Sudan University of Science and Technology

College of Engineering

Electronics Engineering Department

MACHINE LEARNING BASED ADAPTIVE MODULATION AND CODING IN 5G NETWORKS

A Research Submitted in Partial fulfillment for the Requirements of the Degree of B.Sc. (Honors) in Electronics Engineering

(B.eng)

Prepared by:

1. Hamsa Hassan Suliman Ahmed
2. Mozan Mudathir Mohammed Khalid
3. Heyam Sid Ahmed Eltayeb Elhaj
4. Walaa Tarig albahi Eltayeb

Supervisor:

Prof. Dr. Rashid A Saeed

January 2025

الإستهلال

قال تعالي:

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

(اللَّهُ لَا إِلَهَ إِلَّا هُوَ الْحَيُّ الْقَيُّومُ لَا تَأْخُذُهُ سِنَةٌ وَلَا نَوْمٌ لَّهُ مَا فِي السَّمَاوَاتِ وَمَا فِي الْأَرْضِ مَن ذَا الَّذِي يَشْفَعُ عِندَهُ إِلَّا بِإِذْنِهِ يَعْلَمُ مَا بَيْنَ أَيْدِيهِمْ وَمَا خَلْفَهُمْ وَلَا يُحِيطُونَ بِشَيْءٍ مِّنْ عِلْمِهِ إِلَّا بِمَا شَاءَ وَسِعَ كُرْسِيُّهُ السَّمَاوَاتِ وَالْأَرْضَ وَلَا يَئُودُهُ حِفْظُهُمَا وَهُوَ الْعَلِيُّ الْعَظِيمُ)

[سورة البقرة: آيه 255]

DEDICATION

To our parents, who directed our first steps in the right way…

To our brothers and sisters…

To our friends who are always ready to help…

To our colleagues…

We dedicate this work…

ACKNOLEDGMENTS

Our grateful thanks firstly to Allah who guided us to the right way in our life. Then many thanks and appreciations are extended to our supervisor Dr. Rashid A Saeed for his valuable advice and endless effort to make this work come into reality. Our heartfelt gratitude is also extended to our families for their understanding, support and love throughout this project. Lastly, we would like to thank everyone who has contributed to this project in any way. Your help and support have been invaluable and we are truly grateful.

ABSTRACT

In wireless communication, resources like bandwidth and energy are scarce and extremely valuable, any system should serve as many users as possible while preserving high Quality of Service (QoS) for the best user experience. Accordingly, the Base Station (BS) has the responsibility to optimally schedule its resources to the users based on the available information. Consequently, the whole process of scheduling is truly demanding and requires high complex calculations from the overall system. Hence, the request of more sophisticated and effective methods is substantial in order to minimize the challenges of scheduling. This bachelor's thesis focuses on the Modulation and Coding Scheme (MCS) selection in a Time Division Duplex (TDD) based mobile network. The main objective is the simplification and optimization of the downlink process at the base station by predicting the MCS index for a single User Equipment (UE), using Machine Learning (ML). The developed machine learning algorithms is in accordance with the LTE-Advanced Pro (release 12, 13,14) lookup tables and is based on similar parameters. For a given frame, this thesis targets predicting the MCS index of future subframes. Thus, the resource allocation process for independent users is becoming quicker and easier for the BS. The results are based on laboratory measurements at Ericsson, where the collection of data logs for several stationary UEs, occurred on a network testing environment

and their different cell characteristics investigated thoroughly. Concluding, the accuracy level which the ML classification algorithm achieved was approximately 75 percent. Therefore, the prediction accuracy can be described as sufficient for the BS to decrease the computation complexity and energy consumption during the downlink process. The data logs that the project took into account cannot be generalized for real-time scenarios as it is explained in detail finally.

المستخلص

في الاتصالات اللاسلكية، الموارد مثل النطاق الترددي والطاقة نادرة وقيمة للغاية، يجب أن يخدم أي نظام أكبر عدد ممكن من المستخدمين مع الحفاظ على جودة الخدمة العالية (QoS) لأفضل تجربة للمستخدم. وفقًا لذلك، تتحمل المحطة الأساسية مسؤولية جدولة مواردها للمستخدمين على النحو الأمثل بناءً على المعلومات المتاحة. وبالتالي، فإن عملية الجدولة بأكملها تتطلب الكثير من الجهد وتتطلب حسابات معقدة للغاية من النظام الإجمالي. وبالتالي، فإن طلب طرق أكثر تطوراً وفعالية أمر كبير من أجل تقليل تحديات الجدولة. تركز هذه الأطروحة على اختيار مخطط التعديل والترميزMCS في شبكة الهاتف المحمول القائمة على دوبلكس بتقسيم الزمن TDD.الهدف الرئيسي هو تبسيط وتحسين عملية التنزيل في المحطة الأساسية من خلال التنبؤ بمؤشر مخطط التعديل والترميز لمعدات مستخدم واحدة (UE)، باستخدام التعلم الآلي(ML).تتوافق خوارزميات التعلم الآلي المطورة مع جداول البحث LTE-Advanced Proالإصدار 12 و13 و14 وتستند إلى معلمات مماثلة. بالنسبة لإطار معين، تستهدف هذه الأطروحة التنبؤ بمؤشر MCS للإطارات الفرعية المستقبلية. وبالتالي، أصبحت عملية تخصيص الموارد للمستخدمين المستقلين أسرع وأسهل بالنسبة لمحطة القاعدة. وتستند النتائج إلى القياسات المعملية في شركة إريكسون، حيث تم جمع سجلات البيانات لعدة أجهزة مستخدمين ثابتة، في بيئة اختبار الشبكة وتم التحقيق في خصائص خلاياها المختلفة بدقة. وفي الختام، كان مستوى الدقة الذي حققته خوارزمية تصنيف التعلم الآلي حوالي 75 بالمائة. وبالتالي، يمكن وصف دقة التنبؤ بأنها كافية لمحطة القاعدة لتقليل تعقيد الحساب واستهلاك الطاقة أثناء عملية التنزيل. لا يمكن تعميم سجلات البيانات التي أخذها المشروع في الاعتبار لسيناريوهات الوقت الفعلي كما هو موضح بالتفصيل في النهاية.

Table of Contents

LIST OF FIGURES

LIST OF TABELS

ABBREVIATIONS

3GPP	3rd Generation Partnership Project
5G	5th Generation
AI	Artificial Intelligence
AMC	Adaptive Modulation and Coding
BS	Base Station
CQI	Channel Quality Indicator
DL	Downlink
eNodeB	Evolved Node B
FSK	Frequency Shift Keying
HARQ	Hybrid Automatic-Repeat-Request (ARQ)
IoT	Internet of Things
k-NN	k-Nearest Neighbor
LTE	Long Term Evolution
MCS	Modulation and Coding Scheme
MIMO	Multiple-Input-Multiple-Output
ML	Machine Learning
NDF	New Data Frame
NN	Neural Networks
PAM	Pulse Amplitude Modulation
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
SINR	Signal to Interference and Noise Ratio
SVM	Support Vector Machines
UE	User Equipment

Chapter One

INTRODUCTION

Chapter One

INTRODUCTION

1.1 Overview:

Adaptive modulation and coding (AMC) is a technique used in wireless communications systems to improve the efficiency and reliability of data transmission. It dynamically adjusts the modulation scheme and error correction code rate based on the current conditions of the communication channel, such as signal strength, noise level, and interference. We find that ML techniques are more performant than other methods, as they are very suitable for describing the structure of mobile communications networks and the dynamic environment. Now ML is one of the important issues in mobile wireless networks outside of 5th generation (B5G) to improve power control, link adaptation, capacity, and traffic scheduling and routing. By incorporating machine learning techniques, the accuracy and efficiency of AMC are enhanced, thus improving adaptive modulation and coding schemes in 5G networks.

1.2 Problem Statement:

Traditional AMC methods rely on predefined rules and heuristics to adjust modulation schemes and coding rates based on channel conditions. These methods may not adapt quickly or accurately to dynamic changes in network environments, leading to suboptimal performance. The problem addressed by this research is the need for a

more intelligent and adaptive AMC approach that leverages ML to better handle varying channel conditions and improve overall network performance.

1.3 Aims and Objectives:

The main aims of this work to explore and implement ML-based techniques for adaptive modulation and coding in 5G networks to enhance network performance and efficiency.

The objectives of the project are:

1. To review existing AMC methods and identify limitations in their adaptability and performance.
2. To develop ML models that predict channel quality and optimize AMC parameters in real-time.
3. To evaluate the performance of ML-based AMC compared to traditional methods using simulation or real-world data.
4. To analyze the impact of ML-based AMC on network throughput, reliability, and overall efficiency.

1.4 Scope:

The research focuses on the application of ML algorithms to AMC in the context of 5G networks. It includes Reviewing current AMC techniques and their limitations, Developing and testing ML models for predicting channel conditions and adjusting AMC parameters, Comparing the performance of ML-based AMC with traditional approaches in terms of data throughput, reliability, and network

efficiency. The study may use simulations or experimental data from 5G network environments to validate the proposed methods.

1.5 Methodology:

Literature Review a comprehensive review of current AMC techniques and machine learning applications in telecommunications. Model development design and develop machine learning models designed to predict channel quality and optimize AMC parameters. This includes choosing appropriate machine learning algorithms, training models with historical data, and tuning their performance. Simulation/Experimental setup implementing machine learning-based AMC models in a simulated environment or using real-world data from 5G networks. Performance Evaluation the performance of machine learning-based AMC is compared with traditional AMC methods using metrics such as data throughput, error rates, and overall network efficiency. Analysis results are analyzed to evaluate the effectiveness of machine learning-based AMC and identify potential improvements or future research directions.

1.6 Thesis Layout:

This thesis composed of five chapters, Chapter two gives presents background information related to the AMC Techniques in addition Machine Learning in Telecommunication and Existing ML-based AMC Approaches. Chapter three covers a detailed overview of the architecture, design, simulation and Experimental setup of AMC and ML Techniques, Moreover development of ML Models. Chapter four it presents and describes the performance analysis of ML-based AMC the

results archived by this work are compared with Traditional AMC Methods, ln addition impact on network performance. Chapter five concludes the thesis, implications for 5G Networks and recommendations for future research.

Chapter Two

Literature Review

Chapter Two

Literature Review

2.1 Background:

2.1.1 Modulation:

In telecommunications, modulation is the process of conveying a message signal, for example a digital bit stream or an analog audio signal, inside another signal that can be physically transmitted. Modulation of a sine waveform transforms a baseband message signal into a passband signal.

A modulator is a device that performs modulation. A demodulator (sometimes detector or demod) is a device that performs demodulation, the inverse of modulation. A modem (from modulator–demodulator) can perform both operations.

Modulation is the addition of information to an electronic or optical carrier signal. A carrier signal is one with a steady waveform -- constant height (amplitude) and frequency. Information can be added to the carrier by varying its amplitude, frequency, phase, polarization (for optical signals), and even quantum-level phenomena like spin.

There are two types of modification:

2.1.1.1 Analog Modulation:

Process of transferring an analog baseband (low frequency) signal, like an audio or TV signal over a higher frequency signal such as a radio frequency band.

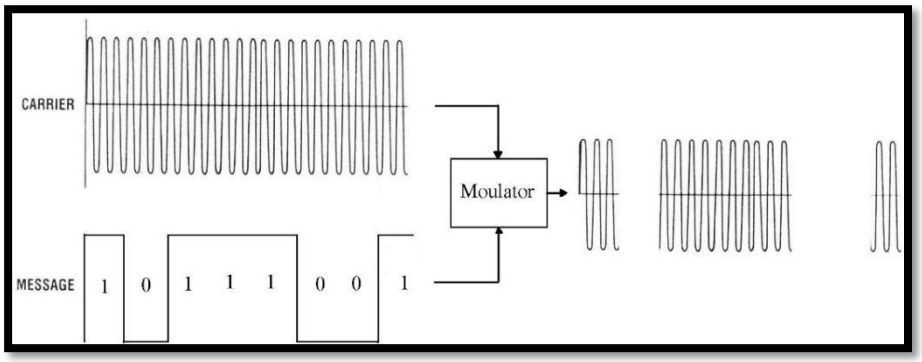


Figure 2-1: Analog Modulation

2.1.1.2 Digital Modulation:

Process of encoding a digital information signal into the amplitude phase, or frequency of the transmitted signal. This encoding affects the

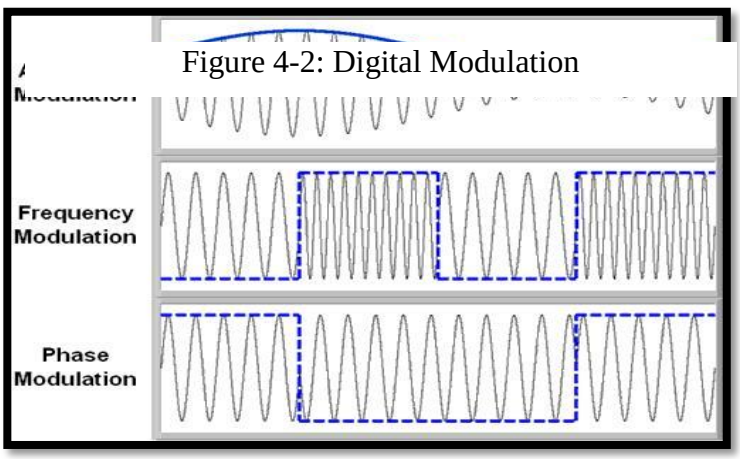


Figure 4-2: Digital Modulation

bandwidth of the signal and its ability to withstand channel impairments.

2.1.1.3 Modulation Techniques:

2.1.1.3.1 Quadrature Phase Shift Keying (QPSK):

Is a form of Phase Shift Keying in which two bits are modulated at once, selecting one of four possible carrier phase shifts (0, 90, 180, or 270 degrees). QPSK allows the signal to carry twice as much information as ordinary PSK using the same bandwidth. QPSK is used

for satellite transmission of MPEG2 video, cable modems, videoconferencing, cellular phone systems, and other forms of digital communication over an RF carrier.

Symbol	**Phase Shift**
00	0 Degrees
01	90 Degrees
10	180 Degrees
11	270 Degrees

2.1.1.3.2 Quadrature Amplitude Modulation(QAM):

Is a method of combining two amplitude-modulated (AM) signals into a single channel, thereby doubling the effective bandwidth . QAM is used with pulse amplitude modulation (PAM) in digital systems, especially in wireless applications.

QAM bits per symbol

MODULATION	BITS PER SYMBOL	SYMBOL RATE
BPSK	1	1*bit rate
QPSK	2	1/2bit rate
8PSK	3	1/3bit rate
16QAM	4	1/4bit rate
32QAM	5	1/5bit rate
64QAM	6	1/6bit rate

2.1.2 Coding:

Process of detects and corrects errors.

2.1.3 Adaptive Modulation:

Is a technique that allows for the adjustment of transmission characteristics and waveforms, enabling improved spectrum access and more efficient use of spectrum while working alongside other signals. It enables a cognitive radio to select the suitable modulation type for seamless interoperability within a specific transmission system.

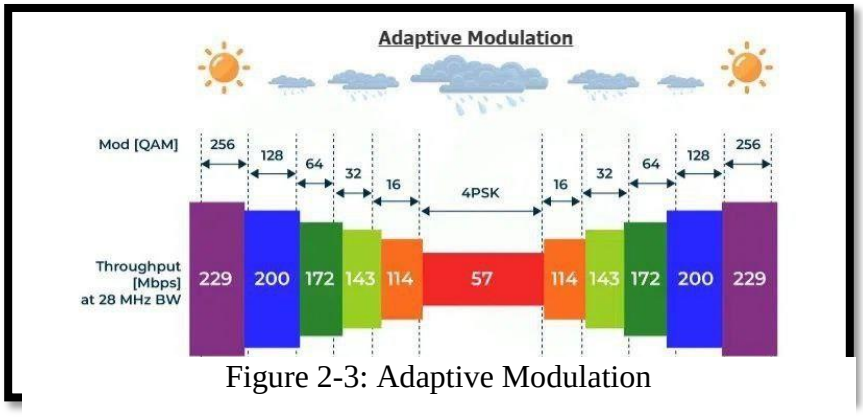


Figure 2-3: Adaptive Modulation

2.1.3.1 Adaptive waveform techniques:

Techniques belonging to this category focus on modifying the parameters of the signal structure used by the system, whenever the link quality is degraded. They can be further classified into three categories: Adaptive coding (ADCOD), Adaptive modulation (ADMOD) and Data Rate Adaptation (DRA).

2.1.3.1.1 Adaptive coding (ADCOD):

When a link is experiencing fading, the introduction of additional redundant bits to the information bits to improve error correction capabilities (FEC) allows to maintain the nominal BER while leading to a reduction of the required energy per information bit. Adaptive coding consists in implementing variable coding rate in order to match impairments due to propagation conditions. A gain of varying from 2 to 10 dB can be achieved depending on the coding rate. The limitations of this fade mitigation technique are linked to additional bandwidth requirements for FDMA and larger bursts in the same frame for TDMA. Adaptive coding at constant information data rate then translates in a reduction of the total system throughput when various links are experiencing fading simultaneously.

2.1.3.1.2 Adaptive modulation (ADMOD):

Under clear sky conditions, high system capacity for a specified bandwidth can be achieved by using modulation schemes with high spectral efficiency, such as coded modulation or combined amplitude and phase modulation. In case of fading, the modulation schemes could be changed to implement more robust modulations requiring less symbol energy. As for adaptive coding, the aim of the adaptive modulation technique is to decrease the required bit energy per noise power spectral

density ratio (Eb/No) corresponding to a given BER, by using a lower level modulation scheme at the expenses of a reduction of the spectral efficiency.

2.1.3.1.3 Data Rate Adaptation (DRA):

With data rate reduction, the information data rate is decreased when the link experiences fading, and this translates in a decrease by the same amount of the required carrier power to noise power spectral density ratio (C/No) if the required bit energy per noise power spectral density ratio (Eb/No) is kept constant (no change in the coding gains and constant BER). The transmitted bit rate is reduced accordingly and turns into a similar reduction of the occupied carrier bandwidth. Operation at constant resource by keeping a constant transmitted data rate is also possible by adjusting the coding rate accordingly. In that case, the coding gain adds to the reduction of the information data rate.

This fade mitigation technique requires services that can tolerate a reduction of the information rate such as video or voice transmission (assuming a change of the source coding at the expense of a reduction of the perceived quality), and data transmission (assuming an increase of transfer duration or a reduced throughput if Internet access). Moreover, extra delay and/or complexity may be required due to the exchange of signaling between the transmitter and the receiver.

2.1.3.2 Goal of Adaptive Modulation:

The goal of Adaptive Modulation is to improve the operational efficiency of Microwave links by increasing network capacity over the existing infrastructure - while reducing sensitivity to environmental interferences. Adaptive Modulation means dynamically varying the modulation in an errorless manner in order to maximize the throughput

under momentary propagation conditions. In other words, a system can operate at its maximum throughput under clear sky conditions, and decrease it gradually under rain fade.

2.1.3.3 Advantage of adaptive modulation:

Adaptive modulation systems improve rate of transmission, and/or bit error rates, by exploiting the channel state information that is present at the transmitter. Especially over fading channels which model wireless propagation environments, adaptive modulation systems exhibit great performance enhancements compared to systems that do not exploit channel knowledge at the transmitter.

2.1.4 Adaptive Modulation and Coding:

Adaptive Coding and Modulation is a statistical, non-static advantage that enables dynamic changes in user throughput. Benefits and value vary over time and are not guaranteed, but are predictable. ACM technology converts link margin to an increase in the data throughput of satellite links. Only non-synchronous data networks (such as Ethernet packet-based networks) can take advantage of a dynamic data throughput rate. All satellite links are designed to function at a certain annual availability. The closer to 100% we demand of our link availability, the more link margin we need to meet this demand and thus the greater the value of ACM.

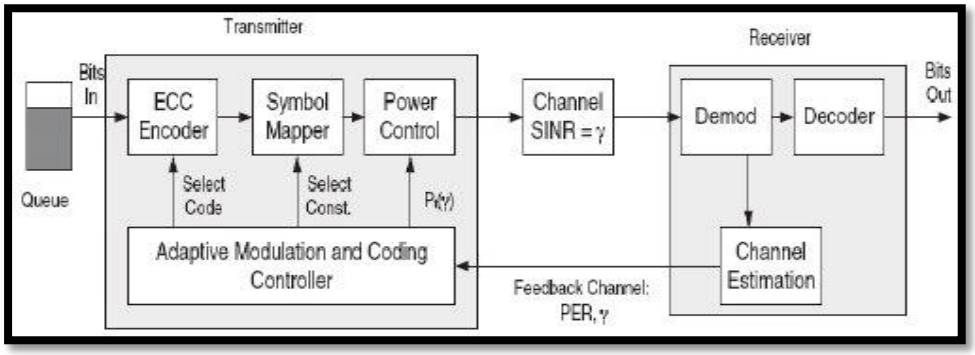


Figure 2-4: Adaptive Modulation and Coding

2.1.4.1 Basics of Modulation and Coding in Communication:

At the heart of wireless communication lies the concept of modulating digital data onto analog carrier signals for transmission. Modulation schemes determine how information is encoded into the amplitude, frequency, or phase of the carrier signal. This encoded signal is then transmitted through the airwaves to the receiver. On the receiving end, error correction codes are used to detect and correct errors introduced during transmission, enhancing the reliability of data reception.

2.1.4.2 Evolution of AMC in 5G Networks:

The concept of AMC has gained prominence in 5G networks as a fundamental feature. It builds upon the adaptive modulation techniques that were present in previous generations but takes them to new heights in terms of efficiency and sophistication. In 5G, AMC is not just an option but a necessity to harness the potential of the high-frequency bands and the diverse range of usage scenarios.

2.1.4.3 Principles of Adaptive Modulation and Coding:

At the core of AMC is the ability to dynamically adjust the modulation scheme and coding rate in real-time based on the current wireless channel conditions. The decisions regarding modulation and coding are linked to channel parameters such as Signal-to-Noise Ratio (SNR), Bit Error Rate (BER), and Channel Quality Indicator (CQI). When the channel conditions are

favourable , higher-order modulation schemes and more aggressive coding rates can be employed to achieve higher data rates. Conversely, in challenging conditions, the system can switch to more robust schemes to ensure reliable transmission.

2.1.4.4 AMC Enhances 5G Network Performance With Multiple Benefits:

1.Optimized Spectrum Use And Increased Throghput:

By adapting modulation and coding based on channel conditions, AMC ensures that the available spectrum is used optimally, resulting in higher data rates and improved spectral efficiency.

2. Improved User Experience:

AMC adapts to the dynamic nature of wireless channels, ensuring consistent and reliable connections. This translates to smoother streaming, faster downloads, and reduced dropped calls, enhancing the user experience.

3. Robust Communication in Challenging Environments:

AMC's ability to adjust to varying channel conditions makes it particularly valuable in challenging scenarios, such as urban areas with high interference or remote locations with weak signal strength.

2.1.4.5 AMC Implementation in 5G Networks:

AMC is seamlessly integrated into the fabric of 5G networks:

1. Integration with MIMO Systems:

Adaptive Modulation and Coding works in synergy with Multiple-Input, Multiple-Output (MIMO) systems, which use multiple antennas for data transmission. Together, they enhance throughput, reliability, and spatial efficiency.

2. Collaborative Role in Hybrid Beamforming:

AMC plays a critical role in hybrid beamforming, where both digital and analog beamforming techniques are used. It ensures that the modulation and coding align with the beamforming strategies, optimizing overall performance.

3. Contributions to Different Usage Scenarios:

AMC caters to the diverse needs of 5G usage scenarios. For Enhanced Mobile Broadband (eMBB), AMC maximizes data rates. In Ultra-Reliable Low Latency Communication (URLLC), it ensures critical data is transmitted reliably. In Massive Machine-Type Communication (mMTC), AMC adapts to the requirements of IoT devices.

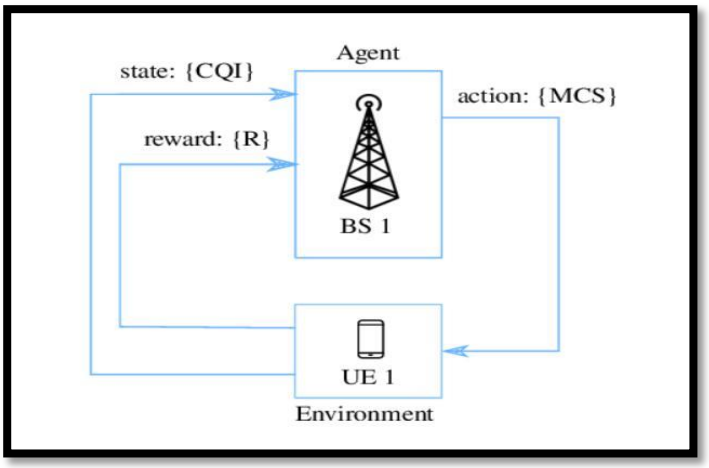


Figure 2-5: AMC Implementation in 5G Networks

2.1.4.6 Machine Learning and AI in AMC:

The integration of Machine Learning (ML) and Artificial Intelligence (AI) techniques has enhanced the capabilities of AMC:

1. Real-Time Channel Prediction:

AI algorithms can predict future channel conditions based on historical data. This predictive capability empowers AMC to anticipate changes and adapt proactively.

2. Adaptive AMC Based on AI-Enhanced Analytics:

AI-driven analytics can provide insights into complex channel behaviors, enabling AMC to make more informed decisions about modulation and coding adjustments.

3. Advantages of AI-Driven AMC:

AI-enhanced AMC offers faster adaptation to changing conditions and improved overall efficiency, as it leverages advanced data processing capabilities.

2.1.5 machine learning:

Machine learning is an application of AI that enables systems to learn and improve from experience without being explicitly programmed. Machine learning focuses on developing computer programs that can access data and use it to learn for themselves.

Similar to how the human brain gains knowledge and understanding, machine learning relies on input, such as training data or knowledge graphs, to understand entities, domains and the connections between them. With entities defined, deep learning can begin.

The machine learning process begins with observations or data, such as examples, direct experience or instruction. It looks for patterns in data so it can later make inferences based on the examples provided. The primary aim of ML is to allow computers to learn autonomously without human intervention or assistance and adjust actions accordingly.

The relation between AI, ML and deep learning is depicted in Fig2-6.

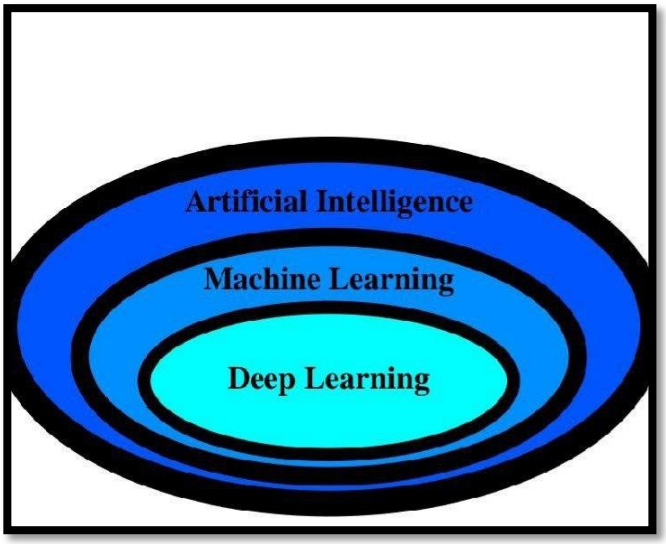


Figure 2-6: The Relation Between AI, ML and Deep Learning

2.1.5.1 Machine Learning Training Methods:

A. Supervised Learning:

Supervised machine learning algorithms apply what has been learned in the past to new data using labeled examples to predict future events. By analyzing a known training data set, the learning algorithm produces an inferred function to predict output values. The system can provide targets for any new input after sufficient training. It can also compare its output with the correct, intended output to find errors and modify the model accordingly.

Supervised techniques adapt the model to reproduce outputs known from a training set (e.g., recognize car types on photos). In the beginning, the system receives input data as well as output data. Its task is to create appropriate rules that map the input to the output. The training process should continue until the level of performance is high enough. After training, the system should be able to assign an output object which it has not seen during the training phase. In most cases, this process is really fast and accurate.

There are two types of Supervised Learning techniques:

A.Regression

B.Classification.

Classification separates the data; Regression fits the data.

Regression:

is a technique that aims to reproduce the output value. We can use it, for example, to predict the price of some product, like a price of a house in a specific city or the value of a stock. There is a huge number of things we can predict if we wish.

Classification:

is a technique that aims to reproduce class assignments. It can predict the response value and the data is separated into "classes". Examples? Recognition of a type of car in a photo.

The most commonly used types of classifications algorithms in machine learning are:

Naïve Bayes:

ML algorithms which take into consideration the principle of Maximum A Posteriori (MAP) for the classification of their problem.

Decision trees:

Approach that splits the training set into distinct nodes, where one node can include one, most of or all of the different categories that the database can be divided.

k-Nearest Neighbors (k-NN):

A non-parametric classification algorithm that measures the distance of the unknown input from every other training example.

Logistic regression:

A statistical ML technique which classifies the dataset records, based on the values of the input fields.

Support Vector Machines (SVM):

Are among the most robust algorithms which are based in the maximization of the minimum distance from the decision line, i.e. separator, to the nearest example.

Neural Networks (NN):

Are based upon the log likelihood function with respect to the network parameters, extending to the multiclass problem and ensuing as output a binary or N-ary result.

B. Unsupervised Learning:

Unsupervised machine learning algorithms are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. At no point does the system know the correct output with certainty. Instead, it draws inferences from data sets as to what the output should be.

In this Machine Learning technique, we do not have any outcome variables to predict. The computer is trained with unlabeled data. Unsupervised techniques aim to uncover hidden structures, like find groups of photos with similar cars, but it's a bit difficult to implement and is not used as widely as supervised learning.

Unsupervised techniques may be used as a preliminary step before applying supervised ones. The internal structure of the data may provide information on how to better reproduce outputs. And in unsupervised techniques, we have clustering and dimensional reduction.

C. Semi Supervise Learning:

Semi supervised learning, as the title describes, is the mixed version of the two different methods previously presented. These algorithms can operate with partially labeled data and the rest of its data, the majority in most of the cases, with unlabeled ones. A good example of semi supervised algorithms is the reinforcement learning, which is a method called an agent and it observes the environment, selecting its best action according to a policy defined by previous situations. These actions are resulting to rewards or penalties, whether there is a positive or negative outcome, respectively. semi supervised techniques are used to explore the capability of implementing ML in the mobile network of LTE systems.

2.1.6 Dynamic Resource Allocation:

Mobile communication systems exhibit significant variation in the number of UEs in a given cell, and in their time-varying channel conditions. Moreover, with the expansion of mobile applications, different UE have different requirements. Thus, resource allocation

became more complex as flexibility and dynamicity is required, while maintaining a resource and energy efficient system. This is done using the scheduler, implemented in the eNodeB in LTE, which allocates the available resource blocks according to the UE's need and condition.

2.1.6.1 Frame Structure:

In LTE, like several communication systems, data is exchanged in frames. A radio frame, depicted in Fig. , is composed of 10 subframes each of 1ms.This composition enables the synchronization of data between the eNodeB and the UE, where each transmitted packet contains the frame and subframe number.

2.1.6.2 Scheduler:

With the increasing number of active UEs, scheduling plays an essential role in the system, more specifically in the eNodeB, by allocating the available yet limited RBs to the correct UEs, thus increasing the efficiency of the system. Although the scheduling strategy is not standardized, different strategies are used by different vendors to provide the required QoS for the UEs.

When compared to older allocation methods like round robin or proportional queuing, the scheduler significantly increases the throughput inside the cell, even though it adds complexity and doesn't has a big advantage at the cell's edges.

2.1.6.3 Modulation and Coding Scheme:

Other than RBs allocation, the scheduler uses the available information to choose the UE's MCS, which directly affects the QoS. MCS is an index based on the channel quality indicator (CQI) sent by the

UE. The UE's UL report is used at subframe N to determine the MCS index for subframe N+2 using a lookup table.

Over the different LTE releases, the lookup tables for MCS selection have been upgraded for different modulation schemes. The number of MCS indexes was 15 at first, as shown in table 3 [15], and it reached 31 in the current version in table 4. Using several parameters like CQI, HARQ and User's data size, the MCS is chosen, and thus, the modulation scheme and coding rate are determined. This process should be accurate and fast, as an inaccurate MCS can result in a retransmission in case of bad channel condition, or inefficiency in case of good channel condition. Also, the MCS selection is done between subframes, so the decision should be fast, especially if several UEs are connected to the eNodeB.

Table 2-1: 3GPP CQI lookup table [28]

CQI index	Modulation	Code Rate x1024	Efficiency (bits/s)/Hz
0	Out of range		
1	QPSK	78	0.1523
2	QPSK	193	0.377
3	QPSK	449	0.877
4	16QAM	378	1.4766
5	16QAM	490	1.9141
6	16QAM	616	2.4063
7	64QAM	466	2.7305
8	64QAM	567	3.3223
9	64QAM	666	3.9023
10	64QAM	772	4.5234
11	64QAM	873	5.1152
12	256QAM	711	5.5547
13	256QAM	797	6.2266
14	256QAM	885	6.9141
15	256QAM	948	7.4063

Table 2-2: 3GPP MCS lookup table [28]

MCS Index	Modulation Order	MCS Index	Modulation Order
0	2	16	4
1	2	17	6
2	2	18	6
3	2	19	6
4	2	20	6
5	2	21	6
6	2	22	6
7	2	23	6
8	2	24	6
9	2	25	6
10	4	26	6
11	4	27	6
12	4	28	6
13	4	29	2
14	4	30	4
15	4	31	6

2.1.6.4 Scheduling Strategies:

The MCS is selected based on the UE's CQI report, which is directly related to SINR and link quality. The scheduler chooses the UE's MCS by comparing the UE's CQI to the lookup tables 3 and 4. However, as mentioned in 2.1.6.2, the scheduling strategy is not standardized and some of strategies used by the vendors are as follows:

1. Normal Strategy:

Using the estimated CQI, the MCS is selected based on the table's CQI equal to the estimated one. This strategy follows the lookup table and the MCS index is increased when an ACK is received from the UE, while it's decreased when a NACK is received.

2. Conservative Strategy:

The conservative strategy targets a correct reception of the transmitted data even if the efficiency is decreased. This strategy selects a lower MCS index than the estimated one, until it reaches the maximum value in the lookup table. When a NACK is received, the index is decreased. However, even if an ACK is received, an increase in the MCS index is done when the CQI increases by two indexes.

3. Aggressive Strategy:

Opposite to the conservative strategy, the aggressive strategy targets a higher efficiency in terms of bit rate over the possibility of incorrect reception. The selected MCS is higher than the estimated one until the SINR reaches its minimum values, where MCS = {0}. When an

ACK is received, the index is increased, and the MCS index decreases only when a NACK is received and the CQI index drops by two indexes. Based on the vendor and the applications used the UEs, the MCS strategy is selected to either increase the data rate by compromising a higher error rate, decrease the error rate by compromising a higher efficiency, or choose a balanced mix between of both efficiency and correct reception.

2.1.7 Duplex Transmission:

In mobile communication systems, the UE and BS exchange information using duplex transmission. Duplexing refers to data transmission in both directions, from the BS to the UE and vice versa. Two types of duplex transmission are available: Half Duplex (HD) and Full Duplex (FD).

FD enables both sides to transmit simultaneously using the same link, while in HD, transmission is only in one direction at a time. Both types are used in several applications, but HD is less complicated and requires less resources, as the same resource can be used in both directions while switching between the transmitter and receiver at given periods.

Time Division Duplex (TDD) and Frequency Division Duplex (FDD) are two duplexing methods used in cellular networks. TDD is HD based as the time domain is shared, but with small time intervals it can emulate FD. However, FDD is always FD given that at the equipment, BS or user side, the uplink and downlink frequencies are predefined.

2.1.7.1 Uplink(UL)

The transmission from the UE to the BS is called uplink (UL). The data sent from each UE includes information about the channel condition and status of the received data. In addition to the essential CQI report, the UE sends reference signals to help the BS estimate the channel state. Furthermore, in the case of an erroneous packet, the UE can request retransmission from the BS. Finally, the UE can request RB allocation from the BS for UL and/or DL transmissions.

2.1.7.2 Downlink (DL)

is the transmission from the BS to one or more UEs. In addition to the information gathered from the user's UL, the BS sends reference signals to provide the connected UEs with channel conditions. Also, the base station sends control signals to provide specified UL slots for each of the UEs.

2.1.8 LTE Transmitter and Receiver:

The block diagram in Fig. shows an overview of the system used in both UL and DL, where the coding, modulation were introduced

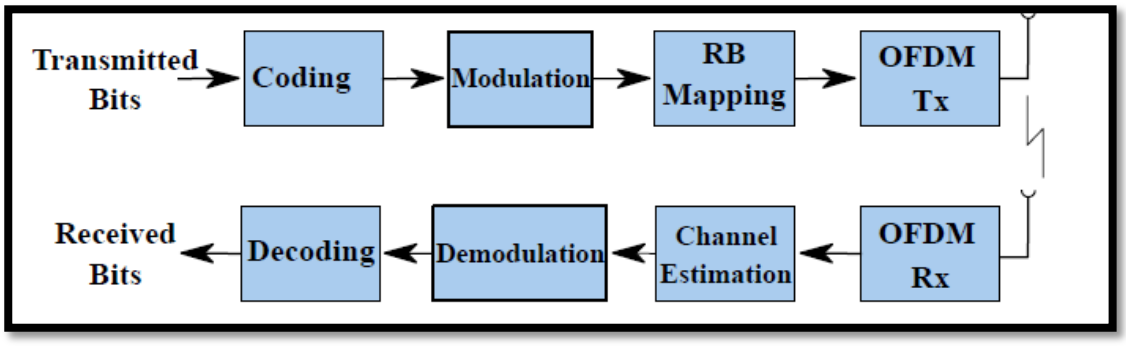


Figure 2-7: Block diagram of LTE Transmitter and Receiver

previously.

To transmit data to or from the UE, the transmitted bits should be processed first as follows:

1. The sequence of bits will be coded to increase its robustness using a coding sequence.
2. Modulation is then used to map the coded bits to complex symbols and transform the data from digital to analog signal.
3. The number of resource blocks is assigned by the scheduler, and the analog signal is mapped to the allocated resources. The main focus of this thesis is this part of the transmitter, called scheduling.
4. In LTE, an OFDM transmitter is used, where orthogonal carrier frequencies are used for transmission. One of the main advantages of OFDM is spectral efficiency, as more symbols can be sent on a given bandwidth when compared to other techniques. This main block combines several sub-blocks like Precoding and Antenna Mapping that deals with transmission using the antennas, and frequency conversion which up-converts the frequency of the signal to the frequencies used in LTE.

After transmitting the signal over the wireless channel, the receiver processes the received signal using the reversed path of the transmitter. The steps are as follows:

1. The OFDM receiver is used to down-convert the frequency of the signal and detect the information symbols of different carrier frequencies.

2. The wireless channel has a big effect on the signal that cannot be ignored for correct detection. Therefore, the variations in amplitude, phase and/or frequency caused by the channel are estimated and included in the demodulation.
3. Demodulation converts the received analog signal, including the channel estimation, and maps the detected symbols to the constellation diagram to extract the sequence of received coded bits.
4. The transmitter's coding sequence is used at the receiver to decode the received coded bits. Several error correction codes, as Turbo code, enable the receiver to find errors and correct them.

After transmitting a signal, the BS waits for a feedback from the UE included in the Hybrid Automatic Repeat Request (HARQ). The HARQ signal is sent in the UL, and it can either be:

1. Acknowledgment (ACK), when the signal is correctly decoded. This indicates that the packet is received, and the next packet can be sent.
2. Negative acknowledgment (NACK), when signal is incorrectly decoded due to a high number of errors. A retransmission of the same packet is then requested by the UE.

2.1.9 Wireless Channel:

The medium connecting the transmitter to the receiver in a wireless communication system is called wireless channel. The properties of this channel directly affect the propagating signal and should be taken into consideration at both the BS and UE to improve the

transmission. These properties include, but are not limited to, noise level and interference level.

2.2 Related works:

In wireless communication, resources like bandwidth and energy are scarce and extremely valuable, any system should serve as many users as possible while preserving high Quality of Service (QoS) for the best user experience. Accordingly, the Base Station (BS) has the responsibility to optimally schedule its resources to the users based on the available

information. Consequently, the whole process of scheduling is truly demanding and requires high complex calculations from the overall system. Hence, the request of more sophisticated and effective methods is substantial in order to minimize the challenges of scheduling. This paper focuses on the Modulation and Coding Scheme (MCS) selection in a Time Division Duplex (TDD) based mobile network. The main objective is the simplification and optimization of the downlink process at the base station by predicting the MCS index for a single User Equipment (UE), using Machine Learning (ML). The developed machine

learning algorithms is in accordance with the LTE-Advanced Pro (release 12, 13,14) lookup tables and is based on similar parameters. For

a given frame, this thesis targets predicting the MCS index of future subframes. Thus, the resource allocation process for independent users is becoming quicker and easier for the BS. The results are based on laboratory measurements at Ericsson, where the collection of data logs for several stationary UEs,

Occurred on a network testing environment and their different cell characteristics investigated thoroughly. Concluding, the accuracy level which the ML classification algorithm achieved was approximately 50 percent. Therefore, the prediction accuracy

can be described as sufficient for the BS to decrease the computation complexity and energy consumption during the downlink process. The data logs that the project took into account cannot be generalized for real-time[1].

A set of ML techniques are explored for MCS prediction in orthogonal frequency-division multiplexing (OFDM) systems, representing the first such investigation in this field. Additionally, it introduces a specialized Deep Neural Network (DNN) architecture with two hidden layers for MCS prediction, guided by performance metrics such as accuracy, precision, recall, and F1-score. The examined ML methods include Artificial Neural Networks (ANN), Support Vector Machine (SVM), Random Forest (RF), and Bagging with k-NN (B-kNN). These methods undergo thorough training and evaluation using a dataset generated from simulations of non-standalone 5G networks. The study incorporates physical layer measurements and employs a ray-tracing path loss prediction model for comprehensive environmental characterization. Also, advanced data mining techniques pre-process raw data, addressing model underfitting and overfitting challenges. Finally,

performance evaluation results reveal that the ANN with two hidden layers achieves the highest accuracy at 98.71%, while RF and B-kNN methods attain the lowest accuracy, below 88.65%. SVM and ANN models, with one and four hidden layers, respectively, demonstrate comparable MCS prediction accuracy, ranging from 97.02% to 97.30%[2].

Machine learning (ML) techniques to adaptive modulation and coding (AMC)-aided wireless systems to allow them to adapt to the variations of channels. It introduces and analyzes two types of ML-assisted AMC schemes in the context of the multiple-input and multiple-output and orthogonal frequency division multiplexing scenarios, where the conflicting demands for high date rate and high reliability are met by adjusting the modulation order and coding rate. Specifically, the chapter considers a supervised learning (SL) approach and a reinforcement learning approach. It first provides an overview of the ML-assisted AMC, and of the AMC schemes specified in the IEEE 802.11n, and gives details of the modulation types and coding rates used in the standards. Then, it provides information on SL-assisted AMC, where both the k-nearest neighbor and support vector machine approaches are considered. Finally, the chapter considers corresponding simulation results and analyses[3].

A mechanism is proposed to allow a cognitive user, also called Secondary User (SU), to access the frequency band of a Primary User (PU) operating based on an Adaptive Coding and Modulation (ACM) protocol. The Spectrum Sensing (SS) technique used considers Higher Order Statistical (HOS) features of the signal and log-likelihood ratios

(LLRs) of the code syndromes in order to constantly monitor the modulation and coding scheme (MODCOD) of the PU respectively. Once the Modulation and Coding Classification (MCC) is completed, a Power Control (PC) scheme is enabled. The SU can attempt to access the frequency band of the PU and increase its transmitting power until it causes a change of the PU's transmission scheme due to interference. When the SU detects the change of the PU's MODCOD, then it reduces its transmitting power to a lower level so as to regulate the induced interference. The proposed blind Adaptive Power Control (APC) algorithm converges without any interference channel information to the aforementioned interference limit and guarantees the preservation of the PU link throughput[4].

The modulation and coding schemes(MCS) given in the 5G standard are simulated under Gaussian channels, and obtains the SNR-BLER curves of different MCS. Then, the scheme of SNR mapping to CQI is improved, and a new mapping relationship between SNR and CQI is obtained by performing region fitting. Finally, for the adjustment factor optimization process, a new adjustment factor is introduced and the calculation formula for the optimization factor is modified. The BLER is taken as the logarithm and then used Logarithmic results to find the mean square error, thereby reducing mapping errors Compared with the traditional scheme, the proposed scheme has higher throughput, so as to better promote IoT technology[5].

Some background information about AMC is presented. First, AMC is described under flat fading channels; secondly, the impact of frequency selectivity and multiple-input multiple-output (MIMO)

techniques are addressed, including how block errors can be modeled to enable the prediction of the most appropriate MCS.

In 3GPP 5G NR as well as in previous LTE technology, AMC is a key technology. A summary of how 5G NR allows link adaptation, including how channel sounding is carried out, is briefly provided. In addition, some insights are given on how current technology copes with errors at the channel state estimation. A widely used algorithm termed outer loop link adaptation (OLLA) is described and its relation to a suboptimal AMC procedure unveiled. The description of some open issues and the suggestion of using machine learning as a new approach to AMC finishes the article[6].

A machine learning-based adaptive modulation (AM) model is proposed for the orthogonal frequency division multiplexing (OFDM) multiple-input multiple-output (MIMO) transmission system. Since 5G new radio (NR) system can be used in a large variety of Internet of Things fields than the conventional systems, the AM scheme that adjusts data rate and reliability according to channel condition can be effectively utilized. The conventional AM schemes are implemented by defining modulation schemes to be used according to each channel condition as table in advance. However, since the rule-based AM cannot analyze the communication performance according to the correlation of channels between antennas and the number of transmission modes is exponentially increased according to the number of available modulation schemes and antennas, it is not suitable for 5G NR system. The learning of the proposed AM model is based on the generated training signal by the extracted features from the received signal and assigned label through the performance analysis for signal detection. They focus on the

application of deep neural network for AM and cover the precedence method of principal component analysis to improve the performance of the model. The simulations on the classification of optimal transmission mode for the MIMO-OFDM signal demonstrate that the proposed model supports the adaptability according to the condition of complex MIMO channel[7].

In 5G enhanced mobile broadband (eMBB), applications that require high spectral efficiency and transmission speeds employ adaptive modulation and coding (AMC) technology. Such technology enables various levels of modulation and coding. AMC technologies use channel state information to select the optimal modulation or coding scheme as well as transmission parameters to improve the transmission speed, transmission quality, and spectrum efficiency of links. Because of equipment limitations at the receiving end, channel estimation algorithms cannot be used to acquire ideal solutions, and thus estimation errors are inevitable. Consequently, parameter information returned from the receiving end may be inaccurate and the radio link is affected by interference, which may impede the reliability of data transmission and reduce spectrum efficiency. This study proposed an AMC with a backpropagation artificial neural network (BP-ANN) scheme (AMC-BP-ANN). This study addressed problems regarding reduced efficiency in conventional AMC technology caused by channel, interference and noise estimation errors. In particular, this study used channel estimation, interference and noise estimation errors as features for machine learning, which allowed eMBB systems to accurately estimate the signal-to-interference plus noise ration and ensure that the estimated value approximated the value was determined through ideal channel

estimation. The simulation results indicated that the AMC-BP-ANN scheme could achieve ideal performance and a superior bit error rate compared with the AMC technologies using lookup table and genetic algorithm, and furthermore, a lower bit error rate could improve the transmission reliability and spectral efficiency in 5G eMBB[8].

Modifications are detected in 5G cooperative multiple-input multiple-output (MIMO) MIMO systems in the presence of spatially correlated channels and incomplete channel state information (CSI). At the destination node, we extract the higher-order statistics of the received signals as the discriminating features. After applying the principal component analysis technique, we carry out a comparative study between the random committee and the AdaBoost machine learning techniques (MLTs) at low signal-to-noise ratio. The efficiency metrics, including the true positive rate, false positive rate, precision, recall, F-Measure, and the time taken to build the model, are used for the performance comparison. The simulation results show that the use of the random committee MLT, compared to the AdaBoost MLT, provides gain in terms of both the modulation detection and complexity[9].

Automatic modulation recognition (AMR) plays a fundamental role in most intelligent communication systems, especially with the emergence of software-defined radio (SDR). AMR is an indispensable task while performing spectrum sensing in cognitive radio (CR). Thanks to the great advances in deep learning (DL) applications, new and powerful tools have been provided that can address the problems in this field. Therefore, the integration of deep learning models into AMR has received the attention of many researchers today. A comprehensive

review of the latest machine learning (ML)-based AMR methods for single-input-single-output (SISO) and multiple-input-multiple-output (MIMO) systems is presented. Furthermore, the architecture of each model is specified along with a detailed comparison in terms of specifications and performance. Finally, an outline of open problems, challenges and potential research directions is provided along with discussion and conclusion[10].

Chapter Three

Methodology

Chapter Three

Methodology

3.1 Introduction:

Nowadays, mobile communication networks are evolving in a tremendous way, constantly increasing their supported bit rates but also the complexity of the network in each upgraded version. A suitable example is the recent Massive Multiple-Input-Multiple-Output (MIMO) technology, which is boosting capacity and throughput in significant rates, while raising the BS processing complexity. Consequently, the need for more intelligent processing in the base station is essential and necessary. An efficient way to approach this problem would be the addition of the capability to predict, in a precise way, the optimal throughput at the Evolved Node B (eNodeB). This tactic would make the scheduling for a single or group of user equipment much more efficient in time and energy perspective.

The scheduler, located at the BS, is responsible for resource allocation in the downlink. Based on the received information from each UE in the uplink, the Channel Quality Indicator (CQI) is used to determine the user's downlink Modulation Coding Scheme (MCS) using a lookup table. With the continuous advance in technology, both at the user and base station sides, the modulation schemes are increasing and the MCS table is growing, reaching 31 distinctive settings in Long Term Evolution (LTE), compared to 15 in Wideband Code Division Multiple Access (WCDMA). Furthermore, with the increasing number of UEs,

especially in 5G where the estimated number is expected to be up to 1 billion subscriptions reaching 2023, a base station will have to serve multiple mobile terminals simultaneously and accurately.

The current scheduler, using the MCS lookup table, is relatively accurate but it only gives the MCS for the next subframe, given information at each frame and subframe. Therefore, to decrease the load on the scheduler, Machine Learning (ML) can potentially improve the performance of the scheduler by estimating and predicting the future MCS while taking into consideration the same user's information. Hence, ML algorithms, applied at the eNodeB, can achieve the prediction capability which we are looking for, while focusing on decreasing the complexity level.

3.2 Model Design:

The current scheduler, implemented in LTE/NR 3GPP, selects the MCS per subframe per user using the CQI provided in the uplink by the UE, However, the current scheduling process has the following limitations:

- MCS selection is limited for the next subframe for a given user. Entering to 5G NR, the processing complexity will increase, and the resources can become very limited. Hence, it is always a significant advantage for the resource planning of the independent UEs to have the expected MCS which will be used for the future subframes.
- The MCS selection becomes very complicated in the MU-MIMO case as the MCS for the user cannot only depend on the CQI, but also depends on the other UEs in the MU-MIMO group (group of users scheduled at the same time-frequency resource). The spatial

relation between all the users in the group needs to be analyzed and this takes lots of critical processing resources, resulting in a bottleneck for the NR.To take care of the aforementioned limitations, we propose the usage of ML at the BS aiming to predict future MCS. In summary, we propose the following:

- MCS prediction for the future subframes/slots using ML.
- Extend the MCS prediction for the candidate users of MU-MIMO. To diminish the process at the scheduler, one can provide it with information for future subframes using ML. ML can provide a good accuracy given the constant flow of data from the UE to the scheduler.

Therefore, given the decision is an integer between 0 to 31, for the LTE advanced Pro , classification algorithm is to be used instead of regression for the following reasons:

- Output is an integer with predefined number of classes and neighbors.
- High complexity of regression as several parameters are taken.

The overall system model is depicted in the following block diagram of Fig. , scheduling is done at eNodeB using the UL information of the user. To increase the efficiency, ML will provide information as well to the scheduler based on a training database,. Thus, the scheme includes three main parts: UE, Channel and Base Station.

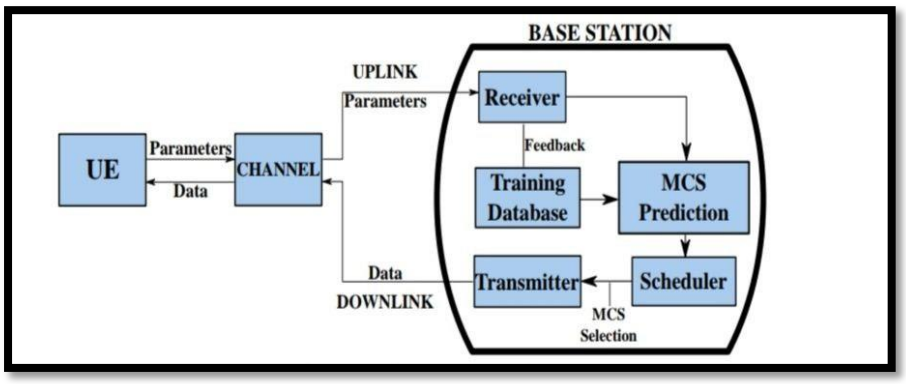


Figure 3-1: Block Diagram of The Suggested System Model

3.2.1 Parameters:

To make the system practical, a set of the parameters used for current MCS selection, are also used in the ML prediction algorithm. It includes SINR, HARQ, current MCS (k) and data size. Moreover, to improve the accuracy of the ML algorithm, the training database includes new features:

3.2.1.1 Current MCS Accuracy:

A flag indicating the accuracy of the MCS selected. In case of a retransmission at the next subframe, the MCS was not optimal and the flag will be indicated as zero. On the other hand, if the transmission was successful, then the flag will be marked as one.

3.2.1.2 UE Scenarios :

The training dataset includes data collected from the laboratory logs, where several UEs, in different channel conditions were tested. To keep track of the different behaviors, each UE was assigned a UserID, which will be used in the ML prediction algorithm. The dataset includes

UEs in alternative scenarios, and later a prediction is made for the current UE. Furthermore, it is useful to investigate the closest historical UE, so the algorithm can track the stored behavior and assign it to the current one.

3.3 Implementation:

Based on the suggested scheme, the prediction method used in this thesis can be implemented, with some add-ons, at the base station to increase the efficiency of the scheduler. These add-ons are the training database and the ML algorithm.

In this section, a step-by-step overview of the work done will be presented in detail, from the selected programming language to the implementation of the ML algorithm.

3.3.1 Programming Language:

Nowadays, several platforms and programming languages include AI and ML in their libraries like Microsoft, IBM, MATLAB, etc. In this thesis, Python was chosen for scripting and programming due to the free and open sourced extensive libraries. Additionally, Python is a dynamic and adaptive programming language, and it can be used for several purposes as it is easy to implement. One of them is scripting to gather useful data from large files Finally, since 2007, Python is continuously updating its ML libraries such as Tensor Flow, SciKit Learn, NumPy, etc. They were used to write the ML algorithm for MCS prediction.

3.3.2 Trace Logs:

Trace logs were collected using Ericsson’s system and equipment. Using an eNodeB and several stationary UEs in the laboratory, various

scenarios were created by adding intercell interference and by varying the BS's transmission power. Data from several UE scenarios with very different channel conditions were included in the trace logs, and even though the UEs weren't moving, the MCS is varying for different scenarios given the interference level from the neighboring cells. The trace logs collection was done as shown in the block diagram in Fig3-2.

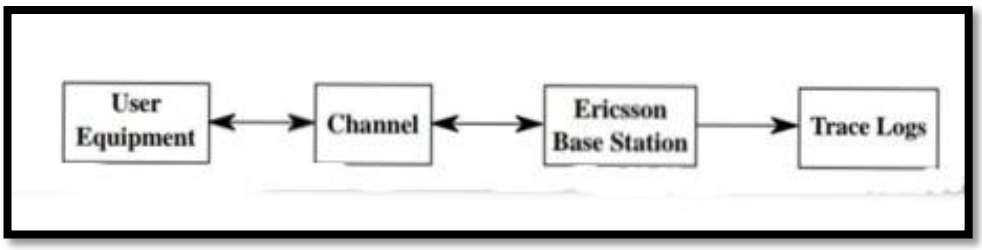


Figure 3-2: Model Used for Trace Logs Collection.

3.3.3 Data Collection:

After running the simulation in section 3.3, the trace logs need filtering to gather useful data for the ML algorithm. The parameters, listed in section 3.2.1, form the columns in the dataset, and the remaining data, after filtering, form the entries. Table 5 represents the useful parameters, including their types and description, collected from the trace logs.

Table3-1: Parameters used in the training dataset

Parameter	Variable Type	Description
userID	integer	Indicates user's scenario
SINR	double	Included in the CQI report
MCS	integer	MCS of current subframe
AccMCS	flag (Bool)	Accuracy of the selected MCS
next MCS	integer	MCS for the next subframe
NDF	flag (Bool)	Based on HARQ feedback
Bytes	integer	Data size to be transmitted

The SINR is updated every frame while the rest of parameters is updated every subframe. To predict the future MCS, a sample must be included in the training dataset. This sample is "next MCS" and it was collected by processing the whole frame of each UE and appending the MCSs in one column.

3.4 ML Algorithm:

Following the steps in the previous sections, the final dataset was used to train and test the algorithm. the training sequence and test data

are both part of the collected data and they contribute together to the ML prediction, and they are divided based on the test size.

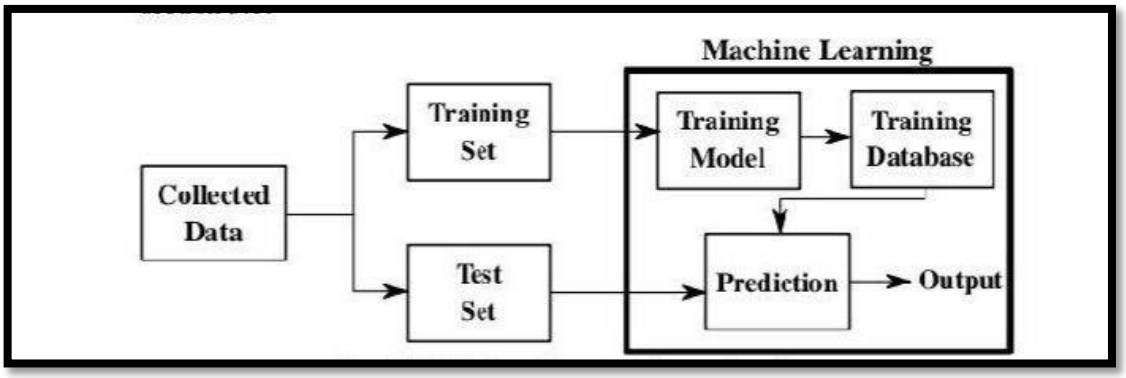


Figure 3-3: ML Algorithm Training and Testing

In this thesis, the prediction was made using the neural network classification algorithm. The main reason behind choosing classification is the integer output using multiple variables, and the neural network is in accord with the prediction steps, as shown in fig3-4.

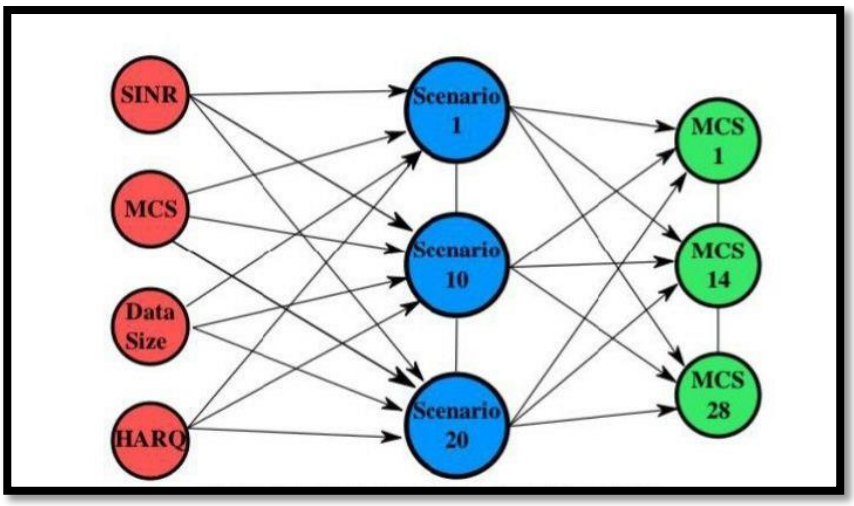


Figure 3-4: Neural Network Algorithm for MCS Prediction

In details, the future MCS prediction, output, is as follows:

- Predict the scenario (hidden node) based on the input data
- Predict the MCS based on output of the hidden node

By choosing the closest UE scenario from the training dataset, using the parameters mentioned in table 3 as inputs, the future MCS can be predicted as a smaller yet more accurate and relative dataset will be used.

The flowchart in Fig3-5. shows the detailed steps taken to make an MCS

prediction based on the input data from the UE.

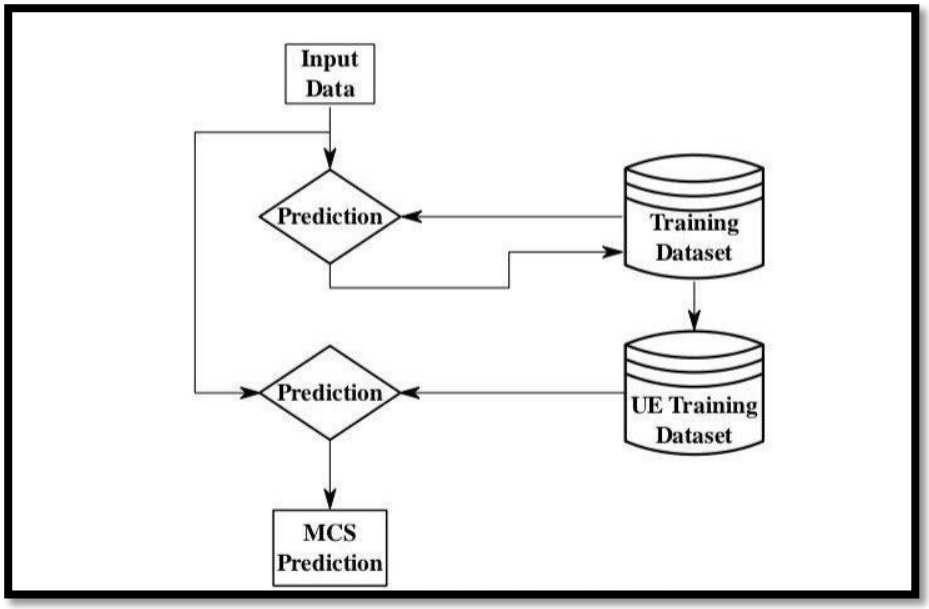


Figure 3-5: The MCS Prediction

The UE training dataset is a sub-dataset of the main training dataset, and it is selected using the predicted UE scenario (hidden node in the neural network). The second prediction follows using the same input data and the selected sub-dataset, so no additional data is added to the overall prediction.

CHAPTER FOUR

RESULTS

CHAPTER FOUR

RESULTS

4.1 Algorithm Construction

4.1.1 Artificial Neural Network Model:

The ANN model is a multi-class classification model, which means it will be used to categorize audio files into one of the classes. The model is trained using backpropagation to adjust the weights of the connections between neurons and minimize the error between the predicted class and the actual class.

4.1.2 Artificial Neural Network Architecture (ANN):

The model is defined using the Sequential class from Keras, which allows you to define a model as a linear stack of layers. The model has seven dense layers; there is input layer, five fully connected layer, output layer, each one with a different number of units.

The layers are described below:

Dense Layer:

It is deeply connected with its preceding layer which means the neurons of the layer are connected to every neuron of its preceding layer.

Input Layer:

A selected type of sound is the input to the model. The input layer has 128 neurons, each with the input shape of (128).

Fully Connected Layers:

Finally, the feature maps produced by the convolutional and pooling layers are fed into a series of fully connected layers, which perform the final classification of the sound.

Output layer:

The output layer produces the final classification results, which indicate the type of sound that was input into the model. It has five neurons, each with a softmax activation function, which is commonly used in multiclass classification problems.

The activation function for each layer is the rectified linear unit (ReLU).

Relu:

The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.

Softmax:

Softmax is a mathematical function that converts a vector of numbers into a vector of probabilities, where the probabilities of each value are proportional to the relative scale of each value in the vector.

The most common use of the softmax function in applied machine learning is in its use as an activation function in a neural network model. Specifically, the network is configured to output N values, one for each class in the classification task, and the softmax function is used to normalize the outputs, converting them from weighted sum values into probabilities that sum to one. Each value in the output of the softmax function is interpreted as the probability of membership for each class.

4.2 Importing the Libraries & Downloading the Data:

These libraries are Python libraries used in data analysis and machine learning:

pandas (import pandas as pd):

Used for data analysis and organization. It provides a powerful and user-friendly data interface.

scikit-learn:

It is a machine learning library in Python. It contains various tools and algorithms that aid in data analysis and building machine learning models.

Numpy:

A library used for numerical calculations. Provides powerful tools for working with arrays and various mathematical operations. It is essential in many data processing applications.

Matplotlib:

A library used to create charts and graphs. It enables you to visually display data in an easy way such as line charts, column charts, pie charts, etc.

Tensor Flow:

A library for Deep Learning and Machine Learning. Used to build and train neural networks. Supports working with big data effectively.

Seaborn:

Is a powerful library built on matplotlib, used to draw graphs and analyze data in a more aesthetic and easy way. Seaborn provides high-level graphics that facilitate visual data analysis through ready-to-use templates.

Figure 4-1: Python libraries Used in Data Analysis and Machine learning

accuracy_score:

Is used to measure the model's accuracy when predicting test data.

4.2.1 Meet the data:

Use the head() function to display some of the first rows and examine the structure of the data.

```
[62]: par.head()
```

[62]:

	userID	SINR	MCS	AccMCS	NDF	Bytes	next MCS
0	1	4.981605	18	24	42	1099	19
1	2	28.028572	7	43	7	1291	5
2	3	19.279758	6	54	41	539	7
3	4	13.946339	15	82	11	1251	17
4	5	-3.759254	18	83	41	831	20

Figure 4-2: First Five Rows in the Data

Afterward, we used several commands to explore the data, and let's review the explanation of their functions, some of which are:

par.shape in Python is used to obtain the dimensions of a Data Frame. The output is a tuple that includes the number of rows and the number of columns in the Data Frame. **par.info** in Python is used to obtain the data types for each column in a Data Frame.

```
[61]: par.shape
[61]: (5000, 7)
```

Figure 4-3: Dimensions of a Data Frame

```
[65]: par.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   userID    5000 non-null   int32
 1   SINR      5000 non-null   float64
 2   MCS       5000 non-null   int32
 3   AccMCS    5000 non-null   int32
 4   NDF       5000 non-null   int32
 5   Bytes     5000 non-null   int32
 6   next MCS  5000 non-null   int32
dtypes: float64(1), int32(6)
memory usage: 156.4 KB
```

Figure 4-4: Types of Data for Each Column in a Data Frame

Since the utilized database is organized, we should use the command **par_randomized** to ensure that we obtain a good model.

df_randomized, randomization process for the rows of the original Data Frame.

4.2.2 Identifying features and target:

```
[77]: # تحديد الخصائص المستقلة والهدف
      X = par[['SINR', 'MCS', 'AccMCS', 'NDF', 'Bytes']]  # تعديل الاسم هنا
      y = par['next MCS']
```

```
[78]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Figure 4-5: Features and Target

StandardScaler:

Part of the Scikit-learn library and is used to standardize data. It transforms data so that its mean is zero and its standard deviation is one, which helps improve the performance of machine learning algorithms.

```
*[79]: scaler = StandardScaler()
       X_train = scaler.fit_transform(X_train)
       X_test = scaler.transform(X_test)
```

Figure 4-6: StandardScaler Library

4.2.3 Splitting the data and Model training:

4.2.3.1 Model Compile:

The model is compiled using the categorical cross entropy loss function, the Adam optimizer and accuracy as a metric.

4.2.3.2 Optimizer:

The Adam optimization algorithm is an extension to stochastic gradient descent that has recently seen broader adoption for deep learning applications in computer vision and natural language processing, as it is efficient and can adapt the learning rate for each parameter.

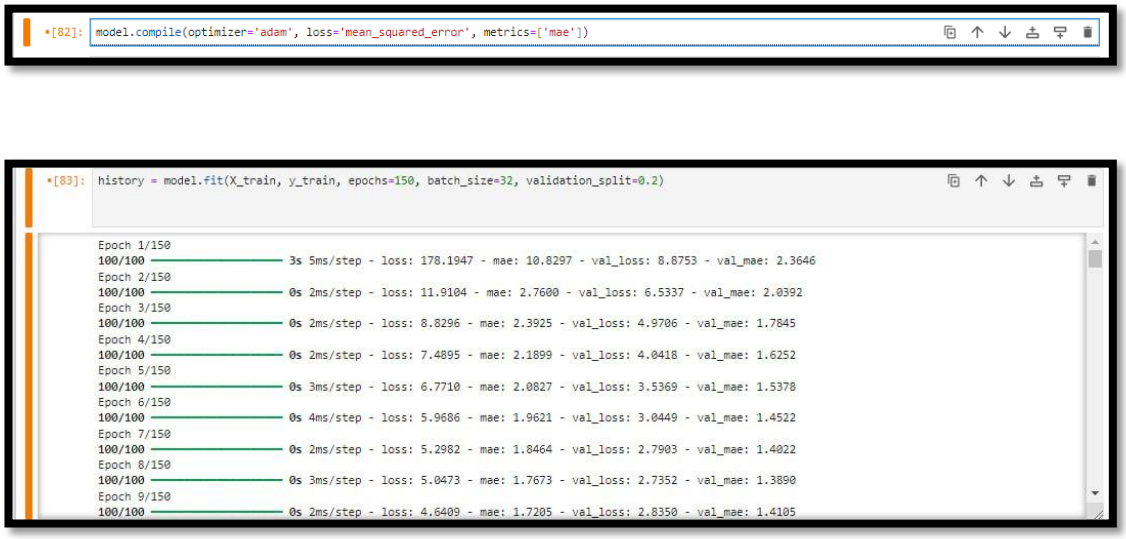
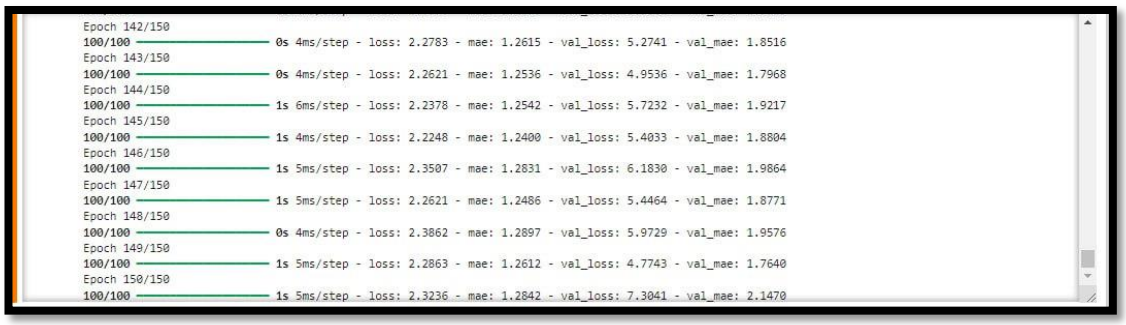


Figure 4-7: Model Compile and Optimizer



And here are Some plots to illustrate the distribution of the data:

```
[66]: par.hist()

[66]: array([[<Axes: title={'center': 'userID'}>,
              <Axes: title={'center': 'SINR'}>,
              <Axes: title={'center': 'MCS'}>],
             [<Axes: title={'center': 'AccMCS'}>,
              <Axes: title={'center': 'NDF'}>,
              <Axes: title={'center': 'Bytes'}>],
             [<Axes: title={'center': 'next MCS'}>, <Axes: >, <Axes: >]],
            dtype=object)
```

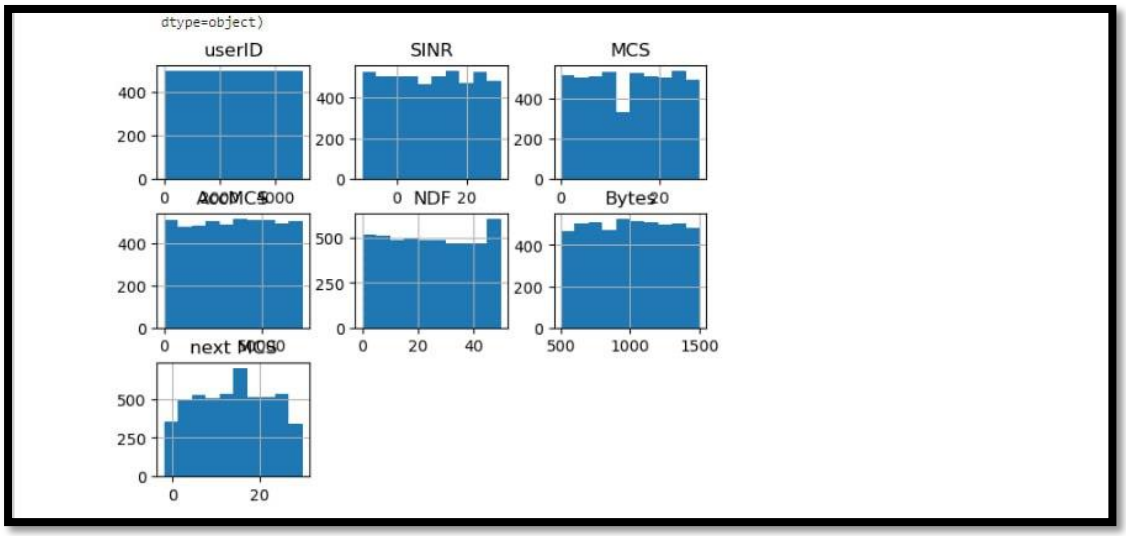


Figure 4-8: Histogram of the features

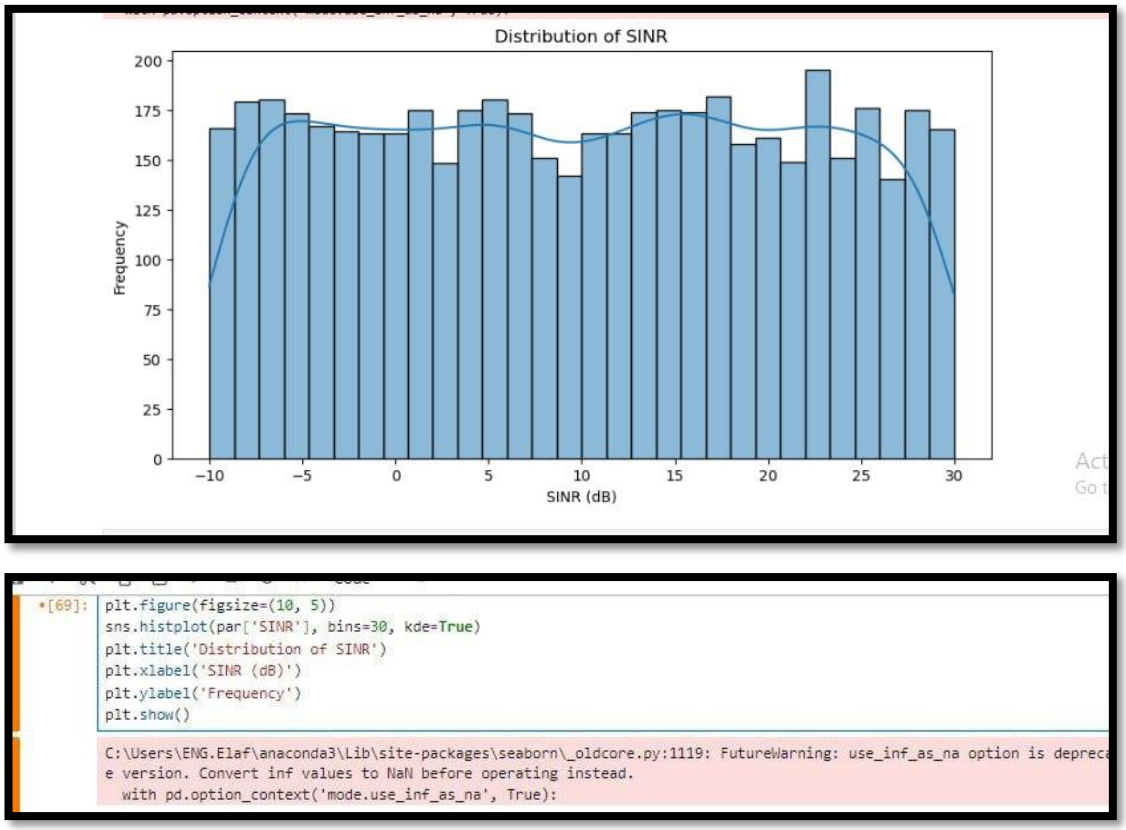


Figure 4-9: Relationship Between Frequency and SINR

```
*[70]: plt.figure(figsize=(10, 5))
       sns.histplot(par['MCS'], bins=30, kde=True)
       plt.title('Distribution of MCS')
       plt.xlabel('MCS')
       plt.ylabel('Frequency')
       plt.show()

C:\Users\ENG.Elaf\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprec
e version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

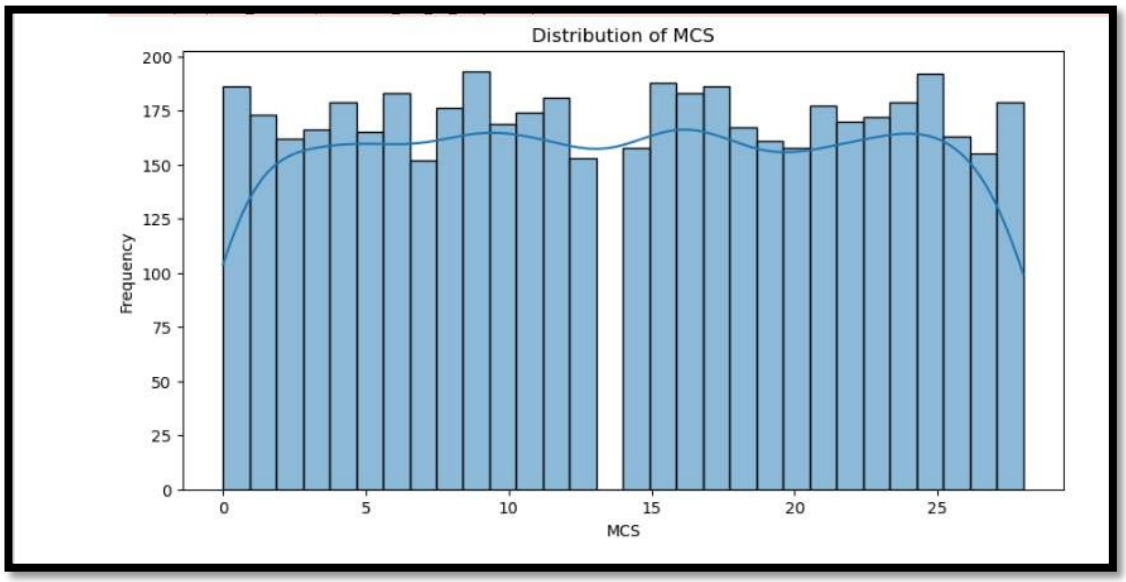


Figure 4-10: Relationship Between Frequency and MCS

```
*[71]: plt.figure(figsize=(10, 5))
       sns.scatterplot(x='MCS', y='next MCS', data=par)
       plt.title('MCS vs Next MCS')
       plt.xlabel('MCS')
       plt.ylabel('Next MCS')
       plt.show()
```

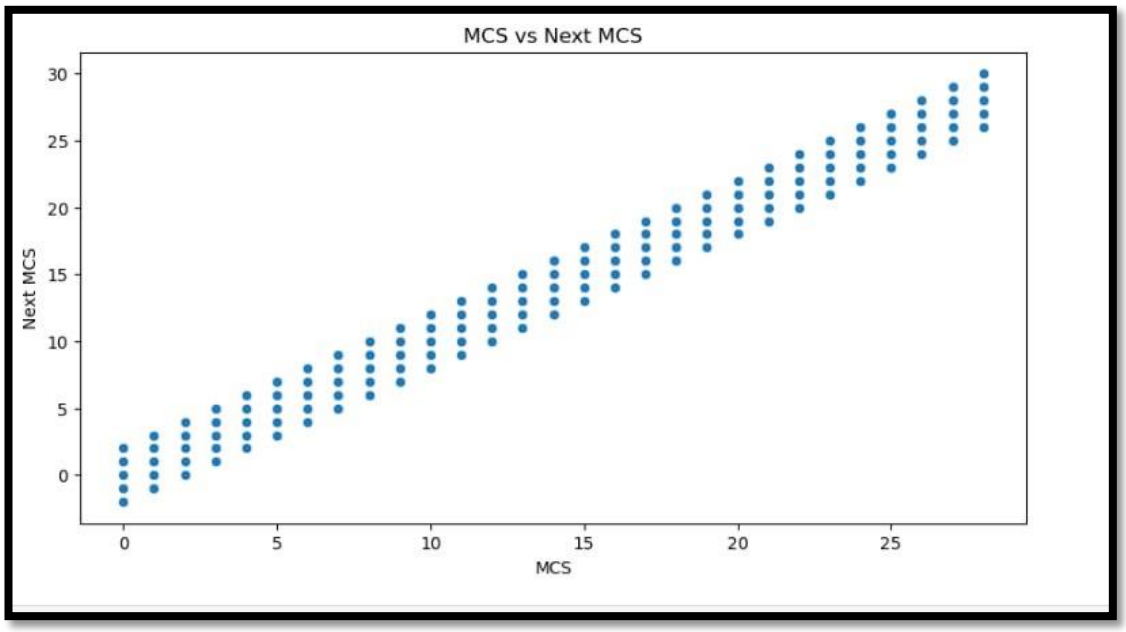


Figure 4-11: Relationship Between Next MCS and MCS

```
[73]: plt.figure(figsize=(10, 6))
      sns.scatterplot(x=par['NDF'], y=par['SINR'], hue=par['next MCS'])
      plt.title('Relationship between NDF and SINR')
      plt.show()
```

Relationship betwee

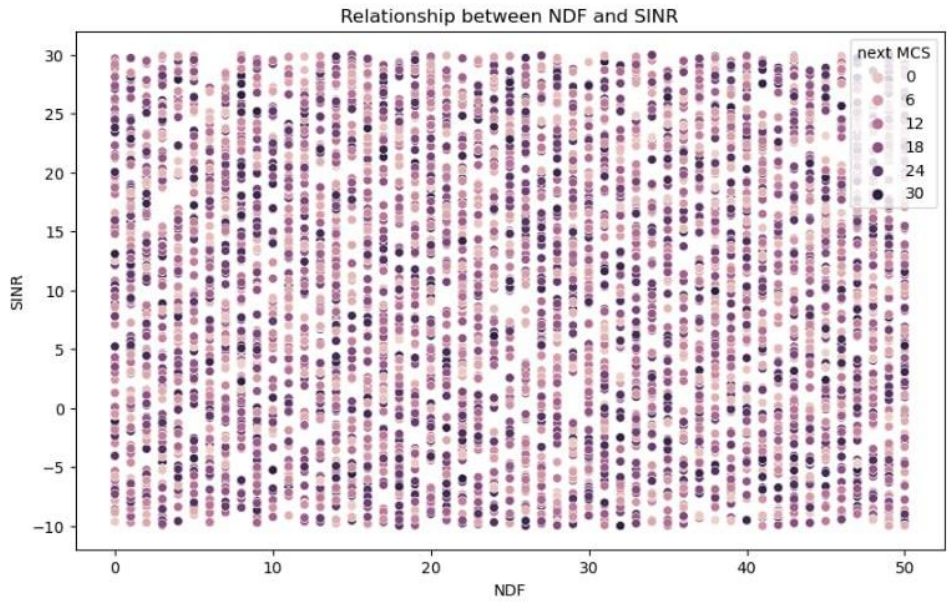


Figure 4-12: Relationship Between Two Variables NDF and SINR

4.2.4 Performance evaluation:

The model's results indicate excellent accuracy in classifying the data, adding value to a better understanding of network performance and future enhancements. Based on the analysis and exploration provided by the model, further research and analysis are encouraged to uncover more opportunities and improvements in the field of network engineering.

Continuing this work, we now have a custom-built Artifical Nural Network (ANN) model designed for classifying telecommunication data.

This model was developed using machine learning techniques to analyze signals from the Third Generation (3G), Fourth Generation (4G), Fifth Generation (5G), and LTE.

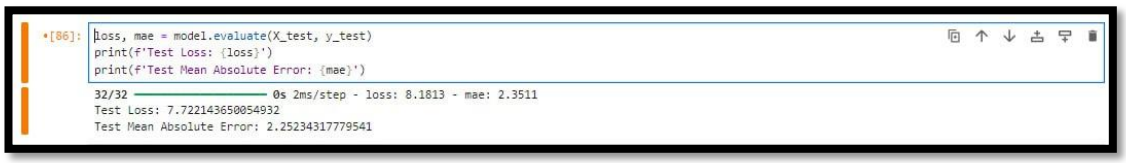


Figure 4-13: Evaluation Process

4.2.5 Prediction:

The prediction function in Python is not a foolproof function, but in the context of matter learning and artificial intelligence, we often see how a trained model is used to predict outcomes associated with new data. This function is usually part of a library such as scikit-learn, Tensorflow, or other machine learning libraries.

In general, the prediction function depends on the type of mathematical model or algorithm used.

Precision calculated by 4.1 formula

$$\text{Precision} = TP/(TP + FP) \quad 4.1$$

Figure 4-14: Prediction Process

4.2.6 Accuracy:

Accuracy of model calculate by formula (4.2).

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad 4.2$$

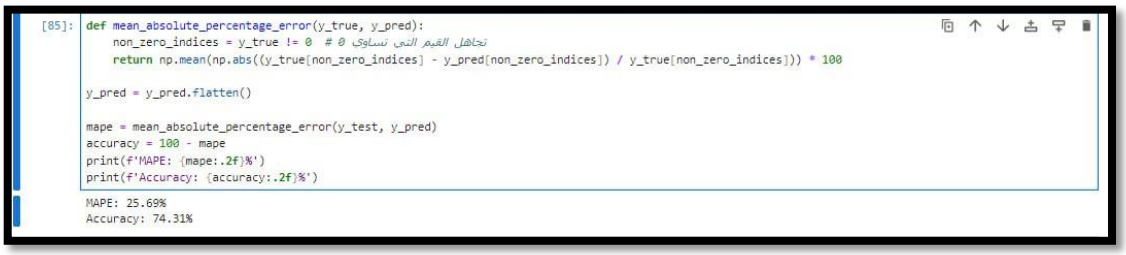


Figure 4-15: Accuracy of the Model

4.2.7 Validation:

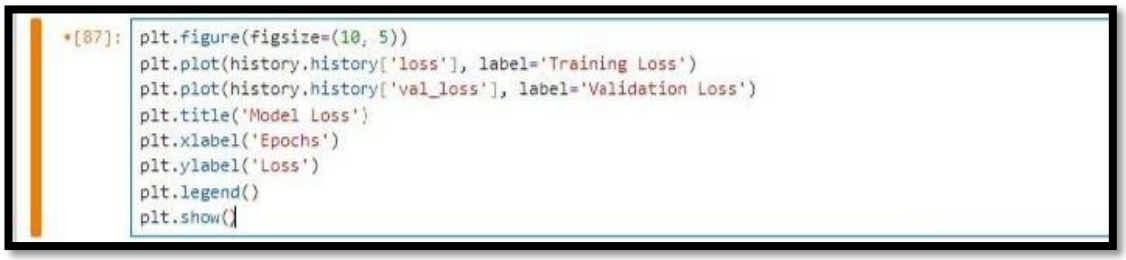


Figure 4-16: Validation Process

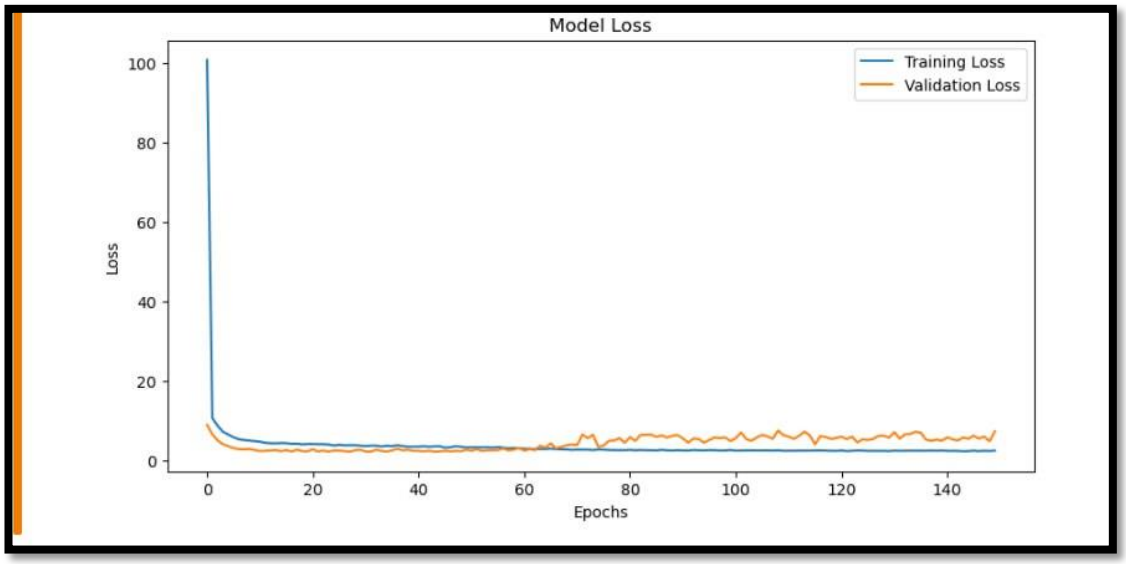


Figure 4-17: Loss per Epochs

Epochs:

In terms of artificial neural networks, an epoch refers to one cycle through the full training dataset. In this project the model trained for 20 epochs.

Chapter Five

Conclusion and Recommendation

Chapter Five

Conclusion and Recommendation

5.1 Conclusion:

With the arrival of 5G and its embedded technologies, the complexity and power consumption are increasing in an “alarming” manner. To solve this problem, we propose a multi-class machine learning model based on neural networks that will predict the MCS index of the expected subframe of an autonomous user device. By exploiting classification learning, we can reduce the above-mentioned “alarming” parameters and enhance the performance of link adaptation. However, some challenges we encountered have limited the results and potential outcomes of our project. The data collected are limited. The reasons for this were, firstly, the limited time of our research compared to the large capabilities of investigating the machine learning topic, and secondly, the lack of real-time large data records that would expand the scope of our thesis exploration in realistic channel conditions. Moreover, the prediction accuracy of the MCS index selection algorithm achieved a peak rate of seventy-four percent, which gives promising rates for further exploration. Hence, this prediction aims to optimize the utilization of the base station (BS), as it is always an important advantage for resource allocation if the management control systems (MCS) of users are already known for the future subframes.

5.2 Recommendation:

To develop the ML-based Adaptive Modulation and Coding (AMC) project in 5G networks, several future improvements can be considered:

1. Improvement of Machine Learning Models:

Use more complex models such as deep neural networks to enhance prediction accuracy, Integrate reinforcement learning methods for better self-adaptation.

2. Larger Data Analysis:

Collect data from a diverse range of networks and users to improve training models and Utilize big data analytics techniques to enhance understanding of behaviors.

3. Network Collaboration:

Develop systems that allow information exchange between different networks to improve overall performance.

4. Personalized Experience:

Create customized solutions for each user based on their specific usage patterns to increase efficiency.

5. Security Enhancements:

Integrate security strategies to protect data during transmission.

6. Implementation of new Technologies:

Explore the use of cloud or edge computing to improve performance.

7. Mobility Prediction:

Use user mobility prediction models to optimize resource allocation.

8. Testing in Different Environments:

Conduct experiments in various environmental conditions (urban vs. rural) to obtain diverse data.

9. Support for 6G:

Begin planning for future support of next-generation technologies (6G) and anticipate their requirements.

By applying these enhancements, the project's efficiency can be improved, and its performance can be increased in advanced network environments.

Reference:

1. Chamat, M. and B.D. Kodra, *Machine Learning Based Modulation and Coding Scheme Selection.* 2019.
2. Tsipi, L., et al., *Machine learning-based methods for MCS prediction in 5G networks.* Telecommunication Systems, 2024: p. 1-24.
3. Zhang, L. and Z. Wu, *Machine Learning–Based Adaptive Modulation and Coding Design.* Machine Learning for Future Wireless Communications, 2020: p. 157-180.
4. Tsakmalis, A., S. Chatzinotas, and B. Ottersten. *Modulation and coding classification for adaptive power control in 5G cognitive communications*. in *2014 IEEE 15th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. 2014. IEEE.
5. Wang, Y., W. Liu, and L. Fang. *Adaptive modulation and coding technology in 5G system*. in *2020 International Wireless Communications and Mobile Computing (IWCMC)*. 2020. IEEE.
6. Aguayo-Torres, M.C., et al., *Adaptive Modulation and Coding.* Wiley 5G Ref: The Essential 5G Reference Online, 2019: p. 1-34.
7. Ha, C.-B., Y.-H. You, and H.-K. Song, *Machine learning model for adaptive modulation of multi-stream in MIMO-OFDM system.* IEEE Access, 2018. **7**: p. 5141-5152.
8. Chen, L.-S., et al., *AMC with a BP-ANN scheme for 5G enhanced mobile broadband.* IEEE Access, 2020.
9. Chikha, H.B., A. Almadhor, and W. Khalid, *Machine learning for 5G MIMO modulation detection.* Sensors, 2021. **21**(5): p. 1556.
10. Jdid, B., et al., *Machine learning based automatic modulation recognition for wireless communications: A comprehensive survey.* IEEE Access, 2021. **9**: p. 57851-57873.

APPENDIX

A.APPENDIX A

Python code

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from keras.models import Sequential
from keras.layers import Dense
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from keras.optimizers import Adam


num_samples = 5000
np.random.seed(42)


par = {
   'userID': np.arange(1, num_samples + 1),  # معرفات المستخدم من 1 إلى 5000
   'SINR': np.random.uniform(-10, 30, size=num_samples),  # SINR - بين 10 و 30 ديسيبل
   'MCS': np.random.randint(0, 29, size=num_samples),  # MCS من 0 إلى 28
```

```
    'AccMCS': np.cumsum(np.random.randint(0, 29, size=num_samples)),  # MCS المتراكم
    'NDF': np.random.randint(0, 51, size=num_samples),  # NDF بين 0 و 50
    'Bytes': np.random.randint(500, 1501, size=num_samples)  # بايت بين 500 و 1500
}

par['next MCS'] = par['MCS'] + np.random.randint(-2, 3, size=num_samples)  # MCS القادم مع بعض الضوضاء
par = pd.DataFrame(par)
print(par.head())
par.to_csv('mock_data_with_target.csv', index=False)

par.shape

par.head()

par.isnull().sum()

par.describe()

par.info()

par.hist()

plt.figure(figsize=(10, 8))
correlation_matrix = par.corr()
```

```
sns.heatmap(correlation_matrix, annot=True, fmt=".2f",
cmap='coolwarm', square=True)
plt.title('Correlation Matrix')
plt.show()

print(par.columns)

plt.figure(figsize=(10, 5))
sns.histplot(par['SINR'], bins=30, kde=True)
plt.title('Distribution of SINR')
plt.xlabel('SINR (dB)')
plt.ylabel('Frequency')
plt.show()

plt.figure(figsize=(10, 5))
sns.histplot(par['MCS'], bins=30, kde=True)
plt.title('Distribution of MCS')
plt.xlabel('MCS')
plt.ylabel('Frequency')
plt.show()

plt.figure(figsize=(10, 5))
sns.scatterplot(x='MCS', y='next MCS', data=par)
plt.title('MCS vs Next MCS')
plt.xlabel('MCS')
plt.ylabel('Next MCS')
plt.show()
```

```
plt.figure(figsize=(10, 6))
sns.histplot(par['NDF'], kde=True)
plt.title('Distribution of NDF')
plt.show()

plt.figure(figsize=(10, 6))
sns.scatterplot(x=par['NDF'], y=par['SINR'], hue=par['next MCS'])
plt.title('Relationship between NDF and SINR')
plt.show()

plt.figure(figsize=(10, 6))
sns.countplot(x='MCS', data=par)
plt.title('Distribution of MCS')
plt.show()

plt.figure(figsize=(8, 6))
sns.countplot(x='next MCS', data=par)
plt.title('next MCS Distribution')
plt.show()

sns.pairplot(par, hue='next MCS')
plt.show()

X = par[['SINR', 'MCS', 'AccMCS', 'NDF', 'Bytes']]  # تعديل الاسم هنا
y = par['next MCS']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

model = tf.keras.models.Sequential()

model.add(tf.keras.layers.Dense(128, activation='relu', input_shape=(X_train.shape[1],)))
model.add(tf.keras.layers.Dropout(0.2))  # Dropout لتجنب overfitting
model.add(tf.keras.layers.Dense(64, activation='relu'))
model.add(tf.keras.layers.Dropout(0.2))
model.add(tf.keras.layers.Dense(32, activation='relu'))
model.add(tf.keras.layers.Dense(1))  # طبقة الإخراج

model.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])

history = model.fit(X_train, y_train, epochs=150, batch_size=32, validation_split=0.2)

y_pred = model.predict(X_test)

def mean_absolute_percentage_error(y_true, y_pred):
    non_zero_indices = y_true != 0  # تجاهل القيم التي تساوي 0
    return np.mean(np.abs((y_true[non_zero_indices] - y_pred[non_zero_indices]) / y_true[non_zero_indices])) * 100

y_pred = y_pred.flatten()
```

```
mape = mean_absolute_percentage_error(y_test, y_pred)
accuracy = 100 - mape
print(f'MAPE: {mape:.2f}%')
print(f'Accuracy: {accuracy:.2f}%')

loss, mae = model.evaluate(X_test, y_test)
print(f'Test Loss: {loss}')
print(f'Test Mean Absolute Error: {mae}')

plt.figure(figsize=(10, 5))
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Model Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()

print(history.history.keys())

plt.plot(history.history['mae'], label='Train MAE')
plt.plot(history.history['val_mae'], label='Test MAE')

# إجراء التنبؤات
predictions = model.predict(X_test)
comparison = pd.DataFrame({'Actual': y_test, 'Predicted':
predictions.flatten()})
```

```
print(comparison.head())
```