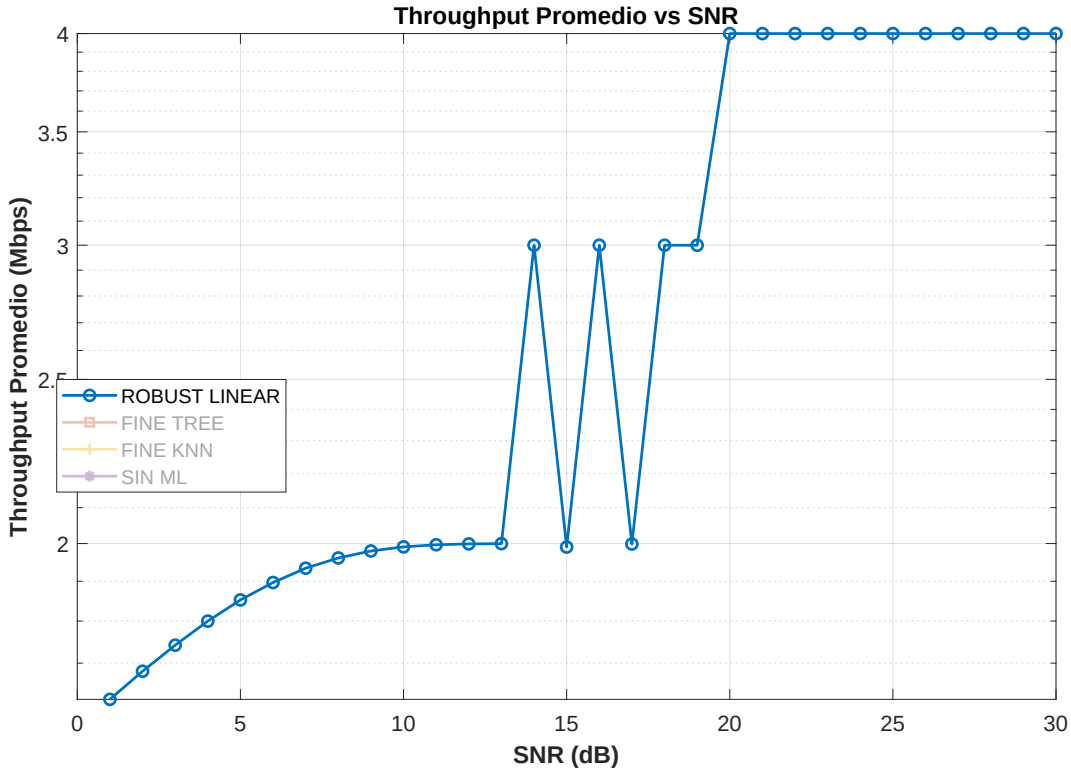

Figura 5.9: Comportamiento del *Throughput* | Modelo *Robust Linear*

En las Figuras 5.6 y 5.7 se muestra la adaptación de la modulación y el cambio del *Throughput*

con respecto al valor del SNR, implementando el modelo de *Robust Linear* de ML. Se observa que no selecciona modulaciones superiores a 16-QAM, siendo el índice de modulación de modulación seleccionado el 16 el más alto, estabilizándose en esta modulación para valores de SNR superiores a 18 [dB]. En el rango de [13 dB - 18 dB] tiene picos que reflejan un cambio de modulación inestable ya que varía el índice entre 4 (QPSK) y 8 (8-PSK), esto se debe a la estimación del canal que se maneja y la forma en que se determina la modulación a la que corresponde la predicción realizada por el modelo, recordando que este no genera valores exactos y se aplica un método de redondeo, que limita la selección de modulación a índices no superiores a 16. Esto provoca que el *Throughput* no aumente incluso con un valor de SNR alto.

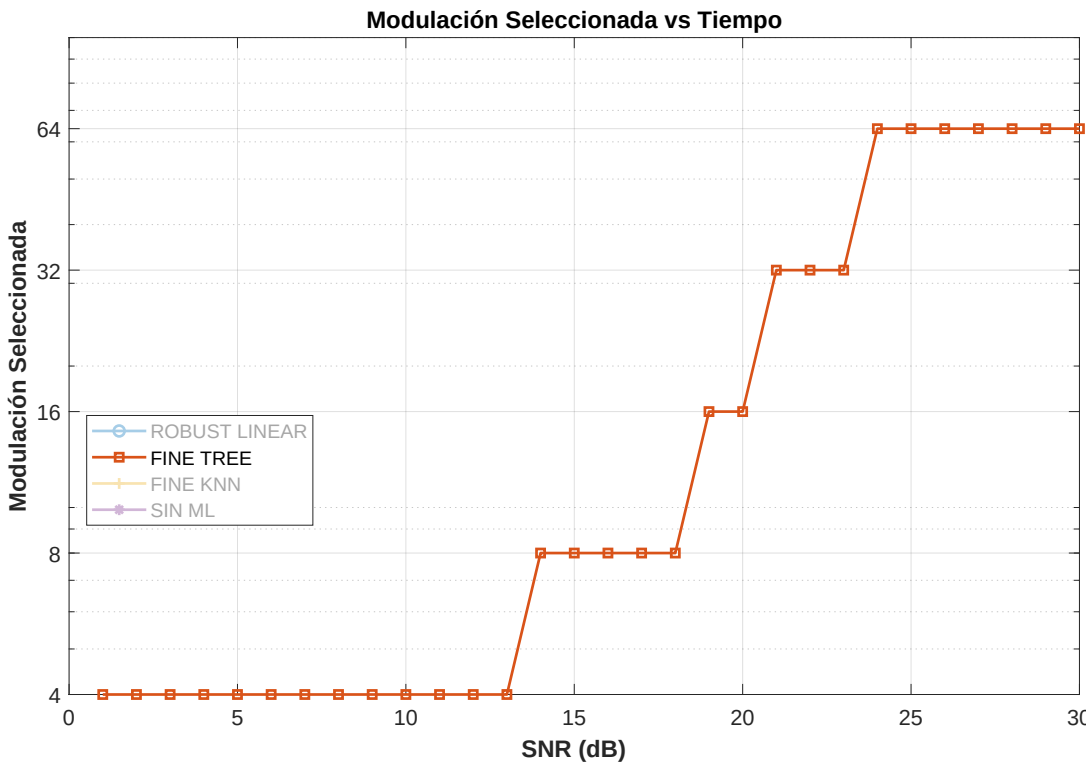

Figura 5.10: Comportamiento del índice de modulación | Modelo *Fine Tree*

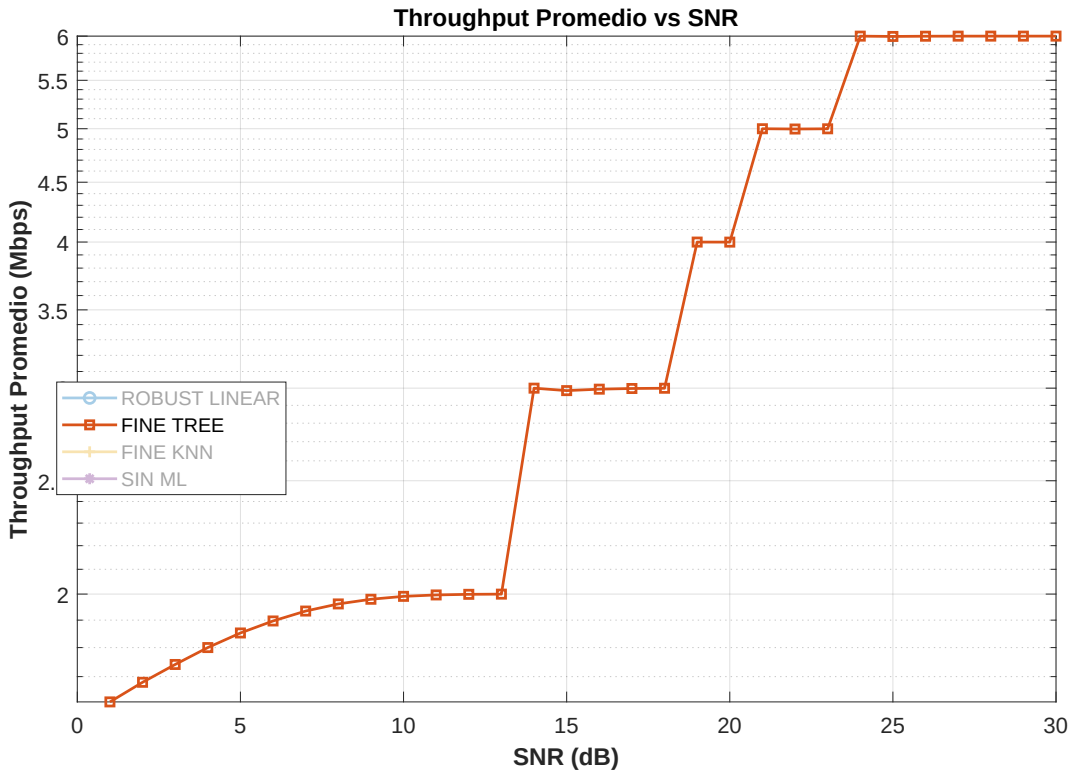

Figura 5.11: Comportamiento del *Throughput* | Modelo *Fine Tree*

En las Figuras 5.10 y 5.11 se muestra la adaptación de la modulación y el cambio del *Throughput* con respecto al valor del SNR, implementando el modelo de *Fine Tree* de ML. Se observa que la selección de modulación se realiza conforme aumenta progresivamente el valor del SNR, cambiando a una modulación superior para valores más altos de SNR. El modelo se maneja de forma estable y a la par el *Throughput* aumenta proporcionalmente ya que a un índice mayor de modulación se incremente la cantidad de bits por símbolo, llegando a un valor máximo de *Throughput* en un SNR de 24 [dB] donde selecciona la mayor modulación de 64-QAM.

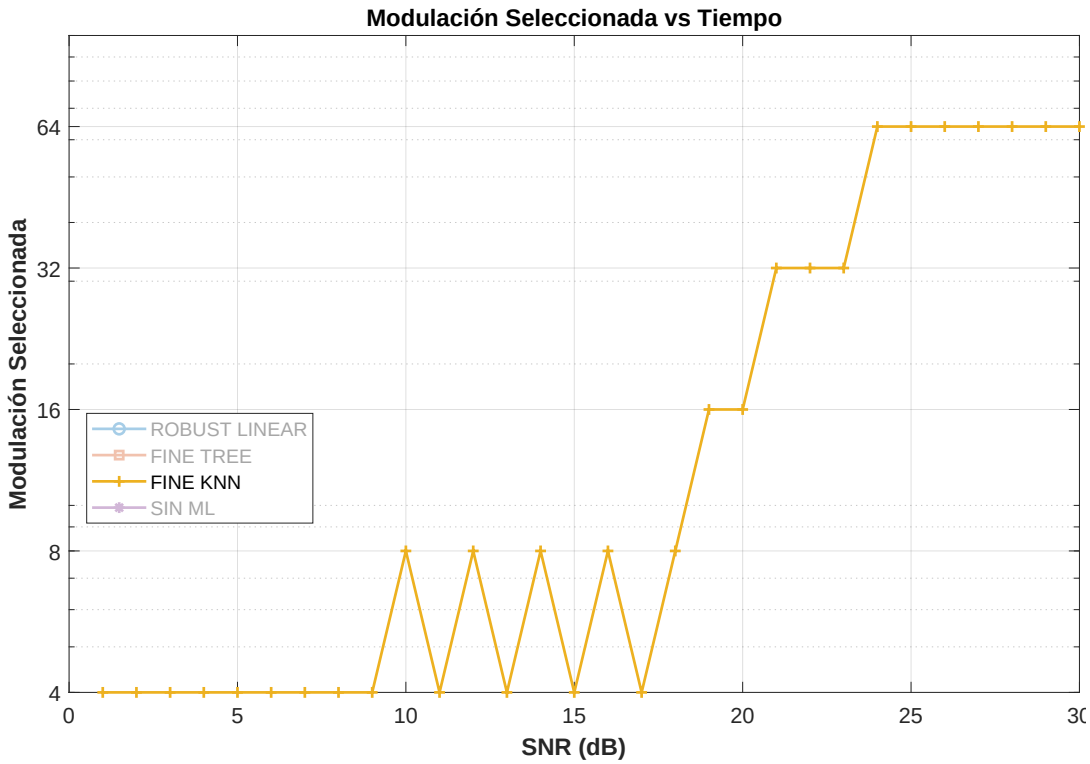

Figura 5.12: Comportamiento del índice de modulación | Modelo *FINE KNN*

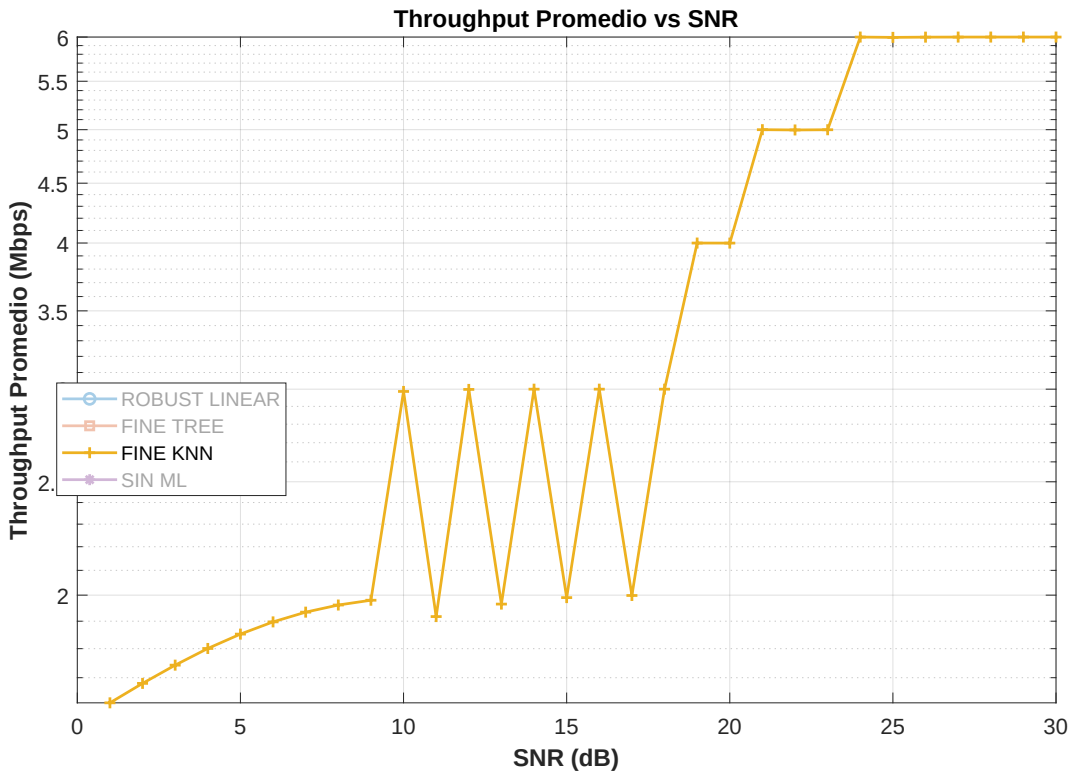

Figura 5.13: Comportamiento del *Throughput* | Modelo *Fine KNN*

En las Figuras 5.12 y 5.13 se muestra la adaptación de la modulación y el cambio del *Through-*

put con respecto al valor del SNR, implementando el modelo de *Fine KNN* de ML. Se analiza en dos rangos, para valores superiores a los 17 [dB] la selección de la modulación es estable pues aumenta conforme incrementa el valor del SNR. En el rango de [9 dB - 17 dB] se observan picos producto de los cambios inestables que se generan, esto se produce por la estimación del canal que se maneja, ya que los valores que se están obteniendo son superiores al valor real y esto no permite que se estabilice la modulación en una sola clase si no que se alterne entre las modulaciones de QPSK y 8PSK. El *Throughput* actúa conforme la selección de la modulación, en el rango de [9 dB - 17 dB] se observa igualmente la variación que se genera por la estimación del canal.

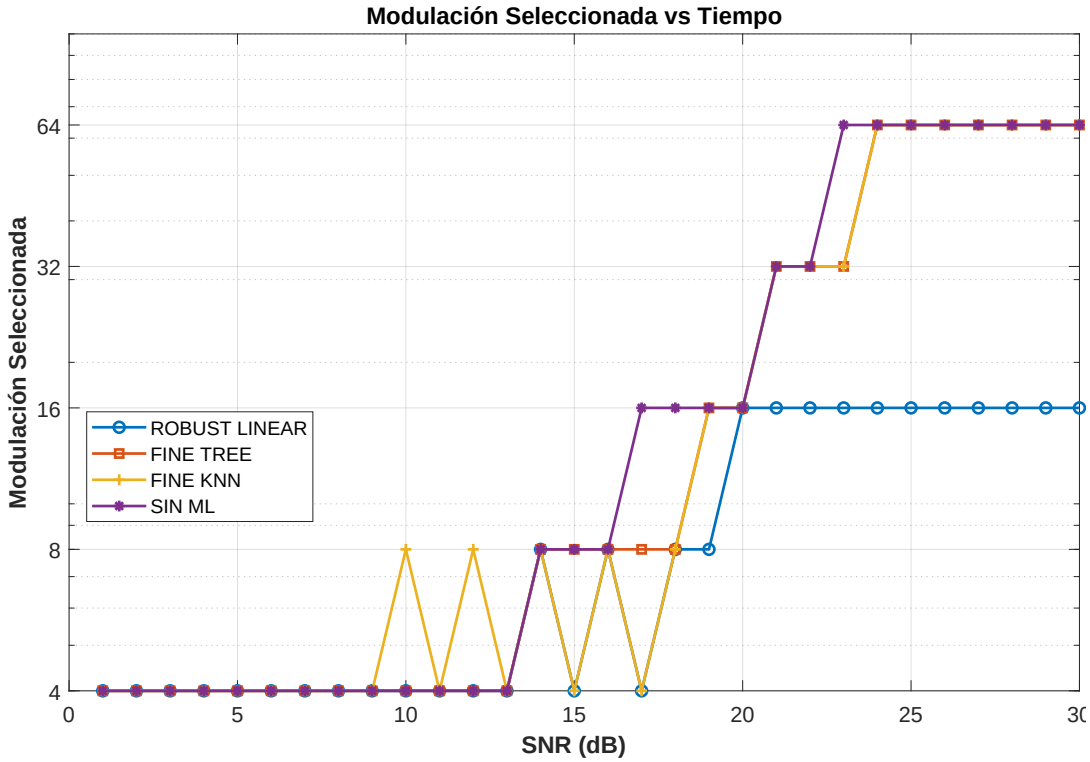

Figura 5.14: Comportamiento del índice de modulación de los 4 modelos

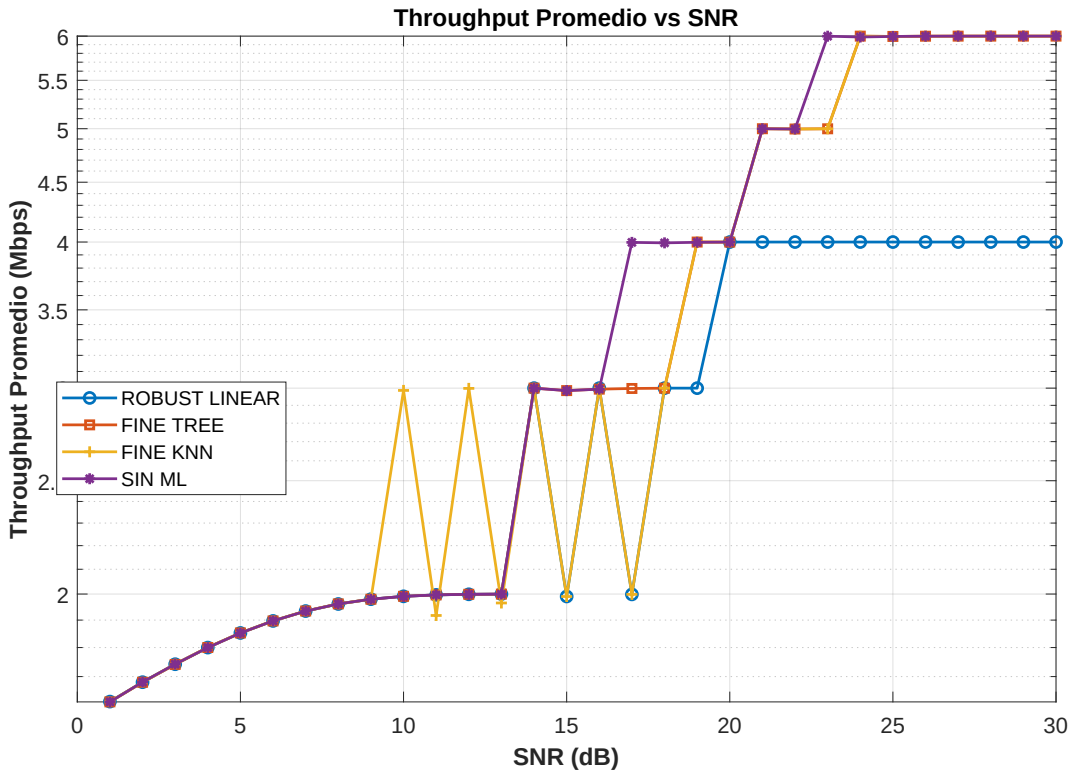

Figura 5.15: Comportamiento del *Throughput* de los 4 modelos

En las Figuras 5.14 y 5.15 se observa el comportamiento de la selección de la modulación y del *Throughput* de los cuatro casos descritos anteriormente. Tanto el modelo de *Robust Linear* como de *Fine KNN* presentan una inestabilidad en la selección para valores de SNR menores a 17 [dB] producto de la estimación del canal, adicionalmente por las características del modelo de *Robust Linear* no alcanza el máximo *Throughput* siendo el modelo con el rendimiento más bajo. Por otra parte el modelo de *Fine Tree* es el que mejor se adapta a los cambios de modulación, a pesar de la estimación del canal, es estable en la selección de la modulación al igual que el modelo sin ML, pues siguen el mismo patrón de selección, a excepción de los valores de 17 [dB], 18 [dB] y 23 [dB] donde selecciona un índice menor de modulación.

En este contexto el sistema OFDM adaptativo sin ML, frente a los dos modelos *Robust Linear* y *Fine KNN* es superior ya que presenta mayor estabilidad en la selección de la modulación y mayor *Throughput*. En comparación con el modelo de *Fine Tree* son similares y en ciertos valores de SNR tiene mayor *Throughput* debido a que selecciona una modulación mayor antes que el modelo con ML.

Escenario 2

Para este escenario se establece un canal aleatorio cuya variación entre los valores aleatorios de SNR esta entre $\pm$ 2 [dB], este comportamiento se da en los canales inalámbricos y de larga distancia, ya que los cambios no son drásticos si no suaves como se muestra en la Figura 5.16.

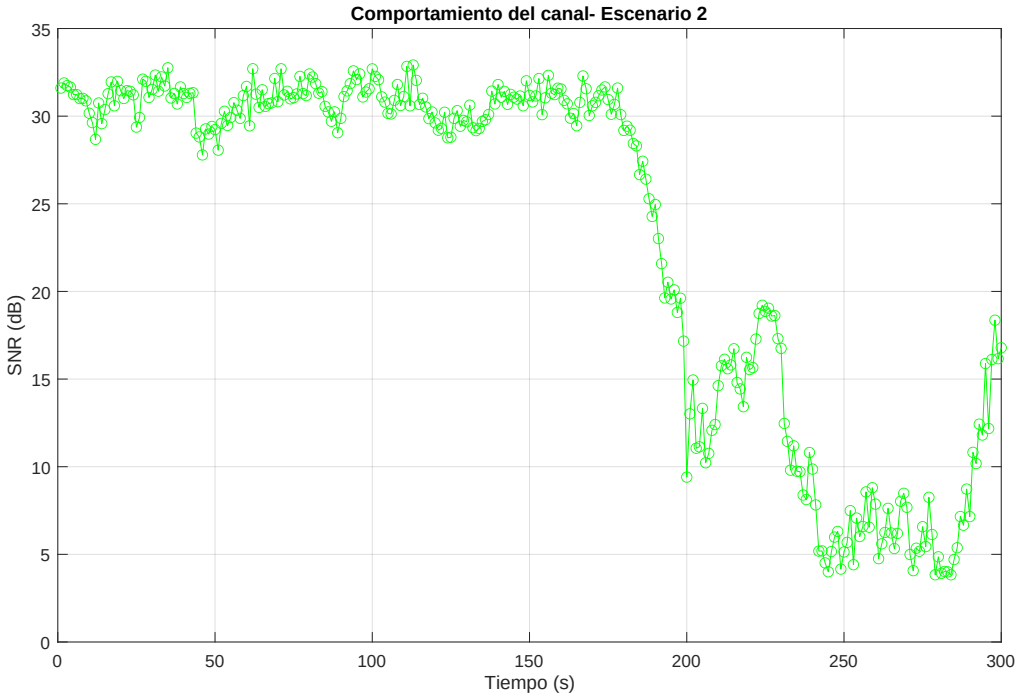

Figura 5.16: Comportamiento del canal | Escenario 2

Para la evaluación se considera los resultados del BER respecto a la variación del SNR. Se analiza el comportamiento del sistema OFDM convencional y empleando técnicas de ML. Tomando en cuenta que el valor del BER objetivo es por debajo de 0.001. Se presentan las gráficas del comportamiento del BER promediado de los cuatro casos planteados, al final de la sección se realiza una discusión con base en la comparación del sistema OFDM Adaptativo con respecto a OFDM Adaptativo con ML.

Comportamiento del BER

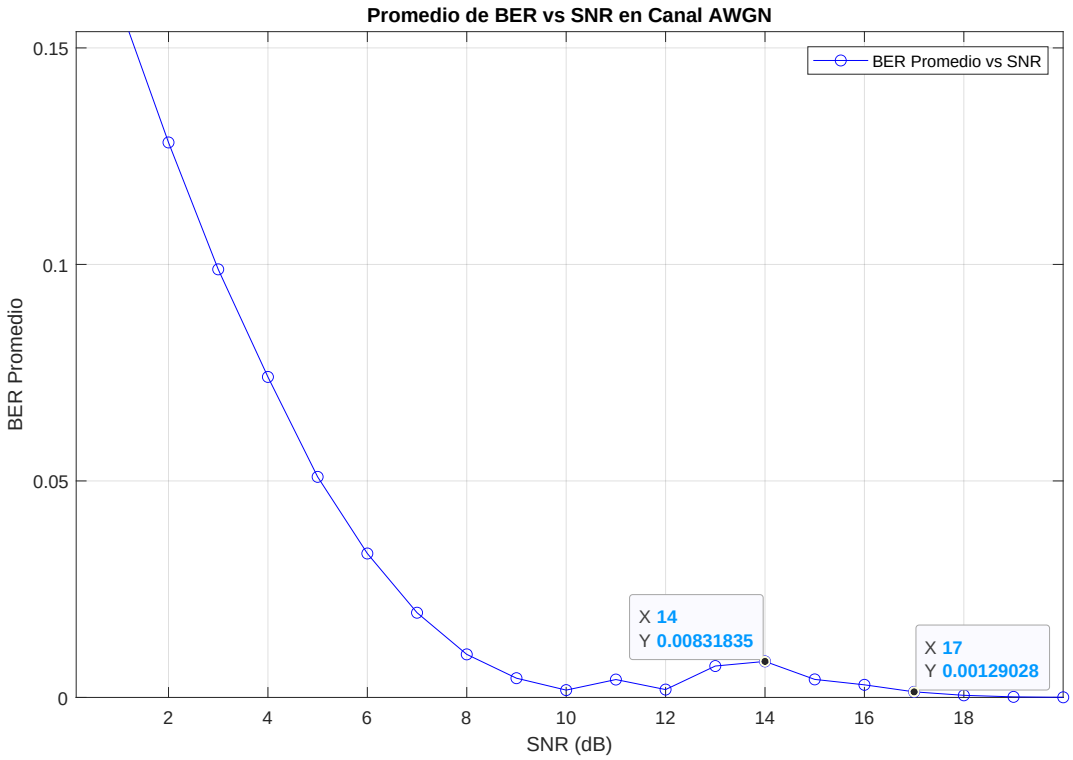

Figura 5.17: Comportamiento del BER | Modelo sin *Machine Learning*

En la Figura 5.17 se observa el comportamiento del BER con respecto a SNR del sistema OFDM sin ML. La gráfica es un promedio del BER en cada uno de los valores del SNR. El pico más alto se da en los 14 [dB] ya que se alcanza un BER aproximado de 0.008 un valor superior al BER objetivo. El sistema se estabiliza a partir de los 17 [dB] donde el BER es 0.0012 y los siguientes valores están por debajo de 0.001.

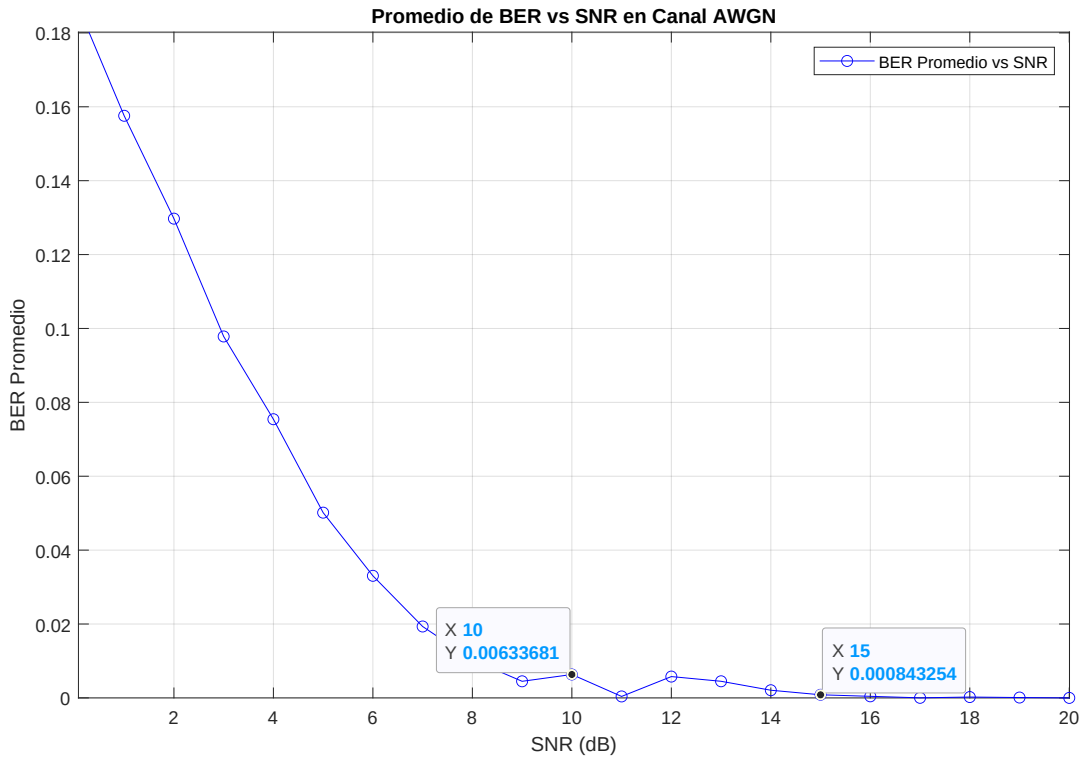

Figura 5.18: Comportamiento del BER | Modelo de *Robust Linear*

En la Figura 5.18 se observa el comportamiento del BER con respecto a SNR del sistema OFDM implementando el modelo de *Robust Linear*. La gráfica es un promedio del BER en cada uno de los valores del SNR. Para este caso el pico más alto se da en 10 [dB] con un BER de 0.006, el BER se estabiliza en valores menores a 0.001 a partir de los 15 [dB]. Tiene un alto rendimiento en función del BER, pero este modelo no selecciona modulaciones superiores a 16-QAM, lo que limita el valor del *Throughput* resaltado en los resultados del escenario 1.

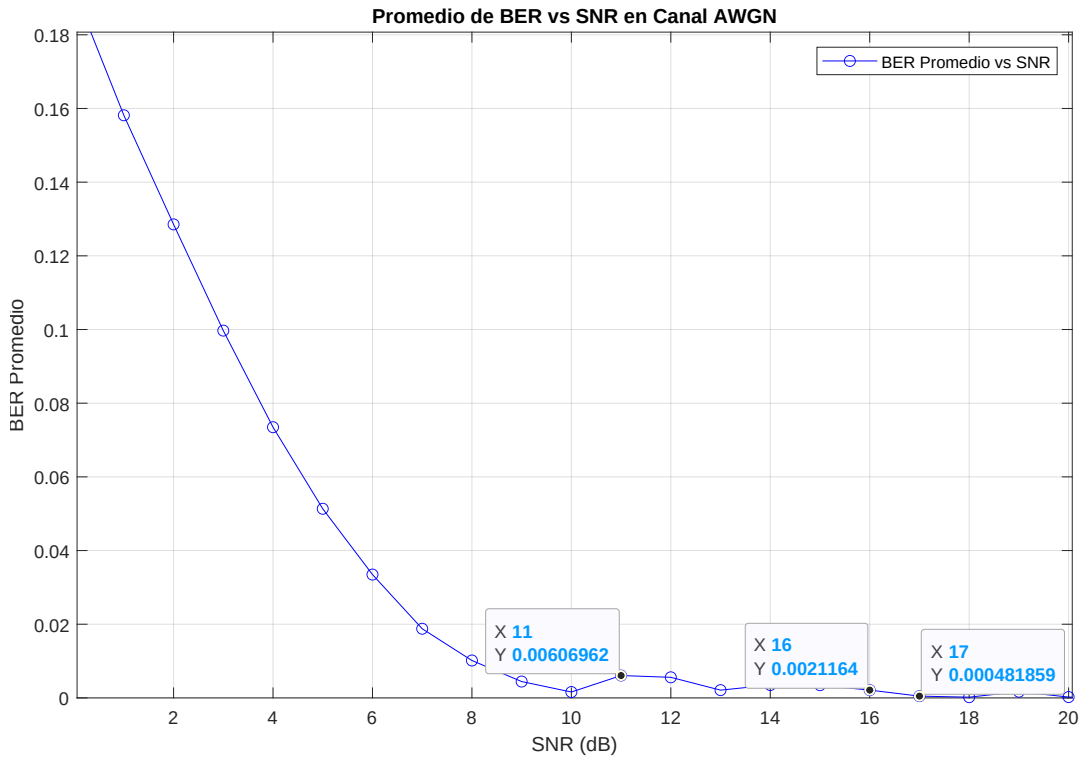

Figura 5.19: Comportamiento del BER | Modelo de *Fine Tree*

En la Figura 5.19 se observa el comportamiento del BER con respecto a SNR del sistema OFDM implementando el modelo de *Fine Tree*. La gráfica es un promedio del BER en cada uno de los valores del SNR. Para este caso el valor pico se ubica en 11 [dB] con 0.006, y a partir de los 16 [dB] con un BER de 0.002 ya se toman valores del BER menores a 0.001, para un SNR 17 [dB] es de 0.0004.

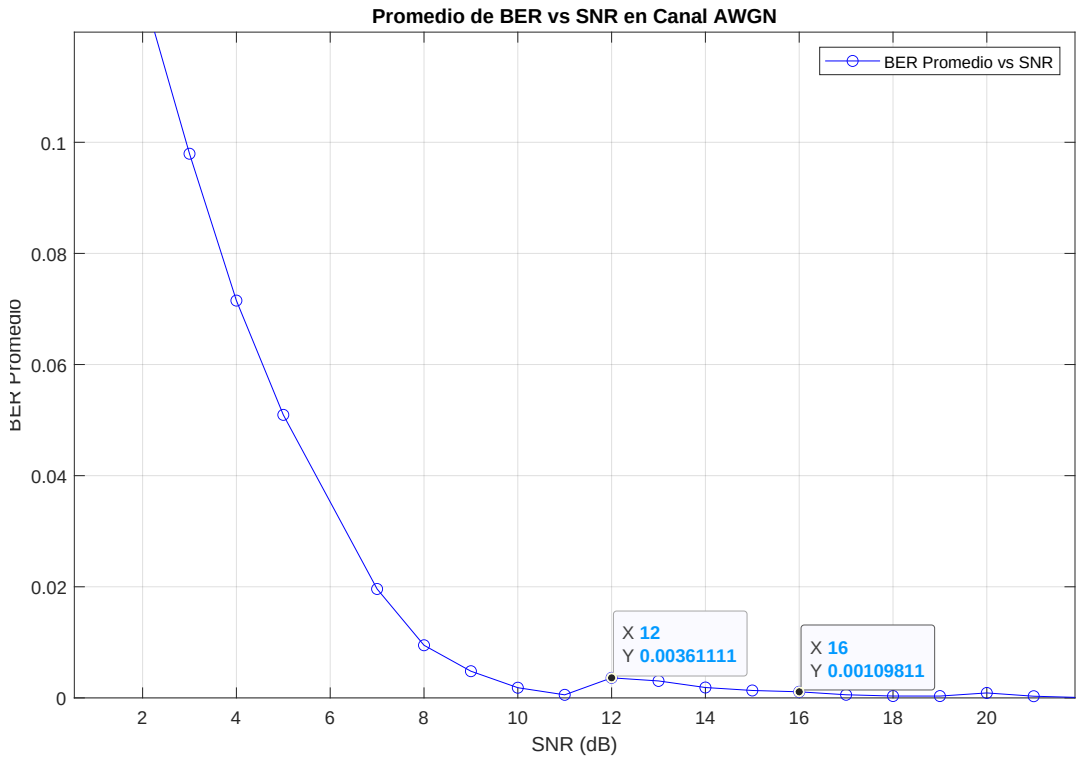

Figura 5.20: Comportamiento del BER | Modelo de *Fine KNN*

En la Figura 5.20 se observa el comportamiento del BER con respecto a SNR del sistema OFDM implementando el modelo de *Fine KNN*. La gráfica es un promedio del BER en cada uno de los valores del SNR. Para este caso el valor pico se ubica en 12 [dB] con 0.003, y a partir de los 16 [dB] con un BER de 0.00109 ya se toman valores del BER menores a 0.001.

Comparación del sistema OFDM Adaptativo con respecto a OFDM Adaptativo con ML

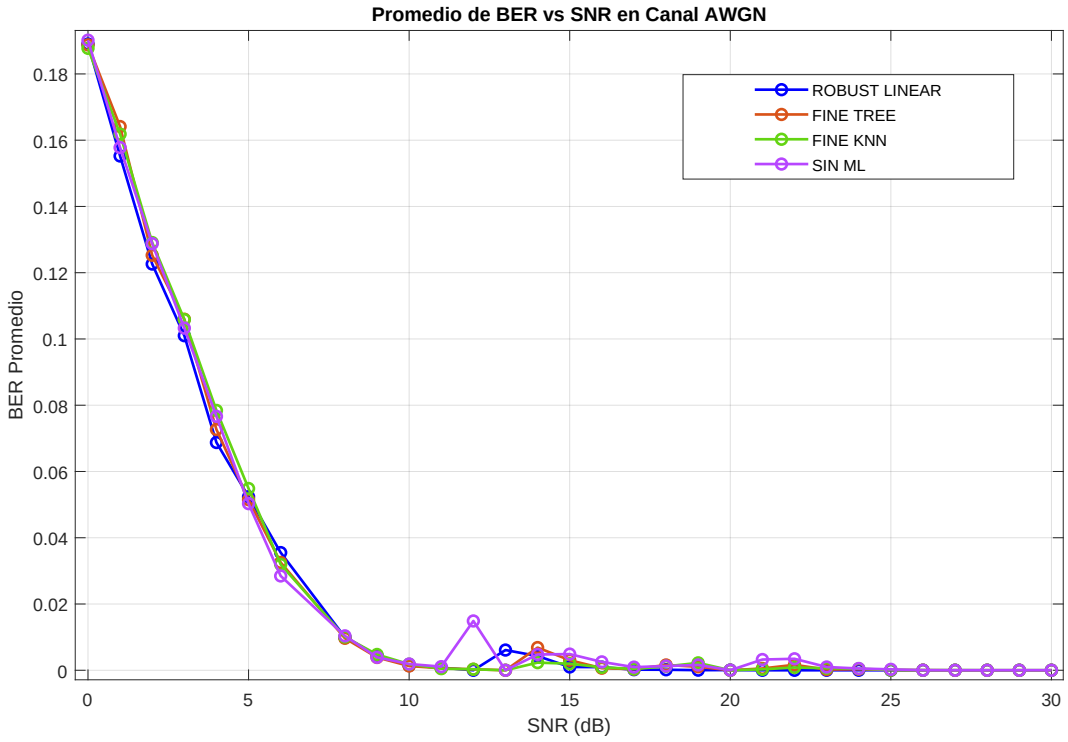

Figura 5.21: Comportamiento del BER de los 4 modelos | [0dB - 30 dB]

En la Figura 5.21, se observa el comportamiento del BER promedio con respecto a SNR, en el rango de [0 dB - 30 dB], para un SNR menor a 10 dB en los cuatro casos se tiene un comportamiento similar, pues el BER es alto para cada uno de los valores de SNR.

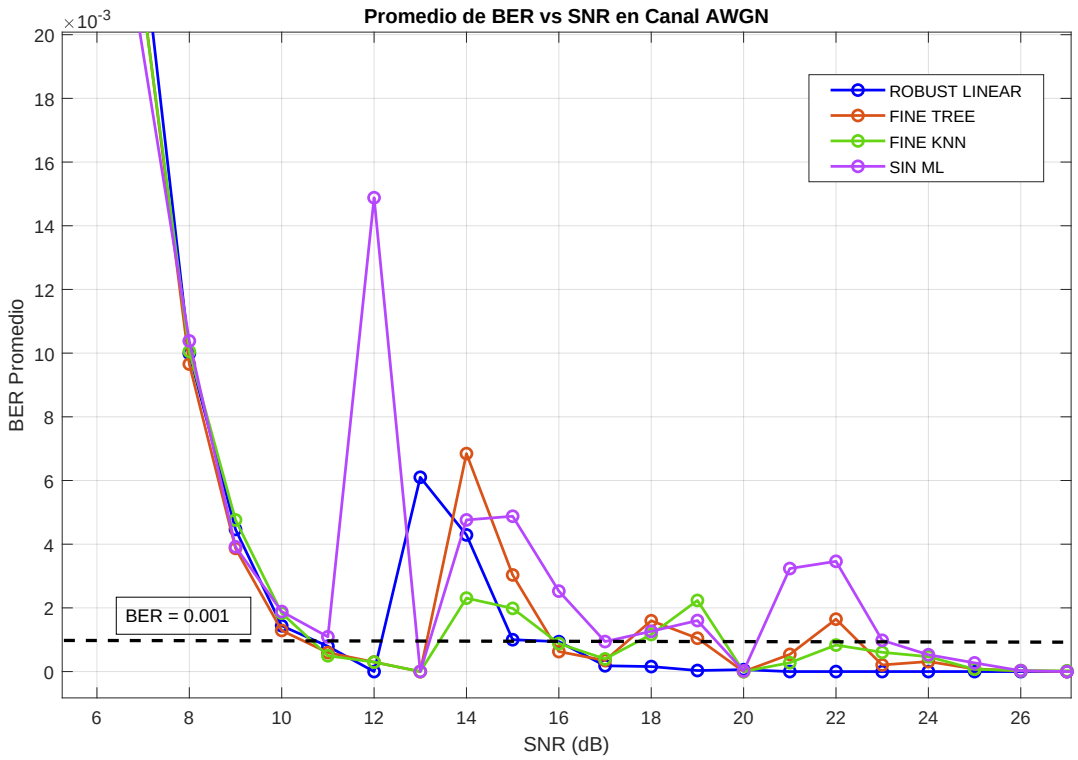

Figura 5.22: Comportamiento del BER de los 4 modelos | [6dB - 26 dB]

En la Figura 5.22 se observa el comportamiento del BER promedio con respecto al SNR para este tipo de canal de los cuatro casos descritos anteriormente, esta Figura 5.22 es un acercamiento de la Figura 5.21 con el objetivo de visualizar los cambios de mejor manera. En la gráfica se resalta el BER objetivo de 0.001, se observa que para este escenario tanto el modelo de *Robust Linear* y de *Fine KNN* presentan valores de BER inferiores a 0.001, en valores de SNR de 16 [dB] y 19 [dB] respectivamente, mientras que el modelo de *Fine Tree* lo realiza en un SNR de 23 [dB], dejando ver que para este escenario el mejor modelo es *Fine KNN*, no consideramos el modelo de *Robust Linear* como el mejor a pesar del su buen rendimiento ya que por el tipo de selección que realiza se limita a tomar las tres modulaciones más robustas que para un SNR de mayor valor siempre el BER será inferior, pero no considerara el cambio de modulación para optimizar y mejorar el *Throughput* del sistema. El modelo con peor rendimiento es el convencional pues presenta muchos picos y se estabiliza en un SNR superior a 20 [dB].

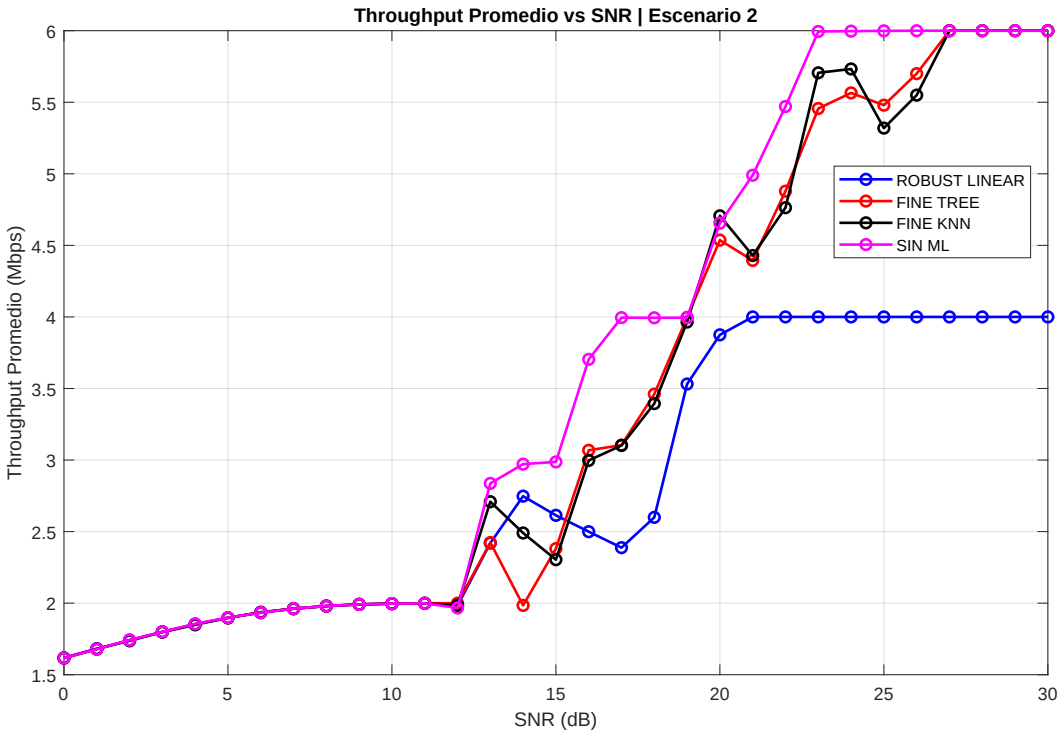

Figura 5.23: Comportamiento del *Throughput* Promedio con respecto a SNR | Cuatro modelos | [0dB - 30 dB]

En la Figura 5.23 se observa el comportamiento del *Throughput* de los cuatro casos descritos anteriormente. Por las características del modelo de *Robust Linear* no alcanza el máximo *Throughput* siendo el modelo con el rendimiento más bajo.

Los modelos de *Fine Tree* y *Fine KNN* poseen características similares en los cambios de *Throughput*, aunque llega a tener un *Throughput* alto como el sistema OFDM adaptativo sin ML llega a tener el máximo *Throughput* en los 27 dB, a diferencia del sin ML que llega a su máximo *Throughput* en los 23 dB.

En este contexto el sistema OFDM adaptativo sin ML, frente a los modelos de ML tiene un menor rendimiento, ya que presenta mayores picos de BER al seleccionar una nueva modulación, lo que genera que el sistema no tenga un BER por debajo de 0.001 mientras no llegue a valores de SNR altos.

Escenario 3

Para este escenario se establece un canal con una variación del SNR totalmente aleatoria, este comportamiento es un caso extremo en el cual se considera para analizar el comportamiento de los cuatro casos frente a un canal totalmente aleatorio como se muestra en la Figura 5.24.

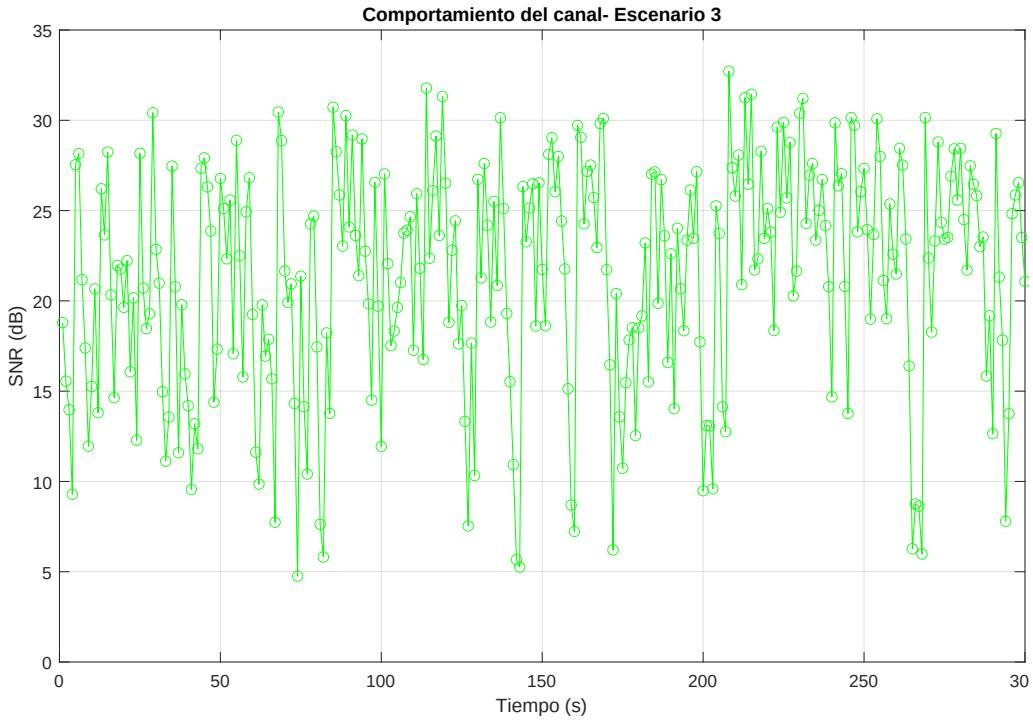

Figura 5.24: Comportamiento del canal | Escenario 3

Para la evaluación se considera los resultados del BER respecto a la variación del SNR. Se analiza el comportamiento del sistema OFDM convencional y empleando técnicas de ML. Tomando en cuenta que el valor del BER objetivo es por debajo de 0.001. Se presentan las gráficas del comportamiento del BER promediado de los cuatro casos planteados, al final de la sección se realiza una discusión con base en la comparación del sistema OFDM Adaptativo con respecto a OFDM Adaptativo con ML.

Comportamiento del BER

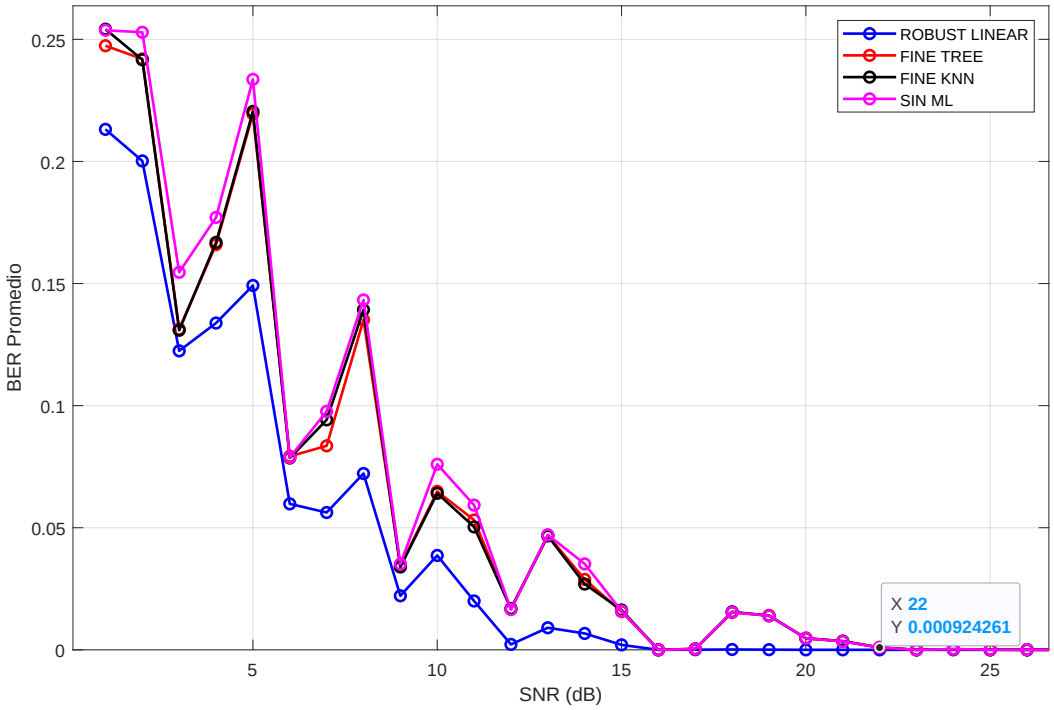

Figura 5.25: Comportamiento del BER Promedio con respecto a SNR | Cuatro modelos

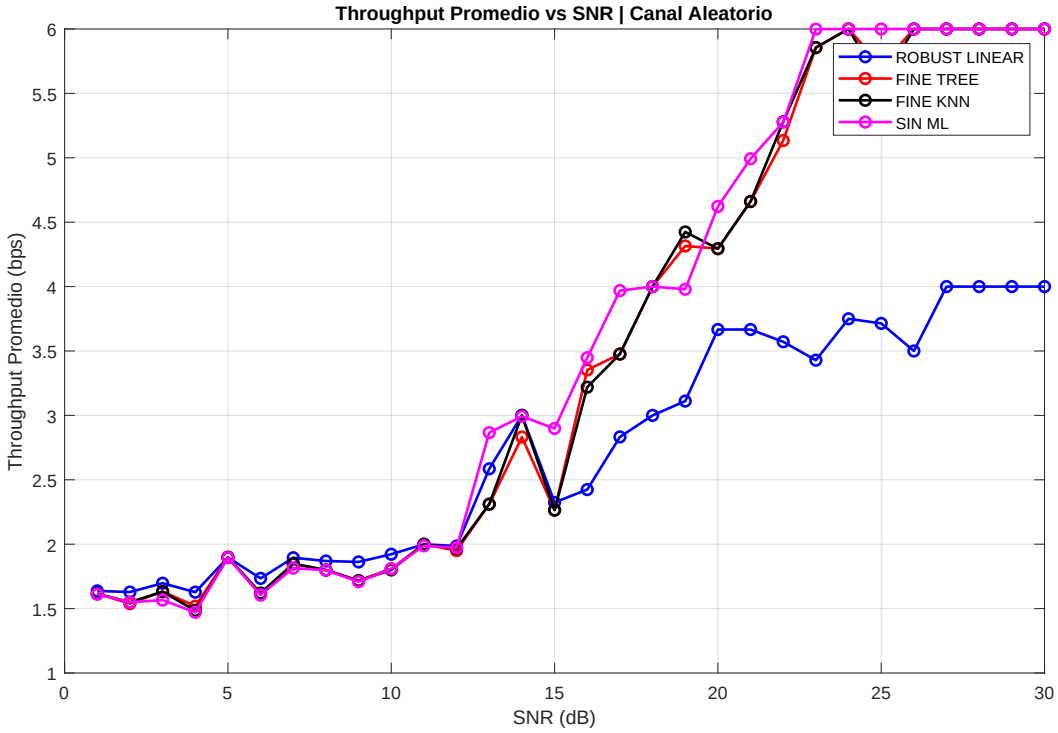

Figura 5.26: Comportamiento del *Throughput* Promedio con respecto a SNR | Cuatro modelos

En la Figura 5.25 se observa el comportamiento del BER promedio con respecto a SNR, este

canal es totalmente aleatorio por lo que los cambios de modulación pueden ser demasiado extremos, lo que genera un BER alto. Para este punto el modelo sin ML, *Fine Tree* y *Fine KNN* tienen un comportamiento similar, con un BER sumamente alto en el rango de [0 dB - 20dB], con un BER de 22[dB] estos tres modelos se estabilizan en valores inferiores a 0.001. El modelo de *Robust Linear* presenta un mejor rendimiento con valores de BER inferiores a los otros modelos, y se estabiliza en un BER de 16 [dB], esto se produce porque la máxima modulación que selecciona es de 16-QAM y puede manejar valores de SNR inferiores a 20 [dB]. En la Figura 5.26 se observa el *Throughput* del sistema, si bien los tres modelos: sin ML, *Fine Tree* y *Fine KNN* tienen un buen rendimiento del *Throughput*, para llegar al BER deseado es necesario alcanzar un SNR alto, por lo que para este caso es mejor el modelo de *Robust Linear*, si bien no alcanza un *Throughput* alto, pero mantiene una estabilidad en el BER. En este contexto, el modelo de OFDM adaptativo con ML es mejor el modelo convencional planteado.

6. Conclusiones y recomendaciones

6.1. Conclusiones

Para lograr la implementación de OFDM adaptativo con técnicas de *Machine Learning* (ML) supervisadas, se utilizó el ajuste dinámico de la modulación en función de tres parámetros: la tasa de Error de Bit (BER), la modulación en tiempo real del sistema y la estimación del canal. Se evaluó el BER del modelo OFDM adaptativo convencional, para establecer la base de datos que tiene la información con respecto al cambio de modulación en función de los valores de los tres parámetros establecidos. Para evaluar el rendimiento del BER en el sistema OFDM adaptativo con ML, se lo realiza en tres escenarios. En el primer escenario, el Signal to Noise Ratio (SNR) incrementa de manera progresiva desde 0 hasta 30 [dB], mientras que en el segundo escenario el SNR varía de manera aleatoria con incrementos entre $\pm$ 2 [dB], y finalmente un tercer escenario donde el SNR varía entre 0 y 30 [dB] de manera aleatoria.

Analizando el rendimiento de OFDM adaptativo convencional en el primer y segundo escenario se tiene un buen rendimiento en términos del BER como en el throughput. En el primer escenario con SNR a partir de los 17 [dB] ya se tienen valores de BER menores a 0.001, con un pico único en 24 [dB] llegando a su máximo throughput de 6 [Mbps] en 23 [dB]. En el segundo escenario se tiene un comportamiento similar pues a partir de los 17 [dB] ya se obtienen solo valores de BER menores a 0.001 y el throughput también aumenta constantemente, y alcanza su máximo valor de 6 [Mbps] en 23 [dB]. En el tercer escenario los valores del BER se estabilizan a partir de 22 [dB], que representan 5 [dB] más que en los dos primeros escenarios y el throughput alcanza su valor máximo en 23 [dB], pero su incremento ya no se da de manera constante como en el primer escenario, debido a las características propias del canal.

Machine Learning tiene un conjunto amplio de técnicas que pueden ser aplicadas, las cuales se utilizan dependiendo del tipo de datos y la aplicación, haciendo que ciertas técnicas sean más apropiadas que otras. El sistema OFDM adaptativo con ML ajusta la modulación en función del índice, determinado por los tres parámetros: BER, modulación en tiempo real y estimación del canal, por lo que el resultado del modelo de ML debe ser un valor exacto que representa al índice de las cinco modulaciones, 4 para la modulación QPSK, 8 para la 8PSK, 16 para la modulación 16QAM, 32 para la modulación 32QAM y 64 para la modulación 64QAM. En este trabajo se implementan y comparan las técnicas de clasificación y de regresión lineal en el sistema OFDM adaptativo. La técnica de regresión lineal ajusta los datos a una función lineal, sin embargo, el valor resultante no es exacto, sino una aproximación, el cual al tener

más de dos parámetros, requiere un proceso adicional para definirlo de manera exacta, lo que representa una limitación al momento de seleccionar la modulación correcta. Por otro lado, la técnica de clasificación determina el índice de modulación exacto en función de los valores de los tres parámetros, lo que da como resultado el número del índice de una de las cinco modulaciones. Estos resultados indican que las técnicas de clasificación son las más adecuadas para el sistema que se implementa, pues el resultado del modelo basado en ML devuelve un valor exacto, que corresponde a una modulación específica, sin requerir algún proceso adicional.

Una vez analizados y seleccionados los tres parámetros: BER, modulación del sistema en tiempo real y estimación del canal, se establecen reglas y rangos para la selección de la modulación. Esto garantiza que el BER sea menor al requerido, 0.001. Con estas reglas y rangos de los parámetros, se genera una base de datos de entrenamiento precisa que refleja las condiciones a implementarse en el sistema, asegurando así que los modelos de ML se entrenen correctamente.

Tanto el OFDM adaptativo convencional como el adaptativo con ML tienen un buen rendimiento en los dos primeros escenarios. Sin embargo, al adicionar otros parámetros de decisión que permitan seleccionar la modulación óptima, como el número de portadoras activas y la tasa de codificación puede permitir una mayor precisión en la adaptación de la modulación para lograr un BER óptimo menor a 0.001. De acuerdo a los resultados, es posible concluir que OFDM adaptativo con ML alcanza un mejor rendimiento, en términos del BER, al OFDM adaptativo convencional.

Es importante notar que en el primer escenario, las técnicas de regresión lineal y vecinos más cercanos K-Nearest Neighbor (KNN) generaron predicciones incorrectas del índice de modulación, situación producida por los errores en la estimación del canal para valores de SNR menores a 18 [dB], donde la estimación varía considerablemente del valor real, lo que afecta negativamente su rendimiento en comparación con la técnica del árbol de decisión. Esta última mostró una mayor precisión en la clasificación de los datos del canal, permitiendo una mejor adaptabilidad del sistema OFDM a las condiciones variables del canal. Por lo tanto, la implementación del árbol de decisión en sistemas OFDM adaptativos ofrece una solución más robusta y eficiente.

Al analizar los resultados de los tres escenarios, se concluyó que, para canales con una variación alta en el tiempo, la adaptabilidad de la modulación debe considerar únicamente modulaciones más robustas [BPSK QPSK 16-QAM], para evitar picos altos de BER como se observó

en el escenario tres.

6.2. Recomendaciones

En los experimentos realizados, se pudo notar que la estimación precisa del canal es un factor importante para que las técnicas de Machine Learning (ML) realicen una predicción o clasificación de los parámetros de las modulaciones acorde a las condiciones precisas del canal, en tiempo real. Por lo que se recomienda implementar técnicas de estimación de canal más precisas y que den mejores resultados.

Las técnicas de *Machine Learning* (ML) supervisadas permiten lograr una predicción o la clasificación de la modulación óptima para el sistema planteado. En este contexto, se recomienda utilizar un enfoque semi supervisado para canales muy variantes como se plantea en el tercer escenario. Con esto el modelo de ML se entrena con la base de datos de entrenamiento y a la vez aprende del sistema de comunicaciones, lo que puede resultar en mejores predicciones del índice de modulación.

6.3. Trabajos futuros

En el presente trabajo de integración curricular se demostró que el uso de técnicas de aprendizaje automático ML mejora el rendimiento en sistemas de comunicación OFDM, a pesar que la estimación del canal posee una baja precisión del SNR. En futuros trabajos, se recomienda investigar y desarrollar técnicas avanzadas de estimación del canal que permitan una mayor precisión en la caracterización del canal en sistemas OFDM. Esto puede incluir métodos basados en aprendizaje profundo.

Obteniendo una mejor estimación del canal se puede predecir el comportamiento como también el cálculo del BER para cada subportadora, esto permitirá realizar la selección de la modulación a nivel de subportadoras, optimizando así el rendimiento del sistema OFDM.

Machine Learning permite predecir el índice de la modulación más óptimo para un sistema de comunicaciones, en función del entrenamiento del modelo a través de la base de datos. Es posible crear esta base de datos mediante reglas y rangos en función del SNR y el BER del modelo del canal. Se propone considerar parámetros adicionales como el número de portadoras activas por las cuales se transmite la información y la tasa de codificación, para la base

de datos utilizada en el entrenamiento del modelo de ML supervisado. Esto permitirá que el modelo sea más robusto y capaz de adaptarse a una mayor variedad de condiciones de canal

Referencias

[1] L. Jiménez Cano, J. J. Parrado Galeano, C. A. Quiza Montaña, y C. A. Suárez Fajardo, “Modulación multiportadora OFDM.”

[2] G. En, I. De, y S. De. (2012, 07) Sistemas OFDM de alta eficiencia espectral. Accessed: Oct. 24, 2023. [En línea]. Disponible: https://e-archivo.uc3m.es/handle/10016/16948

[3] S. Hong, Y. Zhang, Y. Wang, H. Gu, G. Gui, y H. Sari, “Deep learning-based signal modulation identification in OFDM systems,” *IEEE Access*, vol. 7, pp. 114 631–114 638, 2019.

[4] S. A. Çolak, Y. Acar, y E. Basar, “Adaptive dual-mode OFDM with index modulation,” *Physical Communication*, vol. 30, pp. 15–25, 10 2018.

[5] ——, “Adaptive dual-mode OFDM with index modulation,” *PHYSICAL COMMUNICATION*, vol. 30, pp. 15–25, 10 2018.

[6] A. Czylwik, “Adaptive OFDM for wideband radio channels,” in *Conference Record / IEEE Global Telecommunications Conference*, vol. 1, 1996, pp. 713–718.

[7] D. Lakanchanh, S. Noppanakeepong, S. Yoshizawa, y Y. Miyanaga, “Adaptive OFDM modulation techniques over fading channel for wireless communication systems with an 80 mhz bandwidth,” in *INEC 2010 - 2010 3rd International Nanoelectronics Conference, Proceedings*, 2010, pp. 1441–1442.

[8] E. M. Meza y J. B. Aspiazu, “Design of a power line communications transceiver based on OFDM,” *Procedia Technology*, vol. 17, pp. 107–113, 2014.

[9] S. Barua, Y. Rong, S. Nordholm, y P. Chen. (2019) Adaptive modulation for underwater acoustic OFDM communication. Accessed: Oct. 25, 2023. [En línea]. Disponible: https://ieeexplore.ieee.org/abstract/document/8867411/

[10] X. Fang, N. Ying, y X. Yu, “Adaptive transmission scheme for OFDM system,” in *2012 International Conference on Wireless Communications and Signal Processing, WCSP 2012*, 2012.

[11] M. Serda y et al., “A review on adaptive modulation and coding techniques for OFDM,” *International Journal for Scientific Research and Development*, vol. 3, num. 1, pp. 179–182, 04 2015.

[12] L. Goldfeld, “Adaptive OFDM system in a fading channel,” *Transactions on Emerging Telecommunications Technologies*, vol. 14, num. 3, pp. 293–300, 05 2003.

[13] C. Bin Ha, Y. H. You, y H. K. Song, “Machine Learning model for adaptive modulation of multi-stream in MIMO-OFDM system,” *IEEE Access*, vol. 7, pp. 5142–5152, 2019.

[14] M. A. M. A. B. A. Jebur, S. H. Alkassar y C. C. Tsimenidis, “Efficient Machine Learning enhanced channel estimation for OFDM systems,” *IEEE Access*, vol. 9, p. 100839–100850, 2021.

[15] S. Haykin, “Digital communications,” *New York*, 1988.

[16] B. P. Lathi, *Communication systems*. Wiley New York, 1968.

[17] N. Pambudiyatno, B. Harianto, y A. Mauludiyanto, “OFDM simulation using gnuradio on dynamic channels,” in *International Conference on Advance Transportation, Engineering, and Applied Science (ICATEAS 2022)*. Atlantis Press, 2023, pp. 100–110.

[18] E. E. Ordeñana Garofalo, “Análisis comparativo de los canales awgn/rayleigh/rician utilizados en sistemas OFDM.” 2020.

[19] H. Mrabet, E. Giacoumidis, y I. Dayoub, “A survey of applied Machine Learning techniques for optical OFDM based networks,” 05 2021.

[20] J. I. Bagnato, “Aprende Machine Learning,” *Obtenido de Crear una sencilla red nueronal: https://www. aprendemachinelearning. com/una-sencilla-red-neuronalen-python-con-keras-y-tensorflow*, 2019.

[21] C. Artacho Gómez, “Desarrollo y aplicación de técnicas de Machine Learning para la predicción de contagios por covid-19,” 2021.

[22] R. S. i Sole, “Clasificación de árboles de decisión,” 2023, accedido: 11 de junio de 2024. [En línea]. Disponible: https://openaccess.uoc.edu/bitstream/10609/138187/22/Data%20mining_M%C3%B3dulo%203_Clasificaci%C3%B3n%2C%20%C3%A1rboles%20de%20decisi%C3%B3n.pdf

[23] C. Arana, “Modelos de aprendizaje automoático mediante árboles de decisión,” Serie Documentos de Trabajo, Tech. Rep., 2021.

[24] P. Cunningham y S. J. Delany, “K-Nearest neighbour classifiers: (with python examples),” *arXiv preprint arXiv:2004.04523*, 2020.

[25] H. Rui, L. Xiaoping, C. Shidong, C. Xiangqun, L. Jun, y C. Peng, “An Adaptive OFDM Modulation Method for Middle Voltage Power Line communication,” in *International Con-*

ference on Speech Technology and Human-Computer Dialogue, vol. 2019-October, Jan. 2018, pp. 936–940.

[26] M. Asha y P. SurekhaT, “Design of OFDM system using adaptive modulation and sparse channel estimation method for long distance underwater acoustic communication,” *Social Science Research Network*, Jul. 2021.

[27] J. Faezah y K. Sabira, “Adaptive modulation for OFDM systems,” *International Journal of Communication Networks and Information Security*, 2009, accessed: Oct. 25, 2023. [En línea]. Disponible: https://www.academia.edu/download/67893373/Adaptive_modulation_for_{OFDM}_systems20210706-23047-xg9clz.pdf

[28] B. Reddy y L. Boppana, “Adaptive modulation and coding with channel state information in OFDM for wimax,” *International Journal of Image, Graphics and Signal Processing*, vol. 7, pp. 61–69, 12 2014.

[29] A. Kumar, K. K. Srinivas, y S. Majhi, “Automatic modulation classification for adaptive OFDM systems using convolutional neural networks with residual learning,” *IEEE Access*, vol. 11, pp. 61 013–61 024, 2023.

[30] I. Ram y M. Zia, “Adaptive modulation and coding with selective retransmission under OFDM signaling,” *Wireless Personal Communications*, vol. 101, 08 2018.

[31] X. Zhang y G. Tao, “The research of adaptive modulation technology in OFDM system,” in *Lecture Notes in Electrical Engineering*, 2022, vol. 942 LNEE, pp. 739–748.

[32] B. Jdid, K. Hassan, I. Dayoub, W. H. Lim, y M. Mokayef, “Machine Learning based automatic modulation recognition for wireless communications: A comprehensive survey,” *IEEE Access*, vol. 9, pp. 57 851–57 873, 2021.

[33] Y. Ping *y otros*, “Adaptive spatial modulation mimo based on Machine Learning,” *IEEE Journal on Selected Areas in Communications*, vol. 37, num. 9, 2019.

[34] S. Lin, B. Zheng, G. C. Alexandropoulos, M. Wen, F. Chen *y otros*, “Adaptive transmission for reconfigurable intelligent surface-assisted OFDM wireless communications,” *IEEE Journal on Selected Areas in Communications*, vol. 38, num. 11, pp. 2653–2665, 2020.

[35] H. Yigit y A. Kavak, “Adaptation using neural network in frequency selective MIMO-OFDM systems,” in *IEEE 5th International Symposium on Wireless Pervasive Computing 2010*. IEEE, 2010, pp. 390–394.

[36] J. Zhang, C.-K. Wen, S. Jin, y G. Y. Li, “Artificial intelligence-aided receiver for a cp-free ofdm system: Design, simulation, and experimental test,” *IEEE Access*, vol. 7, pp. 58 901–58 914, 2019.

[37] H. Liu, S. Lu, M. El-Hajjar, y L.-L. Yang, “Machine Learning assisted adaptive index modulation for mmwave communications,” *IEEE Open Journal of the Communications Society*, vol. 1, pp. 1425–1441, 2020.

[38] A. Rico-Alvarino y R. W. Heath, “Learning-based adaptive transmission for limited feedback multiuser MIMO-OFDM,” *IEEE transactions on wireless communications*, vol. 13, num. 7, pp. 3806–3820, 2014.

[39] B. Reddy y L. Boppana, “Adaptive modulation and coding with channel state information in OFDM for wimax,” *International Journal of Image, Graphics and Signal Processing*, vol. 7, pp. 61–69, 12 2014.

[40] A. GOLDSMITH, “Wireless communications,” 2005.

[41] M. A. Fernández, “Implementación de técnicas de estimación y sincronización para sistemas OFDM,” *Universitat Autònoma de Barcelona.*, pp. 20–30, 9 2008.

[42] K. Venkatesh, G. Nair, y S. Kulkarni, “Implementation of a Machine Learning based OFDM detector using gnu radio,” in *2021 International Conference on Communication information and Computing Technology (ICCICT)*. IEEE, 2021, pp. 1–5.

7. Anexo

Anexo A: Código Matlab - Árbol de Decisión

```
Decision tree for regression
 1  if selected_modulation<24 then node 2 elseif selected_modulation >=24 then node 3 else 25.4227
 2  if selected_modulation<12 then node 4 elseif selected_modulation >=12 then node 5 else 12.0358
 3  if current_ber<5e-05 then node 6 elseif current_ber>=5e-05 then node 7 else 45.592
 4  if current_ber<2.49929e-05 then node 8 elseif current_ber >=2.49929e-05 then node 9 else 7.12131
 5  if current_ber<5.00048e-05 then node 10 elseif current_ber >=5.00048e-05 then node 11 else 21.9442
 6  if SNR<23.9948 then node 12 elseif SNR>=23.9948 then node 13 else 55.6489
 7  fit = 4
 8  if selected_modulation<6 then node 14 elseif selected_modulation >=6 then node 15 else 10.6052
 9  if selected_modulation<6 then node 16 elseif selected_modulation >=6 then node 17 else 5.46962
10  if SNR<21.0008 then node 18 elseif SNR>=21.0008 then node 19 else 26.1145
11  fit = 4
12  if selected_modulation<48 then node 20 elseif selected_modulation>=48 then node 21 else 46.4373
13  fit = 64
14  if SNR<15.0043 then node 22 elseif SNR>=15.0043 then node 23 else 7.34013
15  if SNR<18.0045 then node 24 elseif SNR>=18.0045 then node 25 else 13.8834
16  if current_ber<2.50157e-05 then node 26 elseif current_ber >=2.50157e-05 then node 27 else 4.00121
17  if current_ber<4.99964e-05 then node 28 elseif current_ber
```

```
    ↪ >=4.99964e-05 then node 29 else 6.86368
18  fit = 16
19  fit = 32
20  fit = 32
21  if current_ber<1.25017e-05 then node 30 elseif current_ber
    ↪ >=1.25017e-05 then node 31 else 60.6644
22  fit = 4
23  fit = 8
24  fit = 8
25  fit = 16
26  fit = 6
27  fit = 4
28  fit = 8
29  fit = 4
30  fit = 32
31  fit = 64
```