People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Mohamed El-Bachir El-Ibrahimi University – Bordj Bou Arreridj

Faculty of Science and Technology

Department of Electronics

Memory

Presented for the award of the MASTER's degree

Field: Telecommunication

Specialty: Telecommunications Systems

By:

Zidoune Yousra Eldjamila

Barkati Lyna

Subject

Classification of advanced radio modulation techniques using deep learning networks

Supported on 12 / 06 /2025

Before the jury composed of:

Dr.MESSOUDENE Idris	Grade	**MCA**	President
Dr.BELAZOUG Massinissa	Grade	**MCB**	Examiner
Dr.TITOUNI Salem	Grade	**MCB**	Supervisor

Academic Year 2024/2025

Acknowledgment:

At the outset of this thesis, we would like to thank Allah, the Almighty, for granting us the opportunity to reach this stage of knowledge, and for bestowing upon us the courage and patience to complete this humble work.

We would like to express our sincere gratitude to our supervisor, Dr. S. Titouni, for his invaluable guidance, continuous support, and unwavering patience throughout this project. His insightful advice and constant encouragement greatly motivated us and played a key role in the successful completion of this work. We hope this thesis stands as a modest reflection of our deep appreciation for his dedication and exemplary leadership.

We are also indebted to the jury members for dedicating their time to assess our research. Their insightful remarks, constructive critiques, and valuable suggestions will surely enhance the quality of this work and guide us in our future academic and professional pursuits.

Special thanks are due to all the professors who have accompanied us along our academic path. Their passion for teaching, generosity in sharing knowledge, and unwavering support have profoundly enriched our education and played a crucial role in shaping both our personal and professional development.

Lastly, we express our heartfelt thanks to all individuals who, in various ways, contributed to the realization of this thesis.

Dedications

I dedicate this modest work to:

To my beloved parents, taken from me far too early, whom I hold forever in my heart.

Although you left me when I was very young, I have always felt the deep weight of your absence. At the same time, your love and memory have stayed with me, giving me strength. This unseen legacy has quietly guided me through life, helping me to keep going, to grow, and to become the person I am today.

This simple but sincere work is dedicated to your precious memory, a way to express my deep gratitude and everlasting love for you, who are the foundation of everything I have become.

To my beloved grandmother, who sacrificed much and raised me with care. Your support has played a key role in who I am today.

To my two brothers « Youcef» and «Akram»

To my nieces «Nour ELhayet» and «Lyna »

To my binomial «Lyna », who throughout this project has been like a sister to me. Even though our journey hasn't always been easy, your presence by my side has made this experience unique and unforgettable.

Yousra

Dedications

I dedicate this modest work to:

My beloved parents, whose unwavering love, patience, and sacrifices have laid the foundation for every success I have achieved. Their enduring belief in my potential has been a constant source of strength and motivation throughout my academic journey.

To my two sisters « Rayane» and «Hanine».

To my beloved aunt « Messaouda», who support and advised me throughout my career.

To my cousins « Amira», « Anfel», « Riham», « Miled » and « Chahd».

To my lovely cats « Milka » and « Bibiche ».

And last but not least my binomial and my best friend « Yousra », who was with me through this project, may Allah make all your wishes happened.

Lyna

Abstract:

This work focuses on the automatic classification of analog and digital modulation schemes in modern wireless communication systems using both deep learning and machine learning techniques. A thorough experimental study was conducted using the RadioModRec-1 dataset, which includes a variety of real-world modulation types under different signal conditions. Three models were implemented and compared: a Convolutional Neural Network (CNN), Random Forest, and eXtreme Gradient Boosting (XGBoost). The results demonstrate that XGBoost achieved the highest classification accuracy of 98.45%, followed closely by Random Forest with 97.32%, while the CNN reached an accuracy of 70.64%. These outcomes confirm the strong performance of ensemble learning methods in structured signal environments, while also highlighting the adaptability of DL models in handling raw input data. Overall, this study emphasizes the potential of Artificial Intelligence driven approaches to improve the efficiency, accuracy, and robustness of modulation recognition, contributing to the advancement of intelligent and reliable wireless communication systems.

Résumé:

Ce travail porte sur la classification automatique des schémas de modulation analogiques et numériques dans les systèmes de communication sans fil modernes, en s'appuyant à la fois sur des techniques d'apprentissage automatique (ML) et d'apprentissage profond (DL). Une étude expérimentale approfondie a été réalisée à l'aide du jeu de données RadioModRec-1, qui comprend une grande variété de types de modulations représentatives de conditions réelles de transmission. Trois modèles ont été implémentés et comparés : un réseau de neurones convolutifs (CNN), un Random Forest et un XGBoost. Les résultats montrent que XGBoost a atteint la meilleure précision de classification avec 98,45 %, suivi de près par Random Forest avec 97,32 %, tandis que le CNN a obtenu une précision de 70,64 %. Ces résultats confirment les performances solides des méthodes d'apprentissage par ensemble dans des environnements de signaux structurés, tout en soulignant la capacité d'adaptation des modèles d'apprentissage profond à traiter directement les données brutes. Dans l'ensemble, cette étude met en évidence le potentiel des approches basées sur l'intelligence artificielle pour améliorer l'efficacité, la précision et la robustesse de la reconnaissance de modulation, contribuant ainsi au développement de systèmes de communication sans fil intelligents et fiables.

الملخص

يهدف هذا العمل إلى تصنيف أنماط التضمين التناظرية والرقمية تلقائيًا في أنظمة الاتصالات اللاسلكية الحديثة، وذلك بالاعتماد على تقنيات التعلم الآلي والتعلم العميق .تم إجراء دراسة تجريبية شاملة باستخدام مجموعة البيانات RadioModRec-1، التي تتضمن مجموعة متنوعة من أنماط التضمين في ظروف إشارة واقعية مختلفة. تم تنفيذ ومقارنة ثلاثة نماذج :الشبكة العصبية الالتفافية(CNN) ، غابة عشوائية(Random Forest) ، و XGBoost.أظهرت النتائج أن نموذج XGBoostحقق أعلى دقة في التصنيف بنسبة 98.45%، يليه نموذج Random Forest بنسبة 97.32%، بينما بلغ أداء شبكة CNNنسبة 70.64 .%تؤكد هذه النتائج فعالية خوارزميات التعلم التجميعي في التعامل مع البيانات المنظمة، كما تبرز قدرة نماذج التعلم العميق على معالجة البيانات الخام بشكل مباشر .بشكل عام، تؤكد هذه الدراسة على إمكانيات الذكاء الاصطناعي في تحسين كفاءة ودقة وموثوقية تصنيف أنماط التضمين، مما يساهم في تطوير أنظمة اتصالات لاسلكية ذكية وفعالة.

Table of Contents:

Table of figures

List of tables

List of abbreviations

AI	Artificial Intelligence.
AMC	Automatic modulation classification
AM	Amplitude modulation
AMR	Automatic modulation recognition
AUC	Area Under The Curve
ASK	Amplitude shift keying
BFSK	Binary Frequency Shift Keying
BPSK	Binary Phase Shift Keying
CNN	Convolution Neural Networks
DL	Deep Learning
DNN	Deep Neural Networks
fi	instantaneous frequency
FM	Frequency Modulation
FSK	Frequency Shift Keying
GMSK	Gaussian Minimum Shift Keying
I\Q	In-phase and Quadrature
ML	Machine Learning
PM	Phase Modulation
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
ROC	Receiver Operating Characteristic Curve
ReLu	Rectified Linear Unit Function
SNR	Signal-to-Noise Ratio
XGBoost	eXtreme Gradient Boosting

General Introduction

In today's era of advanced wireless communications and rapidly expanding digital infrastructure, the accurate transmission and interpretation of information are fundamental to global connectivity. At the heart of this process lies modulation, the technique by which information is encoded onto carrier signals for transmission. Modulation is essential to ensuring the efficiency and reliability of communication systems across diverse transmission media [1].As communication technologies evolve to support increasingly complex and high-capacity networks, there is a growing demand for intelligent and automated methods to analyze and classify modulation schemes. While traditional signal processing techniques have proven effective under controlled conditions, they often struggle in real-world scenarios where noise, interference, and signal distortions are prevalent [2]. This challenge has spurred significant interest in developing more robust and adaptive modulation classification approaches. Recent advancements in ML and DL offer promising alternatives to conventional methods. These data-driven techniques excel at identifying patterns, learning features directly from raw input, and making accurate predictions with minimal manual intervention. In particular, CNNs have shown impressive results in modulation recognition tasks, thanks to their ability to capture intricate spatial and temporal patterns in signal data. Additionally, ensemble learning methods such as Random Forest and XGBoost provide robust, interpretable solutions capable of handling structured features with high accuracy [3]. This thesis compares the performance of CNN, Random Forest, and XGBoost for modulation recognition using the RadioModRec-1 dataset, which simulates real-world wireless conditions with diverse modulation types, Signal to noise ratio (SNR), and channel effects. The study includes key preprocessing steps like constellation diagram analysis and In-phase and Quadrature (I/Q) normalization to improve accuracy. The research aims to identify the most effective model while evaluating the trade-offs between DL and classical ML in terms of accuracy, robustness, computational complexity, and generalization, contributing to the advancement of intelligent wireless communication systems. The structure of this thesis is as follows:

- Chapter 1 presents the theoretical background of communication system architectures and both analog and digital modulation techniques.
- Chapter 2 discusses the application of ML and DL methods in signal classification, with a focus on CNN, Random Forest, and XGBoost.
- Chapter 3 provides a comparative evaluation of the models using various performance metrics and real-world signal data, concluding with key insights and suggestions for future research.

Chapter One: Modulation and Communication System

I.1 Introduction

This first chapter introduces the fundamental concepts underlying modern communication systems and the modulation techniques essential for efficient signal transmission. It begins with an overview of communication system architectures, components, and classifications based on medium, signal type, and communication scope. The chapter focuses on the concept of modulation, emphasizing its critical role in adapting signals for effective transmission. Both analog and digital modulation methods are discussed in detail, including their operating principles, mathematical formulations, and practical applications. Specific schemes of analog modulation, as well as digital techniques are explored to provide a solid technical foundation for the classification tasks addressed later in this thesis.

I.2 Overview of Communication Systems

Communication systems play a crucial role in connecting people and devices around the world. They encompass the transmission and reception of information through various technologies and transmission methods **[4]**. These systems are designed to ensure the accurate delivery of messages via different media, including air, cables, and optical fibers. Depending on the specific application, communication may occur in real time or with some delay, and can involve both simple and complex forms of data. From wireless communication in smartphones to data transfer across computer networks and satellite connections, communication systems are essential to modern life, facilitating seamless interaction in both personal and professional settings.

I.2.1 Definition of Communication System

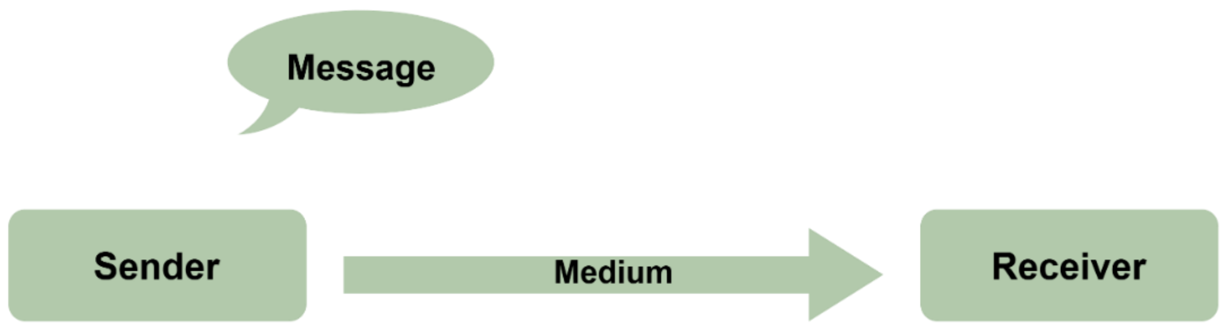

Figure I.1: Simple Communication Process.

A communication system is a model that describes the exchange of information between two or more stations as it is showing in figure I.1, typically involving a sender and a receiver [5]. It outlines the process by which a signal travels from its source to its destination through a transmission channel. Before a signal can be transmitted, it must undergo several processing steps beginning with signal representation, followed by shaping, encoding, and modulation. Once the signal is properly prepared, it is transmitted through the communication channel. During transmission, the signal may encounter various impairments such as noise, attenuation, and distortion, which can affect the quality and integrity of the received message [6].

I.2.2 Types of Communication Systems

Communication systems can be classified according to several criteria**,** including the transmission medium used, the direction of information flow, the underlying technology, the type of information being transmitted [7], and the scope of communication. Below is a summary of the main categories:

1. **Based on Transmission Medium**

Communication systems can be classified into wired and wireless systems based on the medium used for signal transmission.

- Wired communication relies on physical conductors such as copper wires or fiber optic cables. Common examples include traditional telephone networks, cable television, and Ethernet connections. Wired systems generally offer higher reliability and are less prone to interference. However, they can be expensive and challenging to install over long distances.
- Wireless communication, on the other hand, transmits signals through electromagnetic waves without the need for physical cables. Examples include mobile phone networks, Wi-Fi, Bluetooth, satellite communication, and broadcast television. Wireless systems provide greater flexibility and mobility but are typically more susceptible to interference, noise, and potential security vulnerabilities [8].

2. **Based on Direction of Communication**

Communication systems may also be categorized according to the direction of information flow, which gives rise to three primary types:

- Simplex systems allow unidirectional communication, where information flows only from the sender to the receiver. The receiver cannot respond or send information back. Examples include television and radio broadcasts. Half-duplex systems enable two-way communication, but only one direction at a time. This means that while both parties can transmit and receive information, they cannot do so simultaneously. A common example is a walkie-talkie.

- Full-duplex systems support simultaneous two-way communication, allowing both the sender and receiver to transmit and receive messages at the same time. Examples include telephone calls and video conferencing [9].

3. **Based on Type of Signal**

Another classification of communication systems is based on the type of signal they transmit [10]:

- **Analog communication systems:** transmit continuous signals that vary in amplitude or frequency. Examples include AM/FM radio and analog television. While these systems are relatively simple and inexpensive, they are more susceptible to noise, interference, and signal degradation over long distances.
- **Digital communication systems:** transmit data in binary form (0s and 1s), enabling more robust and efficient transmission. Digital systems offer improved signal quality, enhanced security, and effective error correction. They are widely used in internet communication, modern mobile networks, and digital broadcasting.

4. **Based on Technology Used**

Communication systems can also be classified based on the technology they utilize. The main types include optical, radio, microwave, and satellite communication:

- **Optical communication** uses light signals, typically transmitted through fiber optic cables. It is known for its extremely high data transmission speed and minimal signal loss over long distances, making it ideal for backbone internet and telecommunication networks.
- **Radio communication** relies on radio waves and is commonly used in mobile phones, radios, and various wireless networks. It offers reliable transmission over medium to long distances, depending on frequency and power.
- **Microwave communication** employs high-frequency radio waves and is often used in radar systems, satellite links, and point-to-point communication over long distances. It requires line-of-sight between the transmitter and receiver.
- **Satellite communication** uses orbiting satellites to relay signals across vast geographical areas. It is particularly effective for global broadcasting, GPS, and communications in remote or inaccessible regions [11].

5. **Based on Communication Scope**

Communication systems can also be categorized by their scope and intended usage. The main types include:

- **Personal communication systems** are designed for individual use and typically support short-range interactions. Examples include smartphones, Bluetooth devices, and personal messaging applications. These systems prioritize convenience, portability, and ease of use.

- **Telecommunication systems** are built to handle large-scale voice and data transmission over wide geographical areas. They include infrastructure such as landline telephone networks and cellular mobile networks, enabling long-distance communication across cities, countries, or continents.

- **Data communication systems** focus on the transmission of digital information between computing devices. These systems form the backbone of computer networks like Local Area Networks (LANs), Wide Area Networks (WANs), and the internet. They are essential for enabling data exchange in homes, enterprises, and industrial environments [12].

I.2.3 Components of Data Communication System

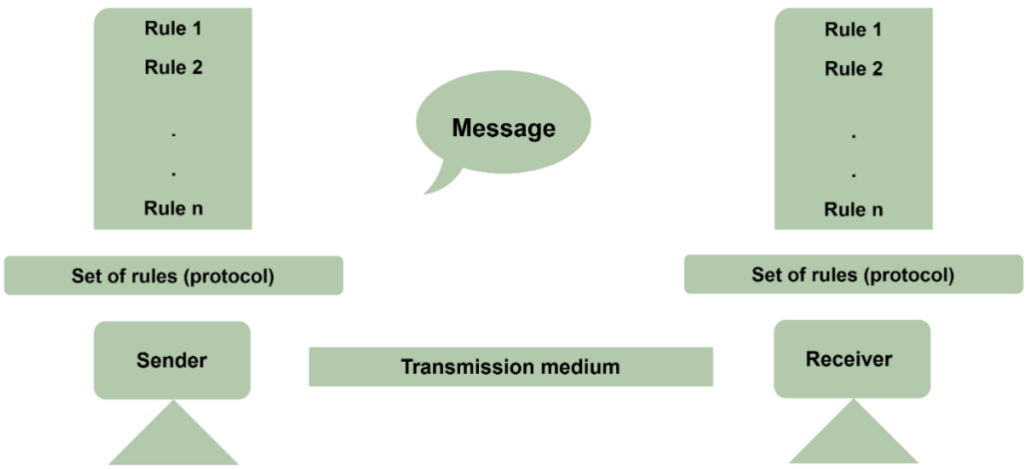

Figure I.2: Basic Components of a Communication System.

A data communication system consists of several key components that work together to enable this transfer [13]:

- **Transmitter:** The device or component that sends the information. It converts the data into signals suitable for transmission.

- **Transmission Medium:** The physical or wireless path through which the signal travels from the transmitter to the receiver. This could be copper wires, fiber optic cables, or radio waves.

- **Receiver:** The device that receives the transmitted signal and converts it back into a usable form for the destination system or user.

These three-core elements transmitter, medium, and receiver are essential for establishing effective communication between two points **[13]**.

There are mainly five components of a data communication system as illustrate in the figure I.2:

- **Message**: This is the most useful asset of a data communication system. The message simply refers to data or piece of information, which is to be communicated. A message could be in any form, it may be in form of a text file, an audio file, a video file [13].
- **Sender:** To transfer message from source to destination, someone must be there who will play role of a source. Sender plays part of a source in data communication system. It is simple a device that sends data message. The device could be in form of a computer, mobile, telephone, laptop, video camera, or a workstation [13].
- **Receiver:** It is the destination where finally message sent by source has arrived. A device receives message. Same as sender, receiver can also be in form of a computer, telephone mobile, workstation [13].
- **Transmission Medium**: In entire process of data communication, there must be something, which could act as a bridge between sender, and receiver, Transmission medium plays that part. Data or message travels from sender to receiver by physical path [13].
- **Set of rules (Protocol):** The protocol is a set of rules that govern data communication. If two different devices are connected but there is no protocol among them, there would not be any kind of communication between those two devices [13].

I.3 Modulation in Communication Systems

Modulation is the process of preparing a signal for transmission by altering a high-frequency carrier signal according to the characteristics of the original information signal such as data, voice, or video. Since raw information signals are often not suitable for direct transmission over communication media whether wireless, wired, or optical modulation ensures that the signal can be transmitted efficiently and effectively. It allows for the use of appropriately sized antennas, extends the transmission range, and improves signal quality. Modulation techniques can be classified as either analog or digital, depending on the type of information being transmitted. The reverse process, known as demodulation, extracts the original information signal from the modulated carrier at the receiver's end **[14]**.

I.3.1 Analog modulation

Analog modulation refers to the process of encoding information onto a carrier signal through continuous variations that mirror the characteristics of the original analog message. In this method, the information signal modifies specific properties of the carrier wave such as amplitude, frequency, or phase in a smooth and continuous manner [15]. This approach is particularly effective for transmitting naturally analog signals, such as human voice and music, as it preserves the original waveform throughout the transmission process. While analog modulation techniques are generally simpler and more cost-effective to implement, they are also more vulnerable to noise, interference, and signal degradation, especially over long distances or in challenging transmission environments. To gain a clearer understanding of analog modulation in practice, it is helpful to explore its primary types, each defined by the specific characteristic of the carrier signal that is varied.

I.3.1.1 Amplitude Modulation (AM)

AM is a fundamental analog modulation technique used to transmit information by varying the amplitude of a high-frequency carrier signal [16]. In this method, the amplitude of the carrier wave changes in direct proportion to the instantaneous amplitude of the modulating (message) signal, while the frequency and phase of the carrier remain unchanged. This means the carrier signal is effectively multiplied by the message signal to encode the information for transmission. As illustrated in Figure bellow:

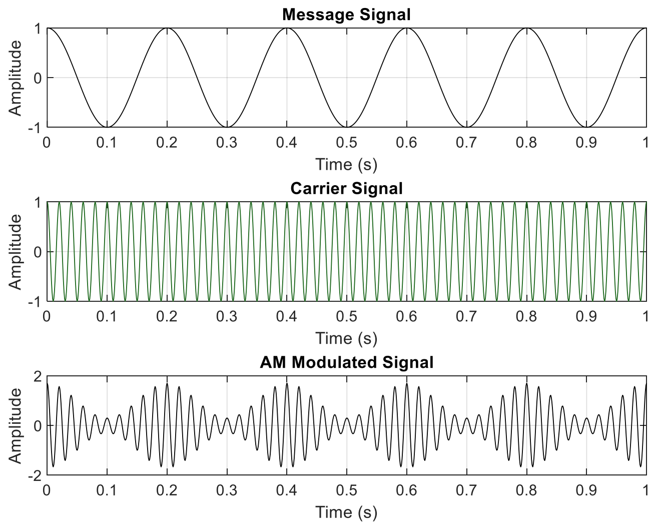

Figure I.3: The visual representation of AM modulation.

- The top plot shows the modulating signal a low-frequency waveform that represents the original information, such as speech or music.

- The middle plot displays the carrier signal a high-frequency sine wave used as the transmission base.
- The bottom plot depicts the resulting AM modulated signal, where the carrier's amplitude varies in accordance with the shape of the modulating signal.

AM is widely used in radio broadcasting and other communication applications due to its simplicity. However, it is relatively more susceptible to noise and signal degradation compared to other modulation techniques. The carrier is a high-frequency sinusoidal signal that acts as a base or support for transmitting information in AM. It is represented mathematically as [17]:

$$u_{c(t)} = A_c \cos(\omega_c t) \tag{1}$$

Where $\omega_p = 2\pi f_p$: The pulsation of the carrier signal is the angular frequency of the carrier. It serves to carry the information over long distances [17].

The modulating signal contains the actual information to be transmitted and is expressed as [17]:

$$u_{m(t)} = A_m \cos(\omega_m t) \tag{2}$$

With $\omega_m = 2\pi f_m$ is the pulsation of the modulating signal.

When modulation occurs, the carrier's amplitude is multiplied by the modulating signal, resulting in the modulated signal [17]:

$$s(\mathrm{t}) = A_c\big(1 + KU_m(t)\big)\cos\ w_c\ (t) \tag{3}$$

or equivalently:

$$s(t) = A_c\ (1 + m\cos(2\pi f_m))\cos(2\pi f_c) \tag{4}$$

Where k is a proportionality factor, and **m** is the modulation index: $m = k\frac{A_m}{A_p}$ (%).

The equation of the modulated wave can be written in the form:

$$s(t) = A_c\ \cos(2\pi f_c) + \frac{A_c m}{2}[\cos(2\pi(f_c + f_m)t) + \cos(2\pi(f_c - f_m)t)] \tag{5}$$

This shows that the AM signal consists of three frequency components: a central carrier frequency f_C an upper sideband at $f_C + f_m$ and a lower sideband at $f_C - f_m$ these sidebands carry the actual information, while the carrier facilitates transmission [17].

I.3.1.2 Frequency Modulation (FM)

FM is a technique used to encode information onto a carrier wave by varying its frequency in response to the instantaneous amplitude of the modulating signal that is, the original message or information being transmitted [18]. In contrast to AM, where the amplitude of the carrier is altered, FM maintains a constant amplitude and instead changes the frequency of the carrier wave. As shown in Figure I.4, the frequency of the carrier wave increases or decreases depending on the strength of the modulating signal at a given moment, while the overall amplitude remains unchanged. This characteristic makes FM significantly more resistant to noise, interference, and signal fading, resulting in higher-quality and more reliable signal transmission especially in audio applications like FM radio broadcasting.

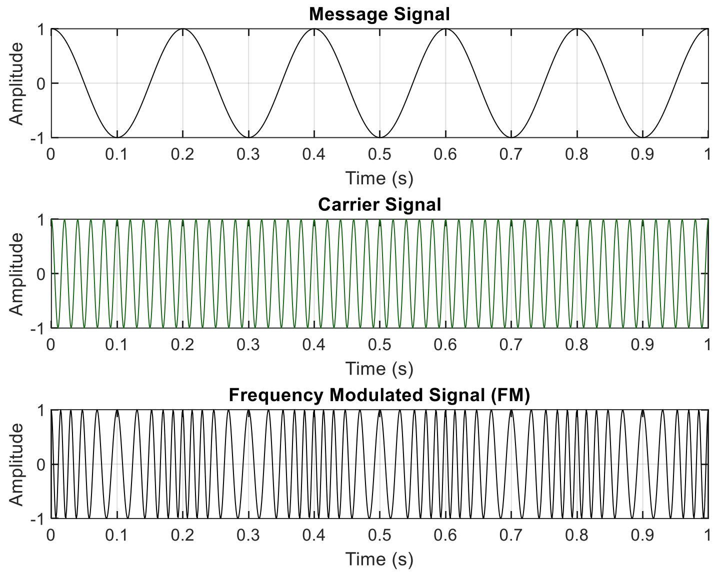

Figure I.4: The visual representation of FM modulation.

This type of modulation is primarily used in radio broadcasting because of its notable advantages in signal quality. One of the key benefits of FM is its high signal-to-noise ratio, which significantly reduces interference from other radio frequencies and electrical noise. This makes FM especially well-suited for transmitting high-fidelity audio, such as music, where clarity and sound quality are essential. As a result, FM has become the preferred modulation technique for most modern radio stations [18].

Let's define the modulating signal u(t) and the carrier signal v(t) as follows [17]:

$$u(t) = A_m \cos\,(2(\pi f_m\ t) \tag{6}$$

$$v(t) = A_c\ \cos(2\pi f_c\ t + \varphi) \tag{7}$$

If the carrier is frequency modulated, this means that f_c varies linearly with the modulating signal u(t). The phase of the carrier signal is given by [17]:

$$2\pi f_c t + \varphi = \delta \tag{8}$$

To understand the relationship between the phase and frequency, we differentiate δ with respect to time. The derivative of the phase gives the **instantaneous frequency** (f_i) [17]:

$$\frac{d\delta}{dt} = \frac{d(2\pi f_c + \varphi)}{dt} = 2\pi f_i => f_i = \frac{1}{2\pi}\frac{d\delta}{dt} \tag{9}$$

This frequency f_i is modulated by the signal u(t), causing it to shift by an amount Δ_f. The instantaneous frequency then becomes [17]:

$$\frac{1}{2\pi}\frac{d\delta}{dt} = f_c + A_m \Delta_f \cos(2\pi f_m t) = f_c + \Delta_f U(t) \tag{10}$$

Next, integrating to find the phase δ [17]:

$$\delta = \int 2\pi f_c\ t + 2\pi \int A_m\ \Delta_f \cos(2\pi f_m t) dt \tag{11}$$

$$\delta = 2\pi f_c\ t + 2\pi \Delta_f \int u(t) dt \tag{12}$$

The expression of the frequency modulated signal s(t) is then:

$$s(t) = A_c \cos(\delta) = A_c \cos\left(2\pi f_c\ t + 2\pi\Delta_f \int u(t)dt\right) \tag{13}$$

$$s(t) = A_c \cos(\delta) = A_c \cos\left(2\pi f_c\ t + 2\pi \int A_m\ \Delta_f \cos(2\pi f_m\ t)\ dt\right) \tag{14}$$

Substituting for u (t), we get:

$$s(t) = A_c \cos(\delta) = A_c \cos\left(2\pi f_c\, t + \frac{A_m \Delta_f}{f_m} \sin(2\pi f_m\, t) + \Phi\right) \tag{15}$$

In this equation, $A_m.\Delta_f$ represents the frequency deviation, which indicates how much the carrier frequency deviates from the center frequency due to the modulating signal. The modulation index

m= $\frac{\Delta_f}{f_m}$ is defined as the ratio of frequency deviation to the modulating signal's frequency

I.3.1.3 Phase Modulation (PM or ΦM)

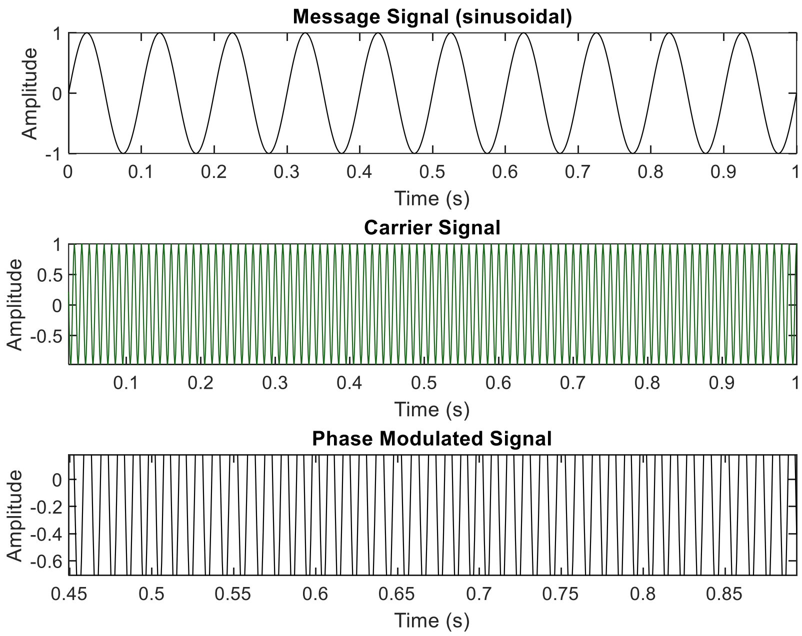

Figure I.5: The visual representation of PM modulation.

PM is a technique used to prepare communication signals for transmission by modifying the phase of a carrier wave in response to the instantaneous amplitude of the message signal [19].In PM, the phase of the carrier changes according to the modulating signal, while its amplitude and frequency remain constant. This method allows the encoded signal to carry information through subtle shifts in the carrier's phase, as depicted in Figure I.5.

This modulation is appreciated for its robust resistance to noise and signal degradation, making it highly reliable in challenging communication environments. As a result, it is often used in radio communications, as well as in critical applications such as drones and UAVs, where stable and precise data transmission is essential [19]. Consider a **modulating signal** u(t) and a **carrier signal** v(t) defined respectively as [17]:

$$u(t) = A_m \cos(2\pi f_m\, t) \quad (16)$$

$$v(t) = A_c\, \cos(2\pi f_c\, t + \varphi) \quad (17)$$

In PM, the phase φ of the carrier signal varies linearly with the modulating signal u(t) As a result, the total instantaneous phase δ of the modulated signal is given by [17]:

$$2\pi f_c\, t + \varphi(t) = \delta(t) \quad (18)$$

Substituting $\delta(t)$ with its linear dependence on the modulating signal, we get:

$$\delta(t) = 2\pi f_c\, t + A_m\, \Delta\theta \cos(2\pi f_m\, t) = w_c + \Delta\theta\, u(t) \quad (19)$$

The f_i of a phase-modulated signal is defined as the time derivative of the total phase δ divided by 2π. Given the expression [17]:

$$f_i = \frac{1}{2\pi}\frac{d\delta(t)}{dt} = f_c - \Delta\theta . A_m . f_m \sin(2\pi f_m\, t) \quad (20)$$

This shows that although the carrier frequency $f_c$remains fixed, the f_i oscillates sinusoidally due to the varying phase.

The **frequency excursion**, which represents the maximum deviation of the f_i from the carrier frequency, is therefore: $\Delta f_{s=}\Delta\theta . A_m . f_m$ [17].

The full expression for the phase-modulated signal is given by:

$$s(t) = A_c\, \cos(\delta) = A_c \cos(2\pi f_c\, t + A_m\, \Delta\theta \cos(2\pi f_m\, t)) \quad (21)$$

By defining the **modulation index**: $m = A_{m\,.}\Delta\theta$ the expression of the phase-modulated signal can be written in a more compact form as:

$$s(t) = A_c\, \cos(2\pi f_c\, t + m \cos(2\pi f_m\, t)) \quad (22)$$

I.3.2 Digital modulation

As previously discussed, analog modulation techniques modify a carrier wave to transmit analog information. In contrast, digital modulation involves encoding a digital information signal by altering one or more characteristics of a carrier wave typically its amplitude, frequency, phase, or a combination of phase and frequency. This process significantly impacts both the bandwidth of the transmitted signal and its resilience to channel impairments such as noise, fading, and interference. A key feature of digital modulation is the grouping of multiple bits into a single symbol, which increases spectral efficiency allowing more data to be transmitted within a given bandwidth. The symbol rate, or baud rate, determines the overall bandwidth requirements for communication. In the following sections, we will explore the most used digital modulation techniques, including their principles, advantages, and typical applications [20].

I.3.2.1 Amplitude Shift Keying (ASK)

ASK is a digital modulation technique in which the amplitude of a carrier wave is varied in accordance with the digital data being transmitted, while the frequency and phase of the carrier remain constant. In simple terms, ASK changes the strength (amplitude) of the carrier signal to represent binary values typically '1' and '0' [21].

ASK can be categorized into two main types:

- **Binary ASK (BASK) or On-Off Keying (OOK)**: This is the simplest form of ASK. A binary '1' is represented by transmitting the carrier signal with a certain amplitude (e.g., 12V), while a binary '0' is represented by the absence of the carrier signal (0V) [22].
- **M-ary ASK (Pulse Amplitude Modulation):** In this version, multiple bits ($\log_2 M$) are transmitted per symbol, with each unique combination of bits assigned a distinct amplitude level. This approach increases data rates by allowing more information to be carried in each transmitted symbol [22].

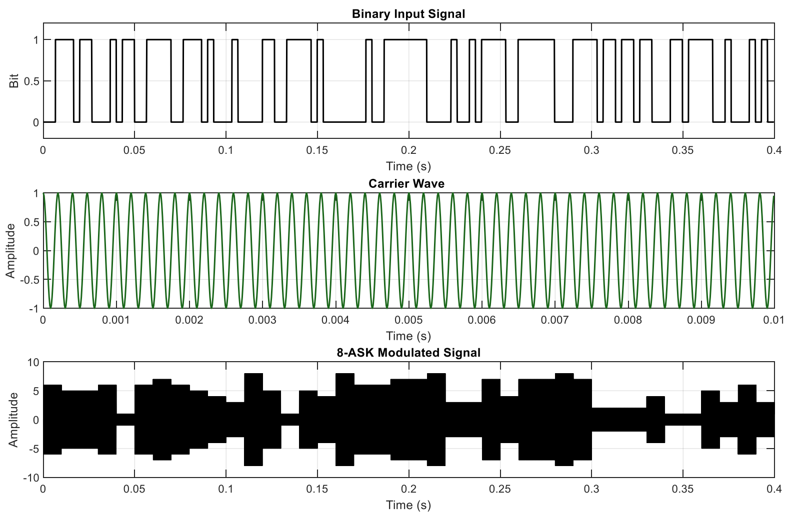

Figure I.6: The visual representation of 8-ASK modulation.

The Figure I.6 demonstrates an example of 8-ASK modulation. The binary stream is grouped into 3-bit symbols (since$2^3 = 8$), with each group mapped to one of eight amplitude levels (from 0 to 7). These levels modulate **a** sinusoidal carrier, resulting in a signal whose amplitude reflects the encoded data. This type of modulation is commonly used in low data rate applications, such as transmitting digital data over optical fibers and in traditional telephone line modems.

Now, let's take a closer look at the implementation of ASK modulation at the hardware level, using the ASK modulator block diagram (Figure I.7). The diagram typically consists of three key components [23]:

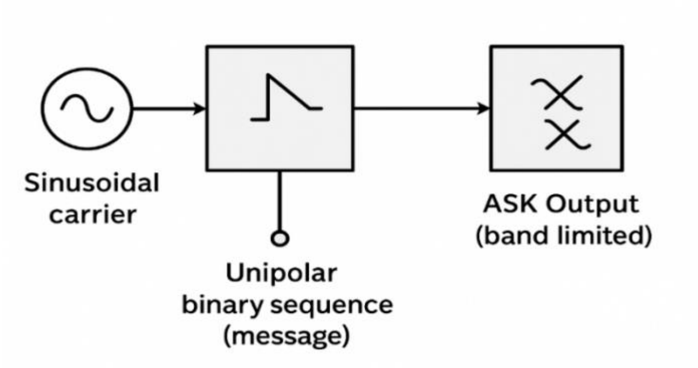

Figure I.7: ASK modulator.

- **Carrier Signal Generator:** This component generates the high-frequency carrier signal, which serves as the base for modulation.
- **Binary Sequence (Message Signal):** The binary data, representing the information to be transmitted, modulates the carrier signal. This binary sequence determines the amplitude variations of the carrier.
- **Band-Limited Filter:** This filter ensures that the modulated signal has the appropriate bandwidth, removing any unwanted high-frequency components that may cause distortion.

I.3.2.1.1 Constellation diagram of ASK modulation

The Figure I.8 shows a constellation diagram representing 8-ASK modulation. In this diagram, eight distinct signal points are aligned along the I axis, with no variation along the Q axis. This indicates that the modulation scheme is one-dimensional, with only the amplitude of the in-phase component being modulated. Each point on the diagram represents a unique symbol, which corresponds to 3 bits of information (since $2^3 = 8$ symbols).

Because the symbols differ only in amplitude, 8-ASK is more susceptible to noise and amplitude distortion compared to multidimensional schemes. However, it offers higher spectral efficiency than lower-order ASK systems by transmitting more bits per symbol.

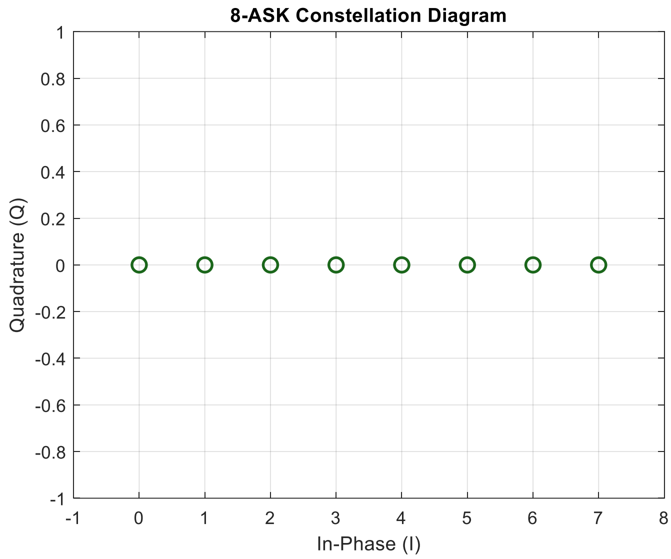

Figure I.8: Constellation diagram of 8-ASK modulation.

I.3.2.2 Frequency shift keying (FSK):

FSK is another fundamental digital modulation technique, where the amplitude of the carrier remains constant, but its frequency is varied according to the digital data being transmitted [24].There are two main types of FSK:

- **Binary Frequency Shift Keying (BFSK):** In BFSK, two distinct frequencies are used to represent binary data. For example, a frequency of 50 MHz may be assigned to logic 1, and a different frequency, such as 25 MHz, to logic 0. The frequency of the carrier wave alternates between these two values, encoding the digital information [25].
- **M-ary Frequency Shift Keying (M-ary FSK):** In M-ary FSK, groups of $\log_2 M$ bits are assigned unique frequencies, enabling more data to be transmitted per symbol. However, as M increases, spectral efficiency decreases, bandwidth requirements grow exponentially, and system complexity rises. This can increase the likelihood of transmission errors and require more sophisticated equipment [25].

The Figure below illustrates the FSK modulation process through three sequential subplots, each highlighting a key stage in the modulation:

- The first plot shows the binary input signal, consisting of a sequence of 0s and 1s over time.
- The second plot represents the carrier signal, a continuous high-frequency wave that serves as the basis for the modulation.

- The third plot displays the BFSK modulated signal, where the frequency of the carrier shifts based on the binary input. A higher frequency is used for bit '1', and a lower frequency for bit '0'. This frequency shift encodes the digital data into an analog waveform, making it suitable for transmission over analog communication channels.

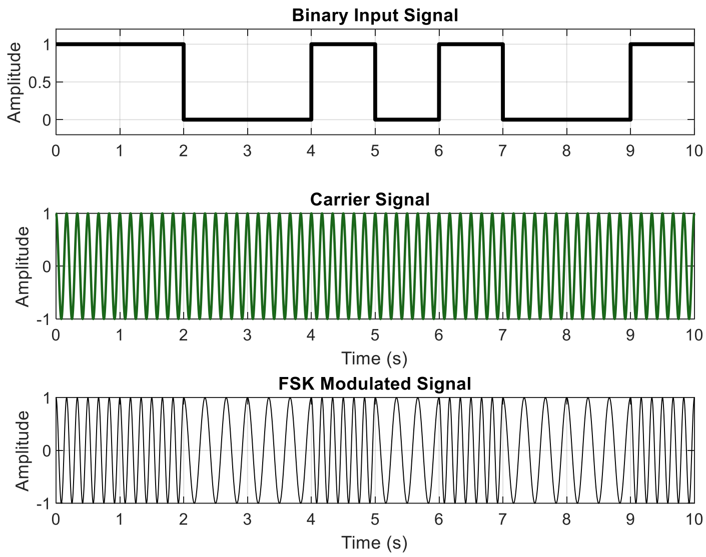

Figure I.9: The visual representation of BFSK modulation.

FSK modulation techniques are widely utilized for their robustness and reliability in various applications, including:

- Telemetry Systems: FSK is commonly used for transmitting sensor data over long distances, where maintaining signal integrity is crucial.
- Two-Way Radios: Devices like walkie-talkies rely on FSK for efficient and clear signal transmission, even in noisy environments, ensuring reliable communication.

These applications benefit from FSK's ability to maintain a stable connection and minimize errors, even in challenging communication conditions.

To better understand how FSK modulation is practically realized, let's explore one of the simplest hardware implementations of a BFSK Modulator (Figure I.10), the oscillator circuit is designed to produce two distinct frequencies, typically one for binary 0 and another for binary 1. These frequencies are then selected by a switch or multiplexer according to the input binary message.

- **Mark Frequency:** The higher frequency generated by the first local oscillator.

- **Space Frequency:** The lower frequency generated by the second local oscillator.

- **Binary Modulating Signal:** This signal determines which frequency is output by the multiplexer.

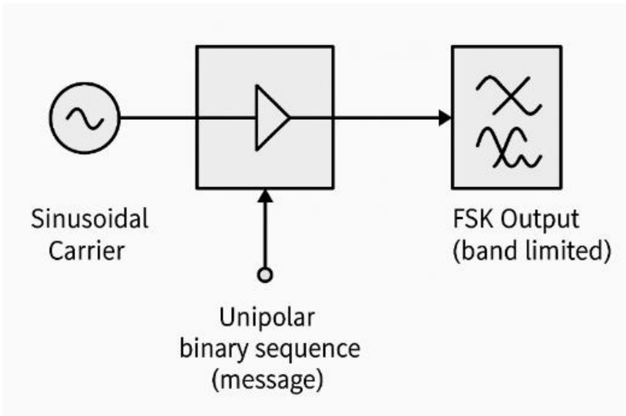

Figure I.10: FSK modulator.

The two oscillators generate the mark and space frequencies, while the binary signal acts as the select signal for the multiplexer. When a binary 1 is received, the multiplexer outputs the mark frequency. Conversely, when a binary 0 is received, it outputs the space frequency [23].

I.3.2.2.1 Constellation diagram of FSK modulation

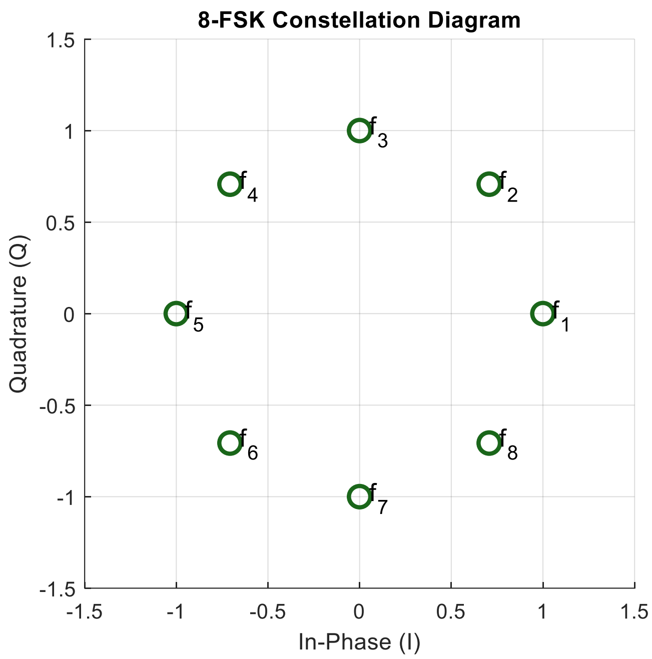

Figure I.11: Constellation diagram of 8-FSK modulation.

This constellation diagram in Figure I.11 represents 8-FSK (8-level FSK) modulation. In FSK, different frequencies are used to represent different symbols. Here's how this diagram fits with 8-FSK. Each point $f1$ to $f8$ corresponds to a different frequency tone, mapped onto the IQ plane after passing through a quadrature receiver or a bandpass-to-baseband conversion. Unlike amplitude- or phase-based schemes, FSK symbols are not confined to a circular or grid structure instead, they appear scattered around the origin, typically lying on a circle or ellipse depending on tone spacing and modulation index.

I.3.2.3 Phase Shift Keying (PSK)

Following ASK and FSK, PSK emerges as the third key digital modulation scheme, where the phase of the carrier wave is shifted in accordance with the digital input. In PSK, the sine and cosine components of the carrier are adjusted at specific time intervals, resulting in a phase shift that encodes binary information, this type of modulation finds widespread application across various communication systems due to its bandwidth efficiency and robustness in noisy environments [20]. It is commonly used in wireless technologies such as Wi-Fi and Bluetooth, as well as in cellular networks including 3G, 4G, and 5G. Additionally, PSK plays a vital role in satellite communication systems and serves as a core modulation technique in digital audio and video broadcasting platforms. There are several variations of PSK, each offering different levels of efficiency and complexity depending on the number of phase shifts used. Two of the most commonly used forms are described below [26]:

- **Binary phase shift keying (BPSK):** Also known as 2-phase PSK or Phase Reversal Keying, BPSK is the simplest form of phase modulation. In this technique, the carrier signal undergoes two distinct phase shifts typically 0° and 180° to represent binary values. Each symbol encodes one bit, making BPSK robust and efficient for low-noise environments, but less bandwidth-efficient than higher-order PSK schemes.
- **Quadrature phase shift keying (QPSK):** QPSK extends the concept of PSK by allowing the carrier to assume four different phase shifts: 0°, 90°, 180°, and 270°. This enables the encoding of **2 bits per symbol**, effectively doubling the data rate compared to BPSK without increasing bandwidth. QPSK strikes a balance between bandwidth efficiency and resilience to noise, making it a widely used modulation scheme in modern digital communication systems.

Beyond BPSK and QPSK, more advanced schemes such as M-ary PSK (where $M > 4$) increase data throughput by encoding multiple bits per symbol, though this comes at the cost of higher complexity and increased susceptibility to noise.

Figure I.12 provides a visual representation of PSK modulation. It illustrates how the phase of the carrier wave changes in response to the binary input signal. Each distinct phase shift corresponds to a unique bit pattern, demonstrating how digital data is encoded into phase variations for transmission. This graphical depiction clarifies the core concept of PSK, where information is conveyed by shifting the phase rather than altering the amplitude or frequency of the carrier.

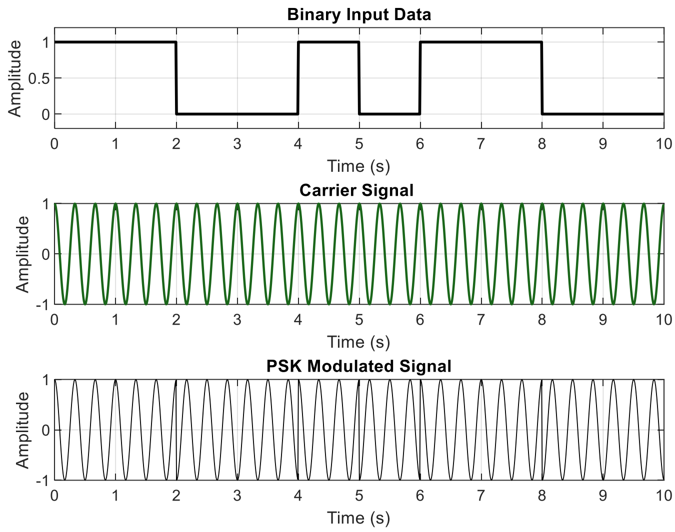

Figure I.12: The visual representation of PSK modulation.

Figure I.13 below presents an overview of a PSK modulator and its operational flow. The system consists of three core components:

- **Sinusoidal Carrier Wave Generator:** Produces the carrier signal.
- **Bipolar Binary Data Source:** Provides the binary sequence representing the digital information.
- **PSK Output:** The resulting modulated signal.

The carrier signal is input into a phase modulator, which is controlled by the bipolar binary sequence. In this sequence, binary '1' is typically mapped to +1, and binary '0' to -1. The phase modulator then shifts the phase of the carrier accordingly, producing a PSK-modulated signal where data is conveyed through distinct phase changes [20].

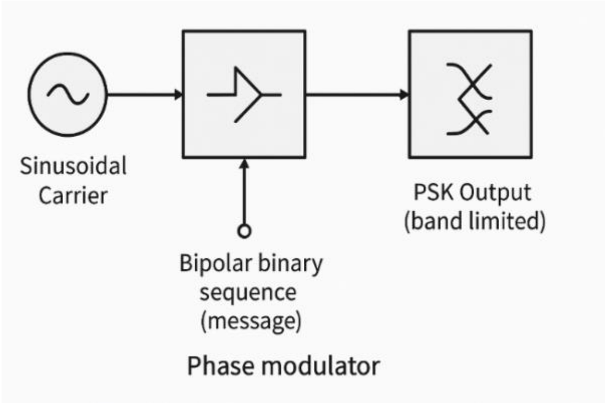

Figure I.13: PSK modulator.

I.3.2.3.1 Constellation diagram of PSK modulation

This constellation diagram in Figure I.14 illustrates 8-PSK. In 8-PSK, the modulation is achieved by varying the phase of the carrier signal while maintaining a constant amplitude. The 8 symbols are evenly distributed on a circle centered at the origin. Each point represents a unique phase shift of the carrier, spaced 45° apart (360°/8), covering the full 360° phase range. All points have approximately the same amplitude (or magnitude), which confirms that the modulation is purely phase-based.

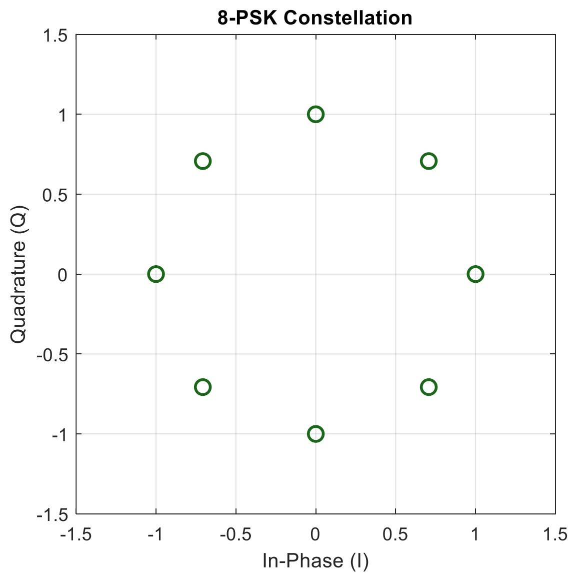

Figure I.14: Constellation diagram of 8-PSK modulation.

I.3.2.4 Quadrature Amplitude Modulation (QAM)

QAM is a modulation technique that combines two carriers, each with different amplitude and phase, to efficiently transmit data. This combination allows QAM to represent multiple bits of information per symbol, significantly increasing the data rate [27]. QAM is widely used in modern communication systems due to its high spectral efficiency. There are various variants of QAM, each defined by the number of unique symbol combinations (or constellation points), such as 4-QAM, 16-QAM, 64-QAM, and 256-QAM [28]. The higher the number of points, the more bits per symbol can be transmitted. However, this also requires a higher SNR to ensure accurate detection. The next Figure I.15 shows the visualization of 16-QAM signal. The explanation is as follows:

- **Top plot:** Represents the binary input signal, a unipolar digital waveform where high amplitude corresponds to binary 1 and low amplitude (zero) corresponds to binary 0. This digital data serves as the message that modulates the carrier.
- **Middle plot:** Shows the sinusoidal carrier wave, which has a constant amplitude and frequency, serving as the base signal for modulation.
- **Bottom plot:** Displays the resulting QAM-modulated signal, where both the amplitude and phase of the carrier wave are varied according to the binary input. As observed, the amplitude changes based on symbol values, and the phase also shifts, demonstrating the essence of QAM.

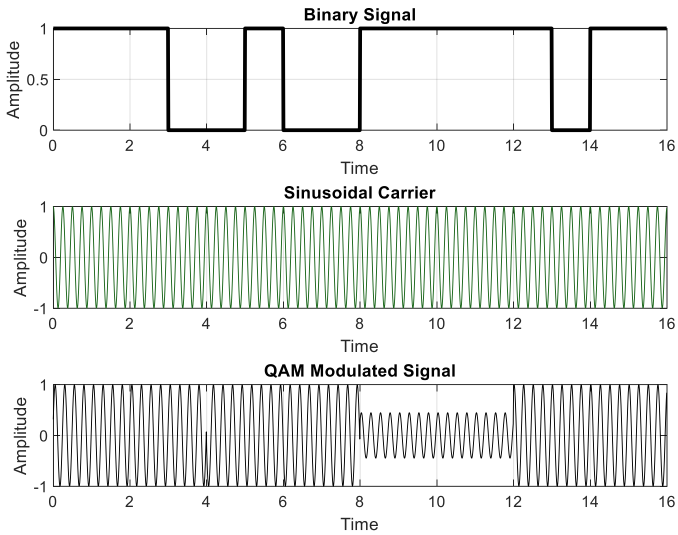

Figure I.15: The visual representation of QAM modulation.

A QAM modulator (Figure I.16) works by combining two signals: an I component and a Q component.

These components are separately modulated using a local oscillator one with a cosine wave (representing the I component) and the other with a sine wave (representing the Q component, which is shifted by 90°). The modulated signals are then summed together, forming a complex waveform that carries both amplitude and phase variations. This technique allows for efficient data transmission by encoding multiple bits per symbol, thus increasing the data rate and improving spectral efficiency [29].

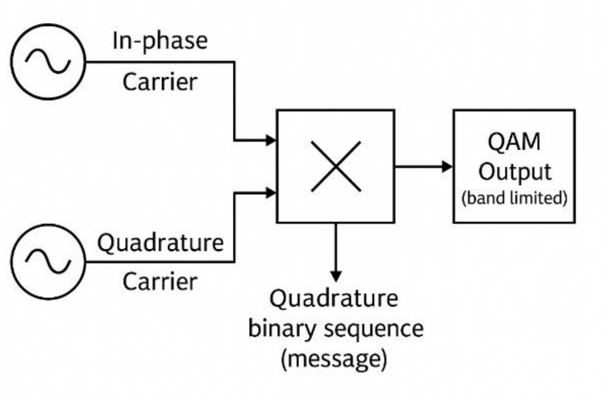

Figure I.16: QAM modulator.

I.3.2.4.1 Constellation diagram of QAM modulation

Figure I.17 presents the 16-QAM modulation scheme through a constellation diagram. The diagram shows 16 distinct symbols, arranged in a 4x4 grid over the I and Q axes. Each point, labeled from s_0 to s_{15}, represents a unique combination of amplitude levels in both the I and Q components. This structure confirms that both amplitude and phase are being modulated, which is characteristic of QAM. The diagram highlights how multiple bits per symbol can be transmitted, increasing data throughput and spectral efficiency.

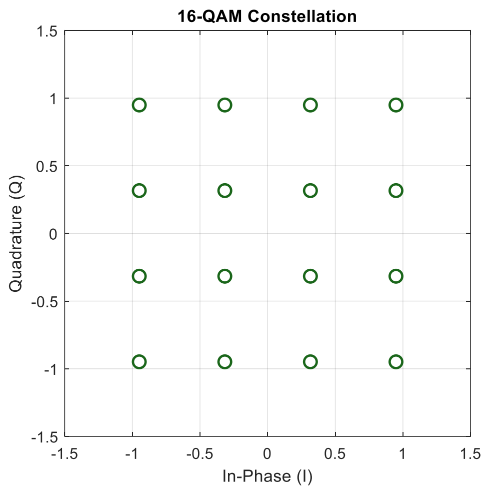

Figure I.17: Constellation diagram of 16-QAM modulation.

I.4 Conclusion

In summary, this chapter has established the essential background for understanding communication systems and the role of modulation in signal transmission. It outlined the structural components and types of communication systems, followed by a comprehensive examination of both analog and digital modulation techniques. Each modulation scheme was explained in terms of its principles, practical use cases, and mathematical representation. This foundational knowledge serves as a critical reference point for the following chapters, where we delve into the use of machine learning and deep learning models for the automatic classification of modulation types based on signal features.

Chapter Two:

Deep Learning and Machine Learning Approaches for Modulation Classification

II.1 Introduction

Building on the foundational principles of communication systems and modulation techniques discussed previously. This chapter explores the integration of ML and DL techniques in signal processing, with a focus on their transformative role in wireless communication and modulation recognition. Traditional signal processing methods, while effective in controlled environments, often struggle with real-world challenges such as noise, nonlinear distortions, and dynamic signal conditions. ML and DL offer data-driven solutions that automate feature extraction and improve adaptability, enabling more robust and scalable systems. The chapter begins with an overview of ML fundamentals, including its definition, mechanisms, and key learning paradigms supervised, unsupervised, semi-supervised, and reinforcement learning. It then delves into DL, particularly CNNs, highlighting their architecture and training processes for tasks like modulation classification. Additionally, traditional ML algorithms such as Random Forest and XGBoost are examined for their interpretability and efficiency in handling structured signal data. By comparing these approaches, the chapter provides a comprehensive framework for selecting and optimizing models to address the evolving demands of modern signal processing applications.

II.2 Overview of ML in Signal Processing

The integration of ML techniques into signal processing has brought about a transformative shift in how signals are analyzed, interpreted, and leveraged across a wide range of applications. Traditional signal processing methods often rely on manually engineered features and domain-specific algorithms. While effective in controlled and well-defined environments, these approaches tend to struggle with variability, noise, and the nonlinearities commonly present in real-world signals. In contrast, ML offers a data-driven framework capable of learning meaningful patterns and representations directly from raw data, providing enhanced flexibility, adaptability, and robustness. This paradigm shift has enabled significant advancements in complex signal analysis tasks and has led to the widespread adoption of ML in areas such as fraud detection, personalized recommendation systems, facial recognition, and automatic language translation all of which fundamentally involve signal processing. The ongoing convergence of ML and signal processing continues to unlock new opportunities for developing intelligent, efficient, and scalable analysis systems [30].

II.2.1 Definition of ML

ML, which falls under the broader field of AI, focuses on the development of algorithms that enable systems to learn from data and improve their performance autonomously, without the need for explicit programming tailored to specific tasks. In domains involving dynamic or structured data such as audio, images, or communication signals signal processing plays a crucial role by transforming raw inputs into meaningful representations that can be more effectively analyzed. Leveraging these representations, ML models can identify underlying patterns and statistical relationships within large and complex datasets. This capability empowers them to support intelligent decision-making and generate accurate predictions across a broad spectrum of applications. As these models are exposed to larger volumes of high-quality, well-processed data, their predictive performance generally becomes more refined and reliable especially when enhanced by advanced signal processing techniques [31].

II.2.1.1 ML Mechanisms

The ML process follows a structured sequence of phases that collectively determine the model's reliability, adaptability, and long-term effectiveness. From data collection to real-world evaluation, each stage must be carefully executed to ensure robust performance across diverse conditions [32]. This workflow is demonstrated in the figure II.1, which illustrates the key steps involved in building and deploying ML models:

1. **Data Collection**

The first phase involves gathering relevant data from appropriate sources such as databases, sensors, simulations, or online repositories. The quality, quantity, and representativeness of the collected data significantly influence the overall success of the ML model [33].

2. **Data Preprocessing**

Raw data is rarely ready for immediate use. It must be cleaned and preprocessed to address noise, missing values, outliers, and inconsistencies. This step ensures that the data is structured, normalized, and suitable for input into ML algorithms [34].

3. **Model Training**

In this phase, a selected ML algorithm is trained using the prepared dataset. The model learns underlying patterns by adjusting internal parameters to minimize a defined error metric. The goal is to capture the relationship between input features and target labels or outputs [34].

4. **Feature Selection and Engineering**

Feature selection involves identifying the most informative variables in the dataset, while feature engineering constructs new features that may improve model performance. These processes help reduce overfitting, improve generalization, and enhance interpretability [35].

5. Model Evaluation and Optimization

Once trained, the model is evaluated using a separate test or validation set. Performance metrics such as accuracy, precision, recall, F1-score are used to assess effectiveness. Hyperparameter optimization techniques may be applied to refine the model further [36].

6. Model Deployment

After achieving satisfactory evaluation results, the model is deployed into a real-world or production environment. Deployment involves integrating the model with existing systems to enable it to make predictions or decisions based on real-time data [37].

7. Monitoring and Updating

Post-deployment, the model's performance must be continuously monitored. Over time, changes in data distributions (concept drift) or new operational conditions may reduce model effectiveness. Regular updates, retraining with new data, and ongoing evaluation are essential to maintaining model accuracy and reliability in a dynamic environment [38].

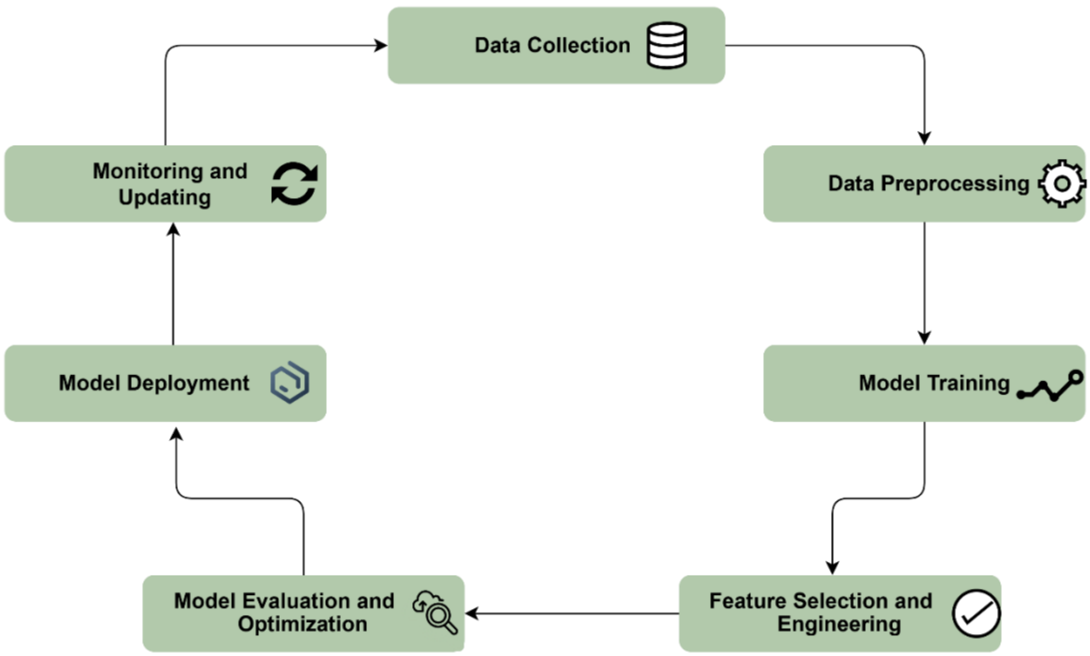

Figure II.18: ML workflow

II.2.1.2 ML methods

As illustrated in Figure II.2, ML techniques are commonly divided into four main categories: supervised, unsupervised, semi-supervised, and reinforcement learning. Each distinguished by the nature of their training data and learning objectives.

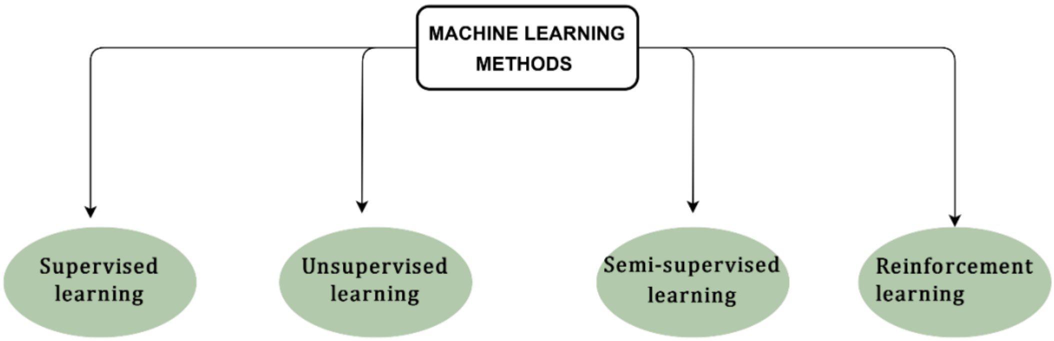

Figure II.19: ML methods

Supervised learning:

Supervised learning is one of the most fundamental and widely adopted approaches in ML, valued for its simplicity and effectiveness in solving well-defined tasks. In this paradigm, the learning algorithm is trained on a labeled dataset, where each input is paired with a corresponding, known output. The goal is for the model to learn the underlying patterns and relationships within the data, enabling it to accurately classify or predict outcomes for previously unseen instances. Supervised learning is particularly effective when a clear and consistent set of labeled examples is available, making it ideal for applications such as image classification, speech recognition, and medical diagnosis [39].

This method is typically divided into two main categories:

- **Classification**: The output variable is categorical, meaning the model assigns input data into distinct classes. For example, classifying emails as spam or not, or predicting the species of a plant based on its features.
- **Regression**: The output variable is continuous, representing a real value. This is used for predicting numerical outcomes, such as forecasting stock prices or estimating weight.

Unsupervised learning:

In the field of AI, unsupervised learning refers to a type of ML in which models learn from data without human supervision. Unlike supervised learning where input data is paired with labeled outputs unsupervised learning algorithms operate on unlabeled data and aim to discover patterns, structures, or relationships without explicit guidance. The primary objective is to uncover hidden groupings, correlations, or underlying structures within the data, enabling deeper insights and understanding without prior knowledge of expected outcomes. This approach is particularly useful in exploratory data analysis, clustering, anomaly detection, and dimensionality reduction [40].

Unsupervised learning is commonly used for tasks such as:

- **Clustering**: Grouping data into subsets or clusters, where elements within the same cluster share similar characteristics. An example is customer segmentation in marketing.
- **Dimensionality Reduction**: Reducing the number of variables in a dataset while preserving its essential features. Techniques like Principal Component Analysis are commonly used for this purpose.

Unsupervised learning is particularly useful when labeled data is hard to obtain or when the goal is to discover unknown structures in the data. However, it can be more challenging to evaluate since there is no "correct answer" to compare the model's results against, in contrast to supervised learning [40].

Semi-supervised learning:

Semi-supervised learning is a hybrid approach that combines aspects of both supervised and unsupervised learning. In this paradigm, the model is trained on a small amount of labeled data alongside a much larger set of unlabeled data. The labeled data provides a foundation for learning, while the unlabeled data helps the model uncover additional patterns and structures to enhance its predictive capabilities. By leveraging the strengths of both approaches, semi-supervised learning enables the model to generalize more effectively and achieve higher accuracy, particularly in scenarios where acquiring labeled data is costly or time-consuming. This method improves learning efficiency and is especially valuable in applications such as text classification, image recognition, and medical diagnostics, where vast amounts of unlabeled data are readily available [41].

Reinforcement learning:

In reinforcement learning, the learning system often referred to as an agent interacts with its environment by performing a series of actions. In return, it receives feedback in the form of rewards: positive when actions lead to beneficial outcomes and negative when they result in undesirable consequences. Unlike supervised learning, the correct action is not explicitly provided; instead, the agent must learn through trial and error. In many cases, rewards may be delayed and only received after a sequence of actions, as is common in complex tasks like playing chess or navigating a robot through an environment. The core objective of reinforcement learning is to learn an optimal policy a strategy that defines the best action to take in each state to maximize the cumulative reward over time [42].

II.2.2 Role of ML in Wireless Communication and Modulation Recognition

ML has emerged as a powerful tool in wireless communication systems, particularly for addressing the challenges associated with modulation recognition. In non-cooperative scenarios where the characteristics of the transmitted signal are unknown ML techniques enable the automatic identification of modulation schemes without requiring prior information. This capability is vital for applications such as cognitive radio, spectrum sensing, signal surveillance, and interference detection. However, signal transmission is frequently affected by various impairments, including noise, multipath fading, hardware imperfections, and synchronization errors. These factors can significantly distort the signal, making accurate modulation classification more difficult. Positioned as a critical intermediate step between signal detection and demodulation, ML-based modulation recognition enhances the reliability of signal interpretation under such adverse conditions [43].

As modern wireless technologies continue to adopt increasingly complex and diverse modulation formats, the demand for robust, adaptive, and intelligent classification models has become more pressing than ever. ML offers the flexibility and scalability required to meet these evolving challenges, ensuring efficient and accurate communication in dynamic environments.

II.3.2 DL Fundamentals

DL, a prominent subfield of ML (Figure II.3), focuses on the development and application of algorithms inspired by the structure and function of the human brain specifically, artificial neural networks (ANNs). It enables the automatic extraction and learning of complex data representations from large-scale datasets, significantly reducing the need for manual feature engineering. Due to its remarkable performance across a wide range of domains including computer vision, speech recognition, and natural language processing DL has become a cornerstone of modern AI research and applications [44].

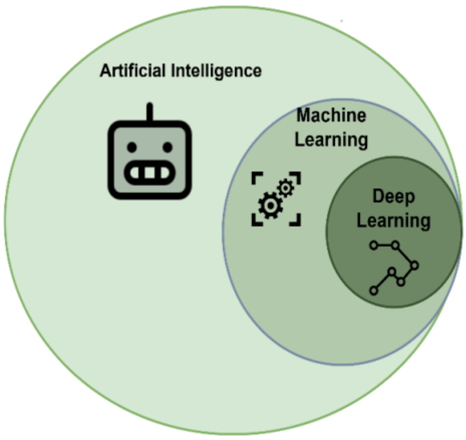

Figure II.20: Structural Overview of AI Subfields.

II.3.2.1 Neural Networks

ANNs form the foundational computational framework underlying DL techniques. These networks consist of layers of interconnected processing units, commonly referred to as "neurons," where each neuron performs a weighted summation of its inputs followed by the application of a nonlinear activation function. This layered architecture enables ANNs to approximate complex, nonlinear functions and to extract hierarchical features from data. Over time, various types of neural networks have been developed to address specific data types and learning tasks. For example, feedforward neural networks (FNNs) are commonly applied to general function approximation problems. Convolutional neural networks (CNNs) excel at processing image and spatial data due to their ability to capture local patterns. Recurrent neural networks (RNNs), on the other hand, are specifically designed to handle sequential or time-dependent data, such as time series and natural language. Deep neural networks (DNNs), which consist of multiple hidden layers, are particularly effective at learning complex and abstract representations from large-scale datasets [45]. Collectively, these architectures enhance the flexibility and power of deep learning in addressing a wide range of real-world challenges.

- **Network Architecture**

A typical neural network consists of three primary types of layers:

- ❖ **Input Layer:** The input layer is where the raw data is introduced to the model. This data could include images, text, or sound, and it serves as the starting point for further processing by the network [46].
- ❖ **Hidden Layers:** The hidden layers (one or more layer), situated between the input and output layers, where the majority of the learning occurs. These layers consist of neurons that apply weights to the input data, transforming it through nonlinear activations. As the data progresses through these layers, the model extracts increasingly complex features, enabling it to recognize patterns at varying levels of abstraction [46].
- ❖ **Output Layer:** The output layer is where the model generates its predictions or classifications. This layer provides the final decision based on the information processed by the preceding layers. For example, in an image classification task, the output layer might classify an image as belonging to a specific category [46].

II.3.2.2 Training Process of DNNs

The training process focuses on fine-tuning its internal weights to reduce prediction errors [47]. This is achieved through an iterative sequence of steps designed to optimize performance over time.

- **Forward Propagation:** Input data passes through the network's layers. Each neuron computes a weighted sum of its inputs, applies a non-linear activation function as it shown in the figure II.4, and transmits the output forward. This results in the network's prediction [48].
- **Loss Computation:** A loss function (e.g., cross-entropy for classification, mean squared error for regression) measures the difference between the network's prediction and the actual target value [49].
- **Backward Propagation:** Using the chain rule, the network computes gradients of the loss with respect to each weight, layer by layer, moving backward from the output to the input. This process identifies how each weight contributed to the error [48].
- **Weight Update (Optimization):** An optimization algorithm (such as stochastic gradient descent or Adam) adjusts the weights in the direction that reduces the loss. This step uses computed gradients and a learning rate hyperparameter [50].
- **Iteration and Convergence:** This cycle is repeated over many epochs (complete passes over the training data) until the loss reaches a satisfactory minimum or a stopping condition is met (e.g., validation accuracy plateaus) [51].

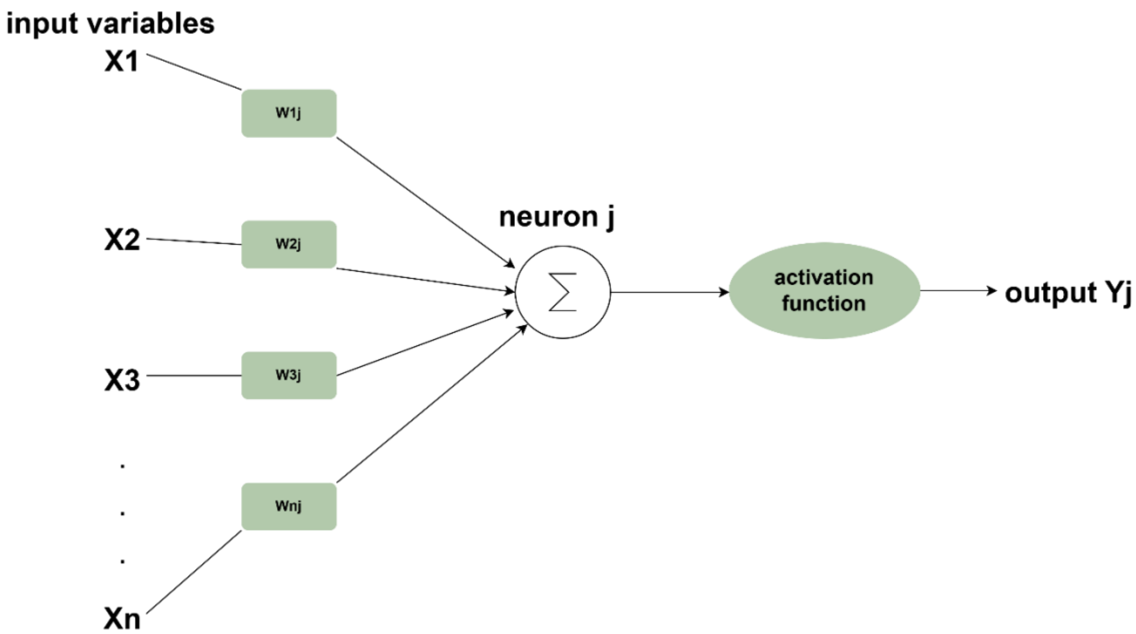

Figure II.21: structure of an Artificial Neuron in a Neural Network.

II.3.2.3 Differences between classical ML and DL:

Traditional ML and DL differ significantly in their approaches to data processing, feature extraction, computational requirements, and application complexity. Traditional ML relies on algorithms such as decision trees, support vector machines, or logistic regression, which require manual feature engineering and typically perform well on structured datasets of limited size. These models are generally easier to interpret, faster to train, and can be run on standard CPUs, making them more accessible in resource-constrained environments. In contrast, DL a subset of ML utilizes multi-layered ANNs to automatically learn features directly from raw data, eliminating the need for manual preprocessing. DL models require large datasets, powerful hardware like GPUs, and longer training and inference times. However, they excel at handling complex, high-dimensional data such as images, audio, and text. While traditional ML approaches solve step-by-step problems using separate components, DL offers an end-to-end learning process where the model simultaneously learns multiple tasks. Despite DL's superior performance in domains like computer vision and natural language processing, its "black-box" nature and high resource demands can restrict its applicability in industries that prioritize transparency and interpretability, such as healthcare and finance [52].

II.4 CNNs

In this project, we investigate the application of CNNs for automatic modulation recognition (AMR) in wireless communication systems. Originally developed for image processing, CNNs have demonstrated exceptional effectiveness across various pattern recognition tasks, thanks to their ability to learn complex, hierarchical representations directly from raw data. This strength is particularly advantageous for modulation recognition, where critical features are often subtle and embedded within noisy or variable signal patterns. Unlike traditional methods that depend on handcrafted feature extraction, CNNs can process raw or minimally preprocessed inputs such as time-domain waveforms or spectrograms allowing them to accurately differentiate between modulation schemes. Moreover, their inherent adaptability to diverse signal conditions significantly enhances their performance in practical, real-world communication scenarios [53].

II.4.1 CNN development

The implementation of a CNN for modulation recognition typically involves two key phases: training and testing. During the training phase, the network adjusts its internal parameters by processing labeled examples and minimizing prediction errors through iterative optimization techniques. This process enables the model to learn and internalize patterns within the signal data that correspond to specific modulation schemes. Conversely, the testing phase assesses the network's ability to generalize by evaluating its performance on previously unseen signals. This evaluation is crucial for validating the model's effectiveness under varying and realistic signal conditions. By adopting this structured approach, researchers can rigorously measure the CNN's capacity to extract discriminative features and apply them successfully in real-world wireless communication environments [54].

II.4.2 Architecture and component of CNN

As illustrated in Figure II.5, the architecture of the CNN used for modulation classification is organized into two primary stages: feature extraction and classification. This design enables the model to automatically learn relevant signal features directly from raw input data and make accurate predictions by processing these features through multiple layers. Each stage serves a distinct purpose feature extraction focuses on identifying key patterns and representations within the input signal, while the classification stage maps these extracted features to specific modulation classes. The following sections provide a detailed explanation of the processes involved in each stage [55].

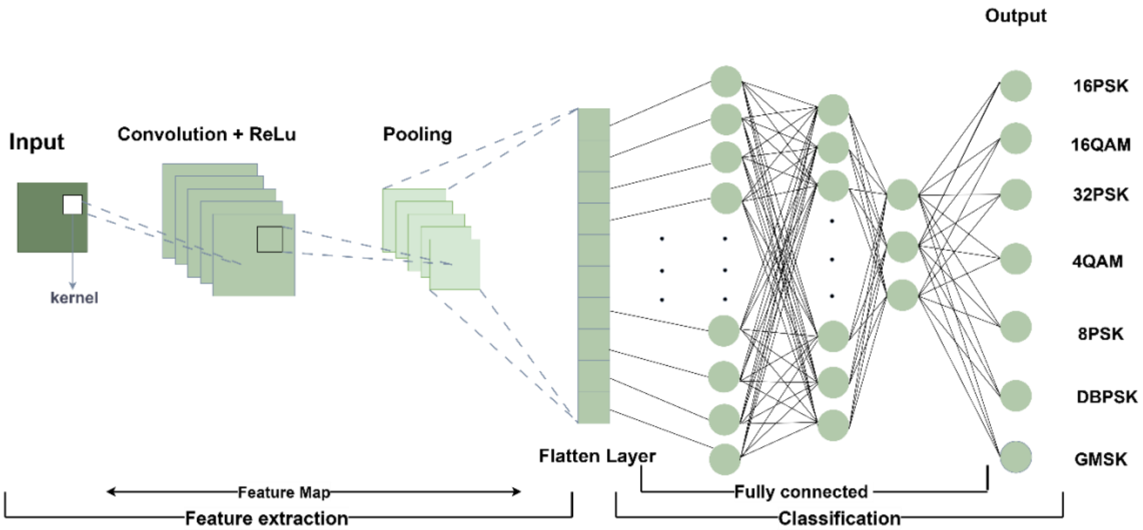

Figure II.22 : CNN Architecture for Modulation Classification.

Feature Extraction Stage

1. **Input Layer**

The CNN starts with an input layer that receives raw signal data, typically represented as I/Q samples. This layer initiates the feature extraction process by applying convolutional operations to capture relevant patterns within the input data [55].

2. Convolution Layers

To extract relevant local features, the CNN utilizes key components namely filters, stride, and padding applied across multiple convolutional layers. This layered approach enables the network to progressively capture increasingly abstract and complex patterns from the data input [55].

1. Filter (kernel)

One of the primary components used to detect local patterns is the filter, also called a kernel. This small matrix slides over the input data, performing element-wise multiplications followed by summations to produce feature maps. The filter's size (e.g., 3×3) determines the region of the input it examines at each step, as illustrated in Figure II.6. By employing multiple filters, the network can capture a wide variety of features from simple edges to more complex and detailed signal structures [55].

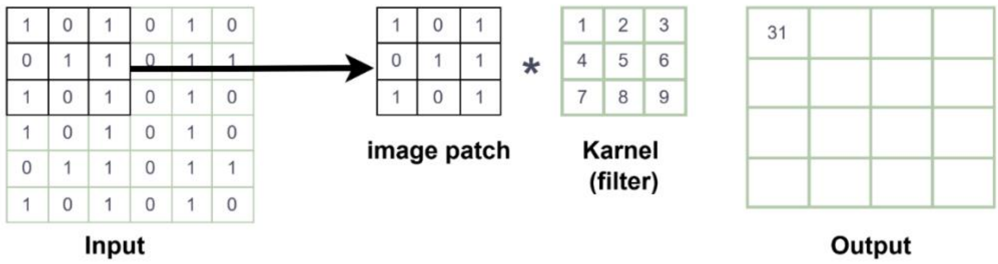

Figure II.23: Convolution Operation in a CNN

2. Stride

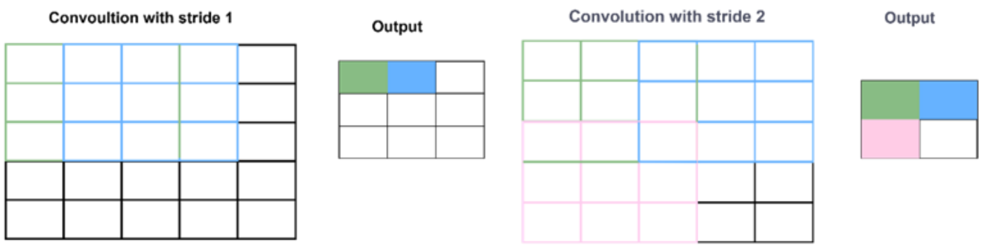

Figure II.24: Effect of Stride on Convolution Output Size in CNNs

To control the movement of the filter across the input, the CNN employs a parameter called stride. When the stride is set to 1, the filter shifts one position at a time, producing a detailed feature map that closely preserves the spatial resolution of the original input. In contrast, a larger stride causes the filter to skip positions, which reduces the output's dimensionality and speeds up computation though this comes at the cost of losing fine-grained spatial details. The impact of different stride values on feature map resolution is illustrated in the following figure [56].

3. Padding

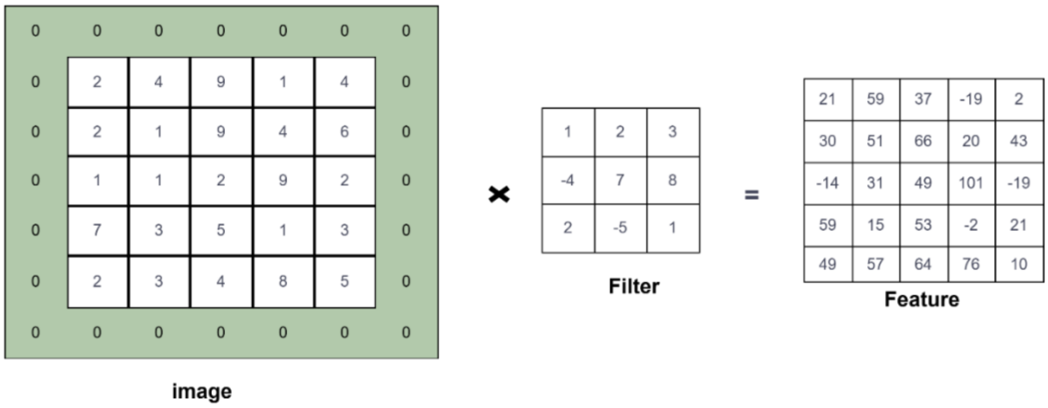

Figure II.25: 2D Convolution with Zero Padding

Finally, to allow the filter to be applied across the entire input including the edges the model uses padding. This technique involves adding extra values, usually zeros, around the borders of the input data. Padding helps preserve the spatial dimensions of the input and prevents the loss of important information, thereby improving the model's ability to learn comprehensive feature representations [57]. This process is illustrated in the figure II.8 below.

3. Activation Layer the Rectified Linear Unit Function (ReLU)

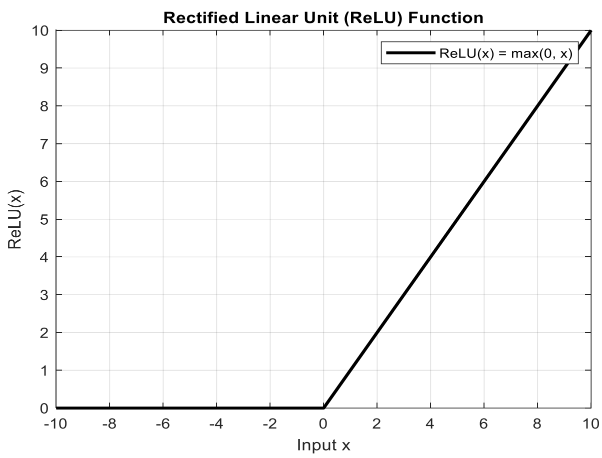

Figure II.26: ReLU Activation Function

Following each convolutional operation, the network applies ReLU activation function, as illustrated in Figure II.9. ReLU introduces essential non-linearity by mapping all negative input values to zero while leaving positive values unchanged. The point z=0 acts as a critical threshold, serving as a decision boundary for neuron activation. Inputs below this threshold are suppressed, resulting in zero output and zero gradient during backpropagation, which prevents the neuron from contributing to learning. Conversely, positive inputs activate the neuron, enabling it to participate in both forward and backward computations. This threshold-based behavior encourages the network to produce sparse representations by filtering out weak or non-informative signals. Furthermore, the inclusion of this non-linear activation function is vital for the network's ability to model complex, non-linear relationships in the data, supporting the extraction of increasingly abstract features in deeper layers. This mechanism is fundamental to achieving high accuracy in tasks such as modulation recognition [55].

1. Pooling

Following the convolutional layers, pooling operations are applied to down sample the feature maps, as illustrated in Figure II.10, which shows both max pooling and average pooling strategies. These operations reduce the spatial dimensions of the feature maps, thereby lowering computational costs and improving the model's robustness to local distortions and noise. Max pooling selects the highest activation within each local region, highlighting the most prominent features, while average pooling computes the mean value, resulting in a smoother and more generalized representation [55].

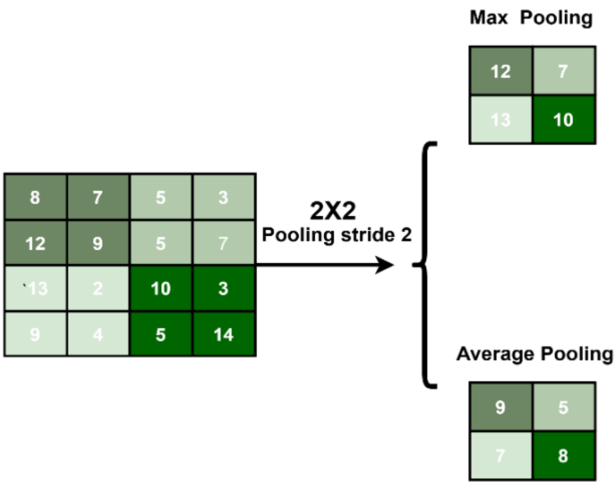

Figure II.27: Max\Average Pooling

Classification Stage

Once the relevant features have been extracted through convolutional and pooling operations, the network transitions to the classification stage, where it interprets the learned feature representations and maps them to specific modulation classes. This stage involves three main components: the flatten layer, the fully connected layers, and the output layer. Together, these components transform the spatial features into a final decision, completing the modulation classification process.

1. Flatten Layer

The first step in this stage is the flatten layer, which reshapes the two-dimensional feature maps produced by the pooling layers into a one-dimensional vector, as shown in Figure II.11.

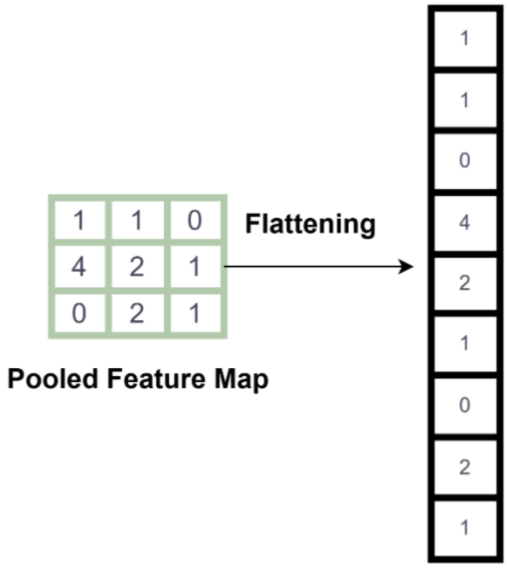

Figure II.28: Example of Matrix Flattening for Neural Network Input

This transformation is essential because the subsequent fully connected layers require input in vector form. By flattening the data, this layer effectively bridges the feature extraction and classification components of the CNN, preparing the information for high-level interpretation and decision-making [58].

2. Fully Connected Layers

Following the flattening process, the fully connected layers (also called dense layers) process the vectorized features by passing them through a series of interconnected neurons. Each neuron computes a weighted sum of its inputs, followed by the application of a non-linear activation function, typically ReLU. These layers enable the network to capture global dependencies across features and perform higher-level abstract reasoning. In the context of modulation recognition, fully connected layers integrate diverse low- and mid-level features to develop a comprehensive understanding of the signal's modulation class [55].

3. Batch Normalization Layer

To enhance training stability and speed, a batch normalization layer is applied after the fully connected layers. This layer standardizes the outputs by normalizing their mean and variance within each mini-batch. By reducing internal covariate shift, batch normalization enables the network to converge more quickly and generalize better to new data. In modulation classification tasks, where signal characteristics can vary widely, this normalization helps stabilize learning and improves robustness [59].

4. Dropout Layer

To improve the generalization ability of the network and reduce the risk of overfitting, a dropout layer is typically applied after the fully connected layers.

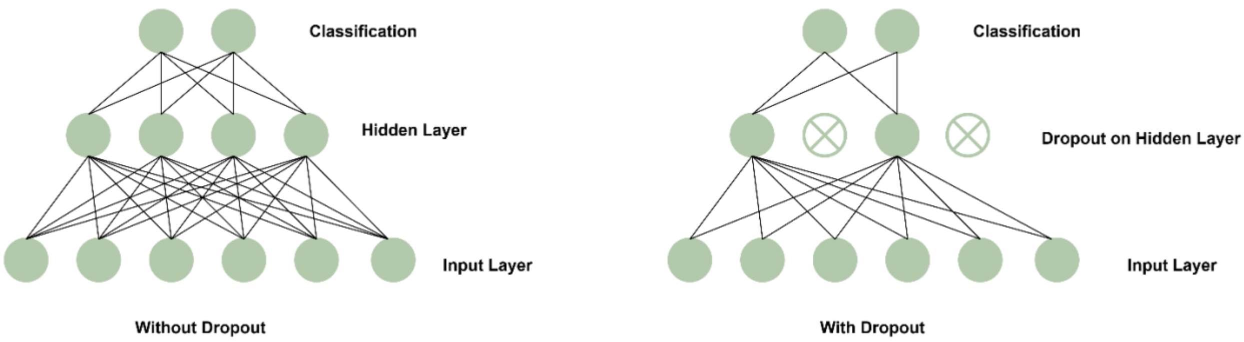

Figure II.29: Comparison of Neural Networks with and Without Dropout

During training, this layer randomly deactivates a fraction of neurons at each iteration based on a predefined dropout rate. This encourages the network to develop redundant and robust feature representations rather than relying on specific activations. In the context of modulation classification, dropout helps the model maintain stable performance even when exposed to noisy or previously unseen signal variations [60]. The effect of this mechanism is illustrated in the next figure II.12.

5. **Output Layer:**

At the final stage, the output layer consists of a number of neurons equal to the total number of modulation classes (e.g., 16QAM, 8PSK, QPSK), as shown in Figure II.5. A softmax function is then applied to the outputs of this layer, converting them into a probability distribution across all classes. The class with the highest probability is selected as the model's prediction. This layer synthesizes all the learned representations from previous stages to deliver the final classification result [55].

II.5 Traditional ML Algorithms

While CNNs have become dominant in modern machine learning tasks, traditional ML algorithms still offer competitive performance, particularly on structured or tabular data. These algorithms are valued for their interpretability, lower computational demands, and effectiveness with smaller datasets. In this study, we compare the performance of CNNs with two widely-used ML algorithms: Random Forest and XGBoost [61].

II.5.1 Decision Trees

Decision Trees are a fundamental type of supervised learning algorithm used for both classification and regression tasks. They operate by recursively partitioning the dataset into subsets based on feature values, forming a tree-like structure. Each internal node corresponds to a decision based on a specific feature, and each leaf node represents a predicted outcome [62].

Decision Trees are valued for their interpretability and simplicity. They can model non-linear relationships and handle both numerical and categorical data. However, a single decision tree is often sensitive to noise in the data and prone to overfitting, especially when deep trees are constructed [62].

Despite these limitations, decision trees serve as the base learners in powerful ensemble methods such as Random Forest and XGBoost. These techniques leverage the strengths of individual trees while addressing their weaknesses, resulting in more robust and accurate models.

II.5.2 Random Forest

Random Forest is a widely used ensemble learning method based on decision trees. It combines the predictions of multiple trees to produce more accurate and stable outputs. The core idea involves two main techniques: bootstrap aggregation (bagging) and random feature selection. Each decision tree is trained on a random sample of the data, and at each node split, only a subset of features is considered. This randomness introduces diversity among trees and enhances generalization performance while reducing the risk of overfitting [63].

In practice, Random Forest is applied to both classification and regression problems. It performs well with high-dimensional, noisy datasets and provides an inherent measure of feature importance, aiding interpretability. In signal classification, Random Forest has been successfully applied to tasks such as modulation recognition, biomedical signal analysis (e.g., ECG and EEG classification), speech processing, and fault detection in sensors [64].

Figure II.13 visually illustrates the Random Forest process: starting from the original dataset, multiple random subsets are sampled; each subset trains an individual decision tree; and the combined output of all trees yields a more robust and accurate prediction.

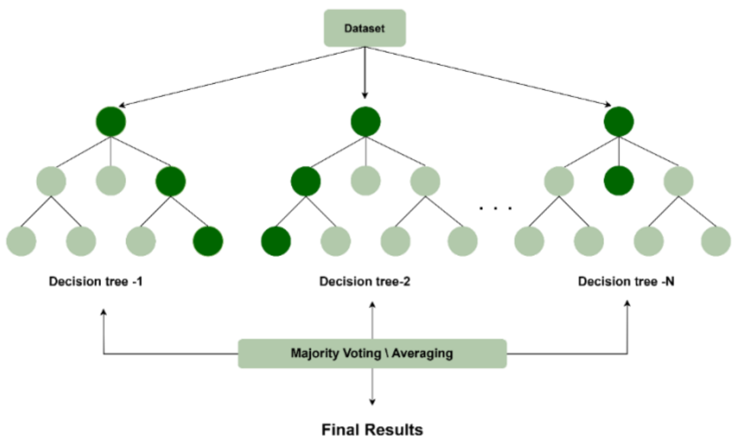

Figure II.30: Random Forest Algorithm.

II.5.2.1 Random Forest Mechanism

The Random Forest algorithm operates using a technique called bootstrap aggregating, commonly known as bagging. This method aims to improve the stability and accuracy of ML models by reducing variance and minimizing overfitting problems often associated with individual decision trees [65].

1. Bootstrap Sampling (Data Diversity)

During the training phase, the algorithm generates multiple decision trees, each trained on a different bootstrap sample. These samples are created by randomly drawing data points with replacement from the original dataset. As a result, some data points may appear multiple times in a single sample, while others may be excluded. This sampling approach ensures that each tree sees a slightly different version of the training data, which promotes model diversity [65].

2. Random Feature Selection (Feature Diversity)

To introduce further randomness and reduce correlation among trees, Random Forest selects a random subset of features at each decision node when determining the best split. Instead of evaluating all available features (as in traditional decision trees), only a limited number ($\sqrt{n}$ for classification, $n/3$ for regression, where n is the total number of features) are considered. This forces each tree to rely on different feature combinations, encouraging varied decision paths and enhancing generalization [66].

3. Independent Tree Construction

Each decision tree is grown to its full depth without pruning, capturing as many patterns as possible from its unique training subset. Because the trees are trained independently on different data and with different features, they learn complementary patterns and make different kinds of errors [67].

4. Prediction Aggregation

Once all trees are trained, the Random Forest makes predictions by aggregating the outputs of the individual trees [67]:

- **For classification tasks**: Each tree casts a “vote” for a class label. The majority vote becomes the final prediction.
- **For regression tasks**: The outputs (numerical predictions) from all trees are averaged to produce the final result.

5. Model Output

The aggregated result is more accurate, stable, and resistant to overfitting than the output of any single tree. The model also provides feature importance scores, indicating which input variables contributed most to the predictive decision-making process [67].

II.5.3 XGBoost

XGBoost is a robust, open-source ML library designed for high performance, scalability, and flexibility. Originally developed by Tianqi Chen, it is widely adopted in both academia and industry for a variety of supervised learning tasks, especially classification and regression. XGBoost extends the traditional gradient boosting framework, an ensemble method that constructs a strong predictive model by sequentially combining multiple weak learners, typically decision trees. The algorithm focuses on optimizing an objective function that balances predictive accuracy (via a loss function) with model simplicity (via regularization) [68].

II.5.3.1 XGBoost Mechanism

XGBoost operates on the principle of gradient boosting, where decision trees are built sequentially, with each new tree aiming to correct the prediction errors made by the previous ones. The algorithm follows a structured, iterative process:

1. **Initialization with a Base Model**: The process begins by constructing an initial decision tree, often predicting the mean of the target variable in regression tasks. This serves as the base model for subsequent improvements [69].
2. **Error Computation:** After generating the initial predictions, the model calculates the residuals, the differences between the predicted values and the actual target values. These residuals represent the errors that the next tree will attempt to correct [70].
3. **Training on Residuals:** a new decision tree is trained on these residuals. The purpose of this tree is to model the errors of the previous prediction and provide corrective adjustments [71].
4. **Model Update:** The predictions are updated by adding the new tree's output to the existing model's predictions. This additive process continues, with each tree refining the overall prediction.
5. **Iteration:** The above steps are repeated iteratively. At each stage, a new tree is trained to minimize the remaining errors. The process continues until a stopping criterion is reached, such as a predefined number of trees or minimal improvement in the loss function [70].

6. **Final Prediction:** The final model aggregates the contributions of all individual trees, combining them to produce a robust and accurate prediction [70].

The accompanying figure clearly shows the principle of Gradient Boosting, where each tree is trained on the residuals of the combined predictions of all previous trees. This sequential correction of errors is what gives boosting its power: instead of relying on a collection of independent learners as in bagging, it builds a strong model by combining the strengths of multiple weak learners, each one progressively improving the overall prediction.

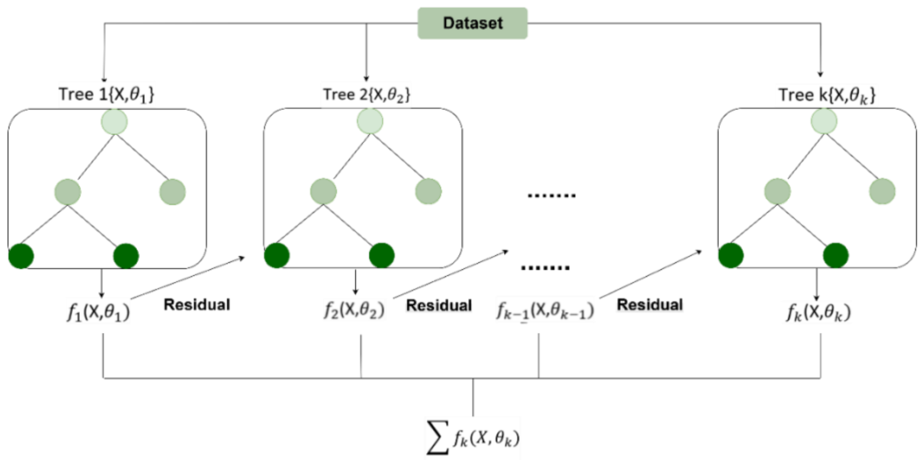

Figure II.31: Gradient Boosting Algorithm

II.5.4 bagging vs boosting

Bagging (Bootstrap Aggregating) and Boosting are both ensemble learning techniques designed to improve the performance and robustness of ML models. Bagging, as used in algorithms like Random Forest, builds multiple independent models (typically decision trees) in parallel using randomly resampled subsets of the training data. The results are then aggregated commonly via majority voting or averaging to reduce variance and prevent overfitting. In contrast, Boosting is a sequential approach where each new model is trained to correct the errors made by the previous ones, focusing on the hardest examples. Gradient Boosting refines this idea by using gradient descent to minimize a loss function, iteratively adding models (usually shallow trees) that predict the residuals of the combined ensemble. XGBoost enhances traditional gradient boosting through regularization, weighted quantile sketching, sparsity-aware learning, and efficient parallel computation, resulting in faster training and superior performance on structured data. However, tuning XGBoost requires careful selection of hyperparameters such as learning rate, maximum tree depth, subsampling ratios, and the number of boosting rounds. When properly tuned, XGBoost offers excellent accuracy, scalability, and generalization, making it a dominant tool in ML competitions and real-world applications alike [72].

II.6 Conclusion

This chapter has systematically examined the application of ML and DL techniques in signal processing, emphasizing their impact on wireless communication and modulation recognition. Beginning with foundational ML concepts, the discussion progressed to advanced DL architectures like CNNs, which excel in automatically extracting discriminative features from raw signal data. Traditional ML methods, including Random Forest and XGBoost, were also evaluated for their robustness and computational efficiency in structured signal analysis. The comparative analysis of these approaches underscores their respective strengths CNNs for high-dimensional pattern recognition, Random Forest for interpretability and noise resilience, and XGBoost for handling complex, imbalanced datasets. Together, these techniques offer versatile solutions for real-world challenges such as spectrum sensing, cognitive radio, and biomedical signal analysis. As the field advances, the synergy between signal processing and ML/DL will continue to drive innovation, paving the way for intelligent, adaptive systems capable of operating in increasingly dynamic and noisy environments. This chapter serves as a guide for researchers and practitioners aiming to leverage these technologies to enhance the accuracy, efficiency, and scalability of signal processing systems.

Chapter Three: Results and Discussion

III.1 Introduction

This chapter presents a comprehensive evaluation of three ML models CNN, Random Forest, and XGBoost for the task of AMC. The performance of each model is rigorously assessed using key metrics such as accuracy, precision, recall, F1-score, and The Receiver Operating Characteristic Curve – Area Under the Curve. These evaluations provide critical insights into the strengths and limitations of each approach, particularly in distinguishing between complex modulation schemes under varying signal conditions. The findings aim to guide the selection of optimal models for real-world wireless communication systems, where accurate modulation recognition is essential for reliable data transmission.

III.2 Performance criteria

Evaluating performance is a crucial step in any AI or ML project, as it determines how well a model can generalize to unseen data. In the context of modulation classification, selecting appropriate evaluation metrics is particularly important. It ensures a fair and comprehensive comparison of different models and provides meaningful insights into their effectiveness. The choice of metrics directly influences how accurately the model's performance can be interpreted especially in complex signal classification tasks, where distinguishing between modulation schemes is essential [73].

To thoroughly understand a model's capabilities, multiple key performance indicators must be considered. Each metric offers a distinct perspective on model behavior, contributing to a well-rounded evaluation. This is especially critical when working with imbalanced datasets, which are common in signal processing applications where certain modulation types may be overrepresented while others are rare.

This section introduces and defines the core performance metrics used to assess the proposed models, emphasizing the specific role each plays in evaluating classification performance within the modulation domain.

1. Confusion matrix

The confusion matrix is a fundamental tool for evaluating the performance of classification models by comparing predicted labels to actual ground-truth classes. In multi-class classification tasks, it extends beyond the binary case into an $n \times n$ matrix, where n represents the number of distinct classes. Each row corresponds to an actual class, while each column represents a predicted class. Correct predictions are located along the diagonal, whereas off-diagonal elements indicate misclassifications between specific class pairs. This structure offers a detailed view of the model's classification behavior and helps identify which classes are most frequently confused.

More than just a count of correct and incorrect predictions, the confusion matrix serves as the foundation for calculating key performance metrics such as accuracy, precision, and recall. These metrics provide a more nuanced and reliable evaluation of model performance, especially in the presence of class imbalance, where relying on accuracy alone can be misleading [74].

2. **Accuracy**

Accuracy is one of the most fundamental metrics used to evaluate the overall performance of a ML model. It measures the proportion of correct predictions made by the model out of the total number of predictions, typically expressed as a percentage. Accuracy is intuitive, easy to compute, and particularly effective when the dataset is balanced meaning that all classes are equally represented. However, in cases of class imbalance, accuracy may give a misleading impression of model performance, as it does not account for the distribution of errors across different classes [75]. The formula for calculating accuracy is:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{23}$$

Where:

- TP **(True Positives):** Correctly predicted positive instances.
- TN **(True Negatives):** Correctly predicted negative instances.
- FP **(False Positives):** Incorrectly predicted positive instances.
- FN **(False Negatives):** Incorrectly predicted negative instances.

While accuracy offers a quick, overall indication of a model's effectiveness, it becomes less reliable in the presence of imbalanced datasets. In scenarios where one class significantly dominates for example, when 95% of instances belong to the "negative" class and only 5% to the "positive" class a model could achieve a high accuracy score simply by always predicting the majority class. Although such a model might appear effective based on its accuracy, it would entirely fail to detect instances of the minority class, thereby missing the true objective of the classification task [75].

Relying solely on accuracy in these cases can lead to a misleading interpretation of the model's performance. To obtain a more comprehensive and informative evaluation, especially in imbalanced contexts, it is essential to consider additional metrics. Precision, recall, and most importantly, the F1-score provide deeper insights into how well the model differentiates between classes. These metrics are particularly valuable for assessing the model's ability to correctly identify minority class instances, which are often of greater interest in real-world applications.

3. Precision

Precision is a key performance metric that evaluates the reliability of a model's positive predictions. It measures the proportion of true positive instances among all instances that the model has classified as positive. In other words, precision quantifies the model's ability to avoid FP, indicating how accurate its positive predictions are. A high precision score implies that when the model predicts a positive outcome, it is very likely to be correct. This is especially important in applications where FP carry a significant cost or risk, such as in medical diagnoses or fraud detection [75]. Mathematically, precision is expressed as follows:

$$precision = \frac{TP}{TP + FP} \tag{24}$$

4. Recall

Recall, also known as sensitivity or true positive rate, evaluates a different aspect of model performance by emphasizing completeness. It measures the proportion of actual positive cases that the model correctly identifies. Unlike precision, which assesses the accuracy of positive predictions, recall focuses on the model's ability to detect all relevant positive instances. A high recall value indicates that the model effectively minimizes FN cases where positive instances are incorrectly classified as negative. This metric is especially critical in domains were failing to identify a positive case has serious consequences, such as in medical diagnosis, fraud detection, or security threat analysis [76]. Recall is defined using this formula:

$$Recall = \frac{TP}{TP + FN} \tag{25}$$

5. F1-score

The F1-Score is a crucial evaluation metric that combines precision and recall into a single, balanced measure of a model's performance. It is particularly valuable in scenarios involving imbalanced datasets, where one class significantly outweighs the others, and where both FP and FN must be minimized. The F1-Score is especially appropriate when precision and recall are equally important, as it ensures that a model is not rewarded for excelling in one metric while performing poorly in the other. By computing the harmonic mean of precision and recall, the F1-Score penalizes extreme disparities between the two, resulting in a more conservative and informative performance indicator [73]. The F1-Score is calculated using the following formula:

$$F1 = 2 \times \frac{precision + recall}{precision \times recall} \tag{26}$$

An F1-Score of 1 represents perfect precision and recall, indicating ideal predictive performance, while a score closer to 0 reflects poor model effectiveness.

6. The ROC

While the confusion matrix and derived metrics like precision, recall, and F1-score provide detailed insights at a specific decision threshold (usually 0.5), The ROC curve offers a broader perspective by evaluating a model's performance across all possible thresholds. The ROC curve plots the TP rate (Recall) against the FP Rate (FP rate = FP / (FP + TN)), illustrating the trade-off between sensitivity and specificity as the threshold varies. This is especially useful in applications where the cost of FP and FN must be balanced carefully [77].

7. The AUC

To quantify the overall performance represented by the ROC curve, the AUC values range from 0 to 1, where 1 indicates perfect classification and 0.5 reflects random guessing. A higher AUC signifies better model capability to distinguish between classes. In imbalanced datasets, where metrics like accuracy can be misleading, ROC and AUC provide a more robust and threshold-independent view of model discrimination [78].

III.3 Dataset description

The dataset employed in this study is the RadioModRec-1 dataset, a publicly available large-scale collection of radio frequency signals designed specifically for automatic modulation recognition research. This dataset was chosen due to its comprehensive coverage of diverse modulation schemes, realistic channel conditions, and suitability for training and evaluating DL models. The RadioModRec-1 dataset provides a robust foundation for developing and validating modulation classification algorithms, ensuring that experimental results are both reliable and applicable to real-world wireless communication scenarios.

III.3.1 Modulation Types and Signal Representation

The dataset encompasses a wide variety of digital and analog modulation techniques, ensuring a thorough evaluation of classification models across different modulation complexities. Key modulation schemes included in the dataset are:

- PSK: The dataset contains multiple PSK variants, such as 8-PSK, 16-PSK, 32-PSK, and 128-PSK, each characterized by distinct phase separations between symbols. These modulations are particularly useful for evaluating a model's ability to discriminate between closely spaced phase transitions.

- QAM: Higher-order QAM schemes, including 16-QAM, 64-QAM, and 256-QAM, are included to assess classification performance under amplitude-phase hybrid modulation. These schemes are widely used in modern communication systems due to their spectral efficiency.
- Differential Modulation: The dataset incorporates differential variants such as DBPSK (Differential Binary PSK) and DQPSK (Differential Quadrature PSK), which encode information in phase differences rather than absolute phases, making them more resilient to phase ambiguities.
- FSK: Continuous-phase and Gaussian-filtered FSK schemes, including CPFSK (Continuous-Phase FSK), GFSK (Gaussian FSK), and GMSK (Gaussian Minimum Shift Keying), are included to evaluate classification in frequency-modulated signals.
- AM: The dataset also includes AM-DSB (Amplitude Modulation with Double Sideband), an analog modulation technique, to test the model's ability to handle non-digital modulation formats.

Each signal in the dataset is represented as complex baseband samples (I-Q components), which capture both I and Q components of the modulated signal. This representation is crucial for preserving amplitude and phase information, enabling the extraction of meaningful features for classification. Additionally, constellation diagrams derived from these I-Q samples are used to visualize the modulation schemes, providing insights into symbol distribution and noise resilience.

III.3.2 Feature Extraction

The **feature extraction** process focuses on identifying distinctive characteristics from the I-Q components and constellation diagrams to enhance modulation classification. Key features extracted include:

1. **Symbol Spacing**: Measures the Euclidean distance between adjacent symbols in the constellation. This feature is critical for distinguishing PSK and QAM schemes, where greater spacing improves noise resilience.
2. **Symbol Density**: Quantifies how tightly symbols are packed in the constellation. Higher-order modulations (e.g., 64-QAM, 256-QAM) exhibit denser symbol distributions, making them more susceptible to noise.
3. **Phase Shift**: Captures angular variations of symbols in the I-Q plane, essential for identifying PSK-based modulations (e.g., 8-PSK, 16-PSK).

4. **I and Q Components**: Represent the real and imaginary axes of the signal, providing amplitude and phase relationships critical for modulation discrimination.
5. **Symbol Clustering**: Identifies patterns in symbol groupings (e.g., grid-like for QAM, circular for PSK), aiding in differentiation.
6. **Symmetry and Regularity**: Evaluates structural patterns (e.g., radial symmetry in PSK, rectangular lattices in QAM) to classify modulation types.
7. **Point Density Distribution**: Analyzes uneven symbol distribution across quadrants, useful for detecting asymmetrical modulations.
8. **Decision Boundaries**: Defines geometric separations between modulation classes, optimizing classification accuracy.

III.3.3 Channel Conditions and Realism

To ensure the dataset reflects real-world communication environments, the signals are subjected to varying channel conditions, including:

- Different SNR Levels: The dataset includes signals across a broad SNR range (e.g., -20 dB to 20 dB), allowing for robustness testing under both high-noise and low-noise conditions.
- Multipath Fading and Interference: Some signals are affected by multipath propagation and interference, simulating challenges encountered in practical wireless systems.
- Variable Symbol Rates: Certain modulation schemes are transmitted at different symbol rates to assess the model's adaptability to varying transmission conditions.

These variations make the dataset highly realistic, ensuring that trained models generalize well to real-world scenarios where channel impairments are common.

III.3.4 Preprocessing and Dataset Structure

Before training, the dataset underwent several preprocessing steps to enhance model performance and ensure consistency:

1. Normalization: The I-Q samples were normalized using Z-score normalization, scaling the features to zero mean and unit variance. This step is critical for stabilizing the training of DL models and improving convergence.
2. Label Encoding: Each modulation class was assigned a unique numerical label, converting categorical modulation types into a format suitable for ML algorithms.

3. Stratified Train-Test Split: The dataset was divided into 80% training data and 20% test data using stratified sampling. This ensures that both subsets maintain a balanced representation of all modulation classes, preventing bias in model evaluation.

The dataset is large-scale, with a substantial number of samples per modulation type, providing sufficient data for training deep neural networks without overfitting. The exact size and distribution of the dataset are detailed in the original publication [79], which confirms its suitability for benchmarking state-of-the-art AMR models.

III.3.5 Advantages of the RadioModRec-1 Dataset

The RadioModRec-1 dataset offers several key advantages for modulation classification research:

- Diversity of Modulation Schemes: The inclusion of both simple and complex modulation types allows for comprehensive model evaluation.
- Real-World Applicability: The incorporation of varying SNR levels and channel impairments ensures that models trained on this dataset perform well in practical scenarios.
- Benchmarking Potential: Since the dataset has been used in prior research, it enables direct performance comparisons with existing state-of-the-art methods.

III.4 Model Evaluation and Performance Analysis

This study presents the results obtained from the classification of digital modulation schemes using three distinct ML models: CNN, Random Forest, and XGBoost. These models were applied to analyze and classify modulation types based on their distinctive signal features, with the overarching goal of achieving high accuracy and robustness for deployment in real-world communication systems. The following sections provide a detailed evaluation of each model's performance, highlighting key metrics such as accuracy, precision, recall, F1-score, and AUC-ROC. A comparative analysis is conducted to assess the strengths and limitations of each approach, offering insights into their effectiveness and suitability for modulation classification tasks under various signal conditions.

III.4.1 CNN Results

This section presents the evaluation results of the CNN model applied to the task of modulation classification. The model's performance is assessed using three key classification metrics: precision, recall, and F1-score, as previously defined in the methodology section. These metrics offer a comprehensive view of the CNN's ability to accurately identify each modulation type, capturing both the correctness and completeness of its predictions.

III.4.1.1 CNN Classification Metrics per Class

The bar chart in Figure III.1 illustrates the CNN model's classification performance across various modulation types, based on precision, recall, and F1-score. Certain modulation schemes, such as DBPSK and 4QAM, exhibit both high precision and high recall, resulting in correspondingly high F1-scores. This indicates that the model effectively identifies a large proportion of TP samples for these classes while minimizing both FP and FN.

For instance, in the case of DBPSK, nearly all true DBPSK signals are correctly classified (high TP, low FN), and the majority of the predictions labeled as DBPSK are indeed accurate (low FP). This reflects the model's strong discriminatory capability and low confusion with other modulation types for these particular classes.

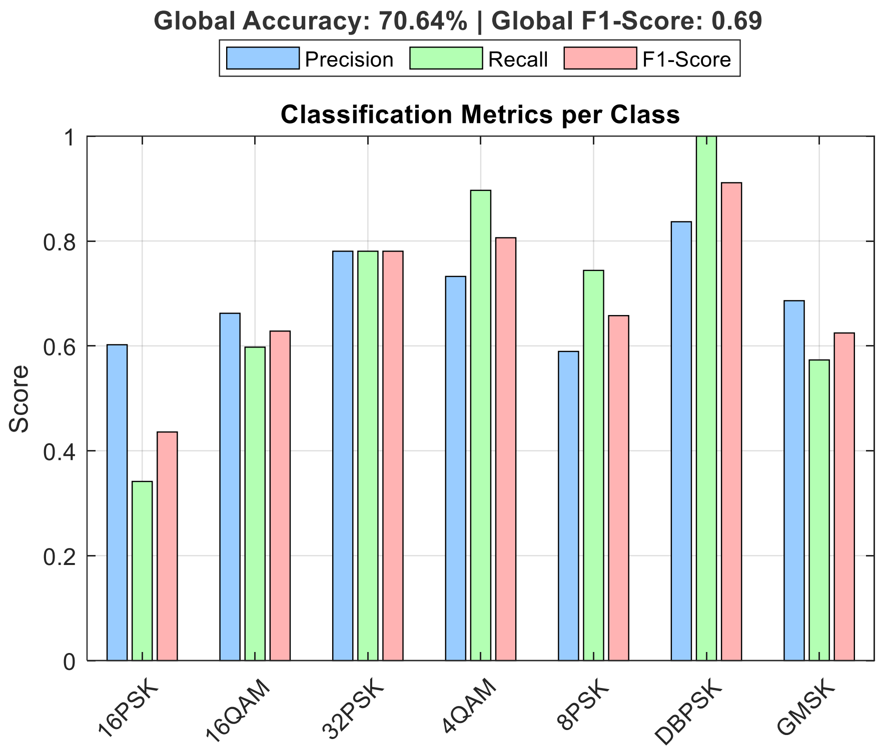

Figure III.32: Precision, Recall, and F1-Score per Class for CNN Model.

In contrast, modulation types such as 16PSK, GMSK, and 8PSK show markedly lower values for all three metrics. For 16PSK, the low recall suggests a high number of false negatives, indicating that many actual 16PSK signals are misclassified as other types. Simultaneously, the low precision reflects a high false positive rate, where signals from other modulation classes are incorrectly labeled as 16PSK. This pattern suggests that the model struggles to differentiate 16PSK from similar modulation schemes, possibly due to overlapping signal characteristics or insufficiently distinctive features learned during training.

In summary, high performance (high TP, low FP/FN) is observed in classes where the modulation pattern is distinct and well captured by the model. Conversely, poor performance (low TP, high FP/FN) is seen in classes with more complex or less distinguishable signal characteristics, or in cases where the training data may be imbalanced or insufficient. The F1-score, as a balance between precision and recall, clearly reflects these variations, emphasizing the model's strengths and weaknesses in handling each modulation class.

III.4.1.2 CNN Confusion Matrix

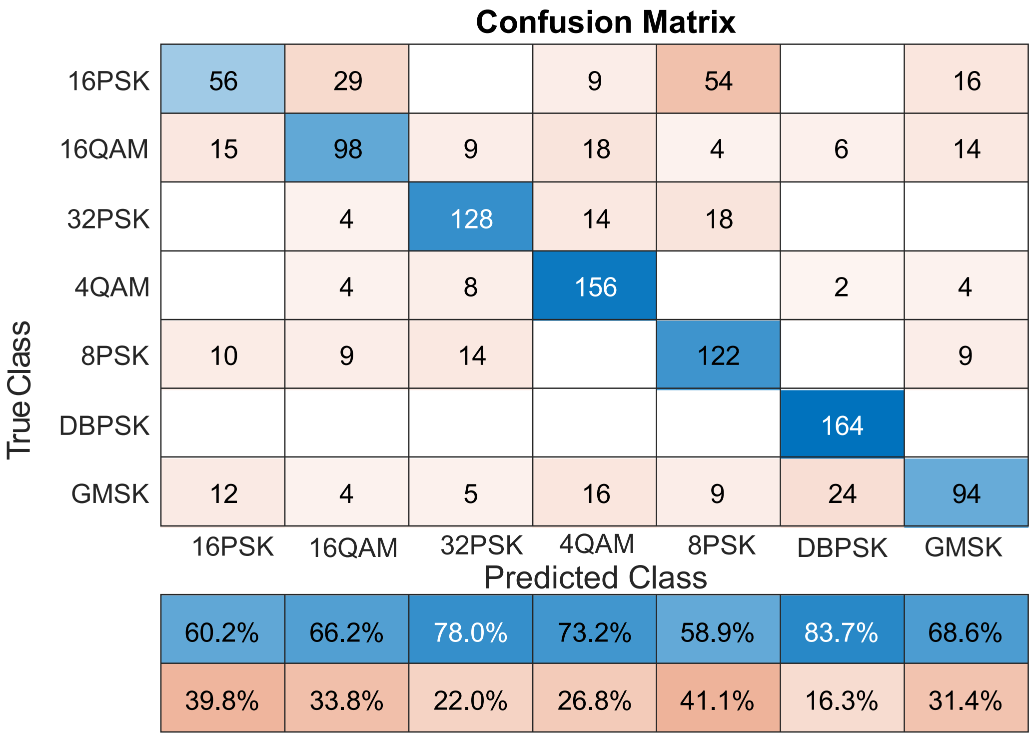

Figure III.33: Confusion Matrix for CNN Classification of Modulation.

The confusion matrix presented in Figure III.2 offers a detailed evaluation of the multi-class classification model tasked with distinguishing among seven digital modulation formats: 16PSK, 16QAM, 32PSK, 4QAM, 8PSK, DBPSK, and GMSK. By comparing predicted labels (columns) with ground truth labels (rows), the matrix provides a comprehensive view of the model’s classification behavior, including both correct and incorrect predictions. Accurate classifications appear along the diagonal where the predicted class matches the true class while off-diagonal entries reveal specific misclassification patterns. The model achieves its highest classification accuracy on DBPSK (83.7%), followed by 32PSK (78.0%) and 4QAM (73.2%), suggesting that these modulation formats possess distinctive signal characteristics that are effectively captured by the model. In contrast, 8PSK (58.9%) and 16PSK (60.2%) yield the lowest class-wise accuracies, indicating difficulties in differentiating these schemes from other modulation types. A closer inspection reveals several notable misclassification trends. For example, 16PSK is frequently misclassified as 8PSK (54 instances) and 16QAM (29 instances), highlighting the model's limitations in discriminating between high-order phase modulation formats. Similarly, 8PSK is often confused with 32PSK and 16QAM, likely due to similarities in constellation shapes or overlapping spectral features. GMSK, another challenging class, is commonly misidentified as DBPSK (24 instances) and 4QAM (16 instances), suggesting possible overlaps in signal envelope or time-frequency representations.

These results underscore the current model's limitations in accurately separating closely related modulation types. They highlight the need for improved feature extraction strategies or the adoption of more advanced model architectures capable of capturing subtle inter-class variations in signal characteristics.

III.4.1.3 ROC Curves by Class

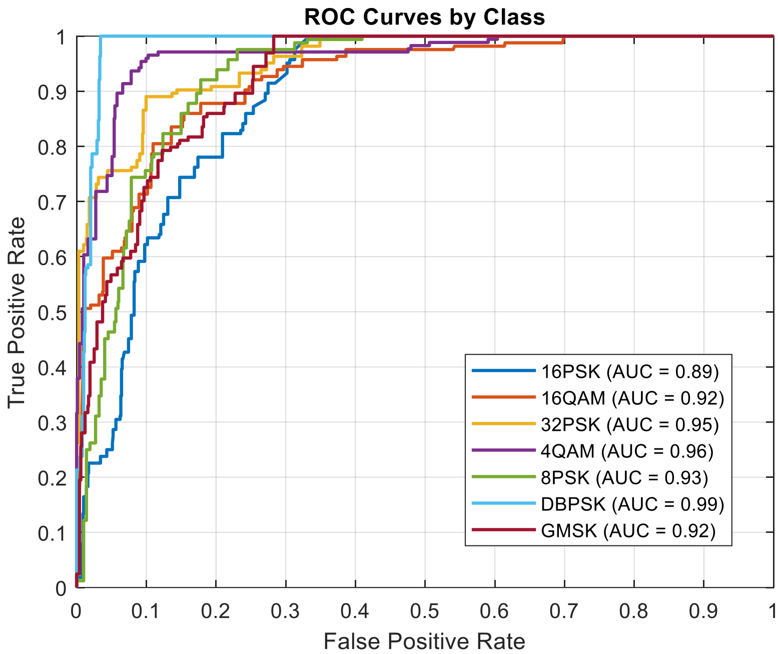

Figure III.34: ROC Curves for CNN Classification of Modulation.

The ROC curve presented in Figure III.3 provides a comprehensive evaluation of the model's classification performance across multiple modulation schemes. The x-axis denotes the FP rate, while the y-axis represents the TP rate, thereby illustrating the trade-off between sensitivity and specificity for each modulation type. Each individual curve corresponds to a specific class, and the AUC quantifies the model's ability to distinguish that class from the others. The classifier demonstrates robust performance across all modulation formats, with AUC values ranging from 0.89 to 0.99. DBPSK achieves the highest AUC of 0.99, indicating near-perfect separability and classification reliability. This is followed by 4QAM (0.96), 32PSK (0.95), and 8PSK (0.93), all of which reflect strong model performance. 16QAM and GMSK both exhibit solid discrimination capabilities with AUC scores of 0.92. The lowest performance is observed for 16PSK, which attains an AUC of 0.89, suggesting comparatively reduced separability—likely due to its similarity to other high-order phase modulation formats.

Overall, the ROC analysis confirms the model's high effectiveness in classifying diverse modulation types, particularly excelling in the detection of DBPSK. These findings complement the earlier metric-based evaluations, reinforcing the model’s suitability for real-world signal classification tasks.

III.4.2 Random Forest Results

The following section presents the evaluation results of the Random Forest model applied to the modulation classification task. To maintain consistency and ensure a fair comparison with the previously evaluated CNN model, the same set of performance metrics is employed. These include classification metrics such as precision, recall, and F1-score, a confusion matrix to visualize prediction accuracy and misclassification patterns, and ROC curves along with their corresponding AUC scores to assess the model’s ability to distinguish between different modulation types. This comprehensive evaluation framework facilitates a thorough analysis of the Random Forest model’s strengths and limitations in classifying complex signal formats.

III.4.2.1 Random Forest Classification Metrics per Class

The bar chart in Figure III.4 illustrates a marked improvement in the Random Forest model’s classification performance across all modulation types, reflected in a high overall accuracy of 97.32% and a global F1-score of 0.97. Each modulation class achieves consistently high values for precision, recall, and F1-score, with most metrics exceeding 0.95, underscoring the model’s strong ability to accurately differentiate between modulation schemes. The minimal gap between precision and recall for each class indicates that the model maintains a high TP rate while keeping FP and FN to a minimum. Notably, modulation formats such as DBPSK and 4QAM achieve near-perfect scores, suggesting highly reliable classification with negligible confusion. Furthermore, modulation types that previously underperformed—such as 16PSK, 8PSK, and GMSK—now exhibit substantial performance gains. This improvement implies that the model has successfully learned to extract more discriminative features, possibly due to enhanced feature representation, data balancing, or augmentation strategies during training.

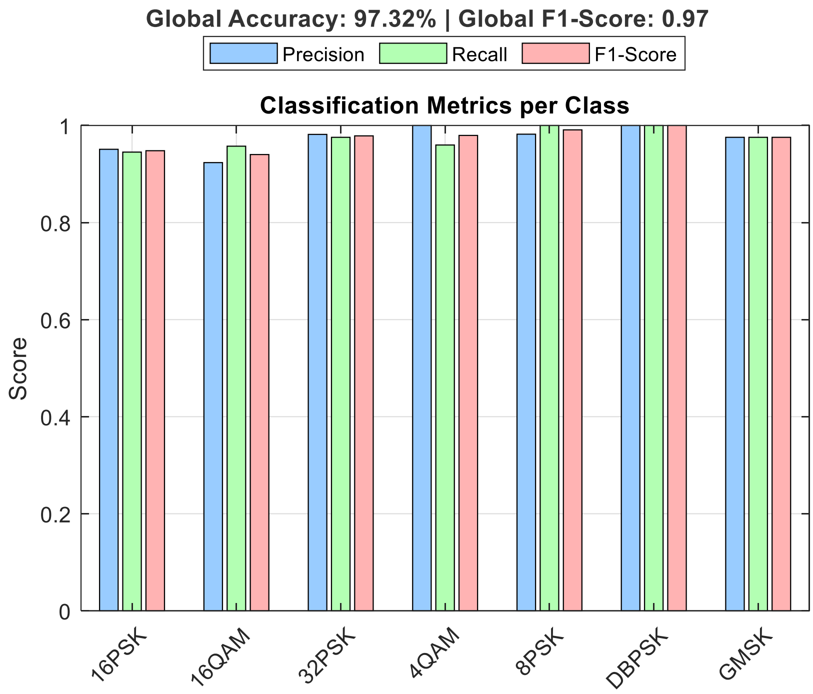

Figure III.35: Precision, Recall, and F1-Score per Class for Random Forest Model.

The consistently high F1-scores across all classes demonstrate the model's ability to maintain a balanced trade-off between precision and recall, highlighting both its robustness and generalization capability. These results confirm the suitability of the Random Forest model for practical deployment in real-world modulation recognition systems, where both accuracy and reliability are critical.

III.4.2.2 Random Forest Confusion Matrix

The confusion matrix presented in Figure III.5 provides a detailed evaluation of the Random Forest model's performance in classifying seven digital modulation schemes: 16PSK, 16QAM, 32PSK, 4QAM, 8PSK, DBPSK, and GMSK. The matrix reflects strong overall classification accuracy, with the majority of correct predictions concentrated along the diagonal, indicating effective model behavior and minimal confusion between classes. The highest accuracies were achieved for 4QAM and DBPSK, both with 100% correct classification, followed by 32PSK (98.2%), 8PSK (98.2%), GMSK (97.6%), and 16PSK (95.1%).

The lowest though still high accuracy was observed for 16QAM, at 92.4%. These results confirm the model's ability to accurately distinguish between modulation types, particularly those with simpler or more distinct constellation structures. Misclassifications were infrequent and generally minor. For instance, 16PSK was occasionally confused with 16QAM (9 instances), while 4QAM exhibited a small number of misclassifications as 8PSK or GMSK. Similarly, 16QAM showed limited confusion with

16PSK and 32PSK, reflecting some overlap in spectral or constellation features among higher-order modulations.

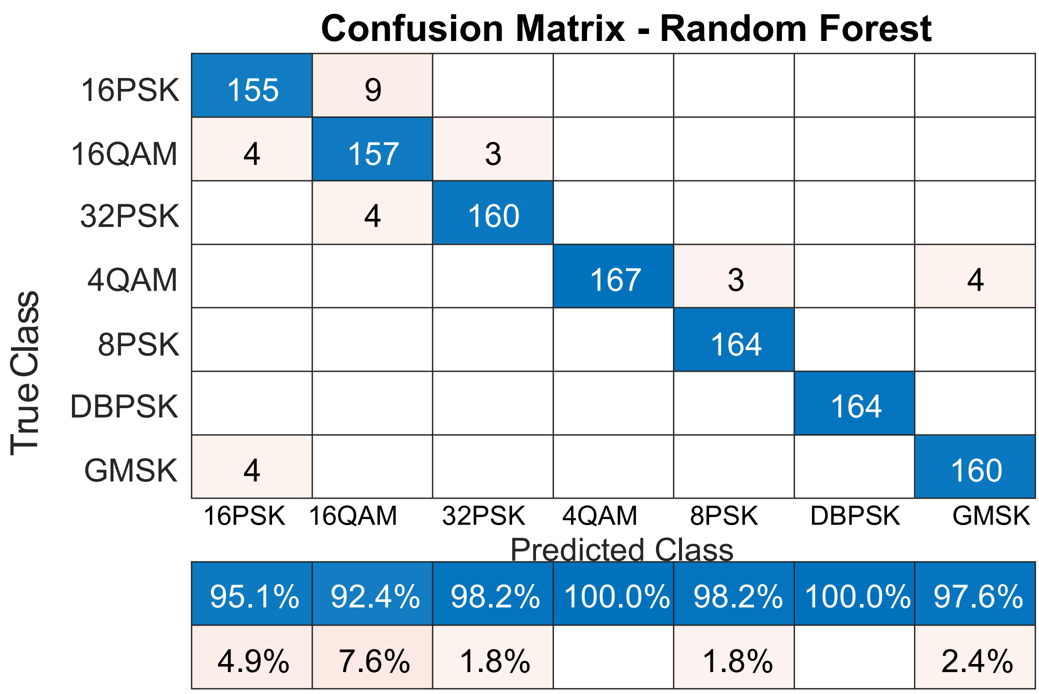

Figure III.36: Confusion Matrix for Random Forest Classification of Modulation.

In general, the low off-diagonal error rates, ranging from 1.8% to 7.6%, underscore the model's high discriminative capability and robust generalization. This performance is likely attributed to effective feature extraction and the inherent ensemble learning strengths of the Random Forest algorithm. The confusion matrix thus reinforces the model's suitability for reliable and precise modulation classification in practical communication systems.

III.4.2.3 ROC Curves by Class

The ROC curve presented in Figure III.6 demonstrates a substantial improvement in the model's classification performance across all modulation schemes. Each curve corresponds to a specific modulation type, with the AUC providing a quantitative assessment of the model's ability to distinguish between classes. Notably, modulation formats such as 32PSK, 4QAM, 8PSK, DBPSK, and GMSK achieve an AUC of 1.00, indicating perfect classification performance. Additionally, 16PSK and 16QAM register near-perfect AUC values of 0.99, signifying highly reliable discrimination with minimal overlap between predicted and actual classes. The ROC curves for all classes are tightly clustered near the top-left corner of the plot, reflecting high TP rates coupled with low FP rates across the board.

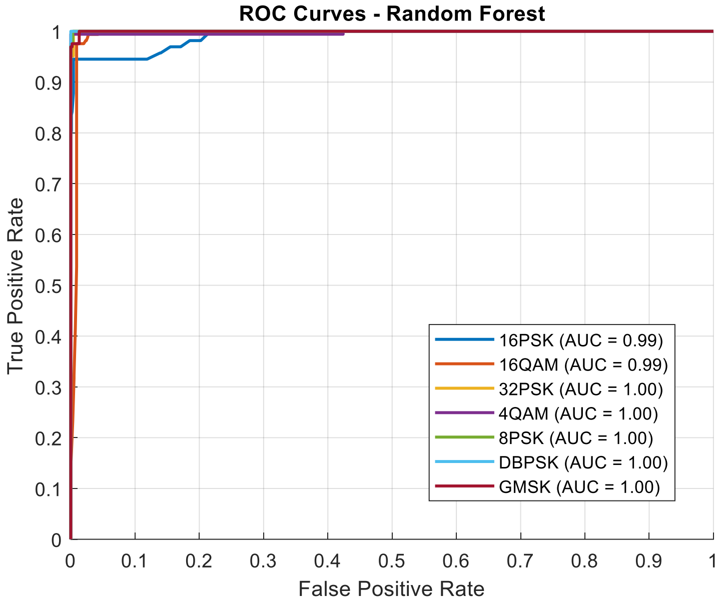

Figure III.37 : ROC Curves for Random Forest Classification of Modulation.

These results highlight the model's excellent discriminative power, which can be attributed to factors such as optimal hyperparameter tuning, effective feature extraction, and a well-structured dataset with strong inter-class separability. Collectively, the high AUC values reinforce the model's robustness and its suitability for accurate and reliable modulation classification in real-world communication systems.

III.4.3 XGBOOST Results

The following sections provide a comprehensive evaluation of the XGBoost model in the context of digital modulation classification, showcasing its remarkable capability to accurately distinguish between a wide range of modulation schemes. The analysis emphasizes the model's high classification accuracy, robust generalization performance, and consistency across diverse signal types. These findings not only validate the effectiveness of XGBoost for this task but also demonstrate its superiority over other ML approaches explored in this study. As such, XGBoost emerges as a highly reliable and efficient solution for complex signal processing challenges, particularly in wireless communication systems where precise modulation recognition is critical.

III.4.3.1 XGBOOST Classification Metrics per Class

The classification results illustrated in Figure III.7 demonstrate a further improvement in model performance, achieving a global accuracy of 98.45% and a global F1-score of 0.98. All modulation formats 16PSK, 16QAM, 32PSK, 4QAM, 8PSK, DBPSK, and GMSK exhibit consistently high values across precision, recall, and F1-score, underscoring the model's strong generalization capabilities. In particular, the model performs exceptionally well on 32PSK, 4QAM, DBPSK, and GMSK, with precision and recall scores nearly identical and approaching unity. This indicates both a high TPR and a low misclassification rate, suggesting that the distinguishing features of these modulation schemes are effectively captured by the model. Similarly, 8PSK maintains excellent performance, comparable to the best-performing categories.

While the overall performance remains outstanding, 16PSK and 16QAM show slightly lower recall values relative to their precision. This suggests that although the model is highly accurate when predicting these classes (i.e., few FP), it misses a small number of true instances (i.e., moderate FN). Despite this, their high F1-scores confirm that the model achieves a well-balanced trade-off between sensitivity and specificity.

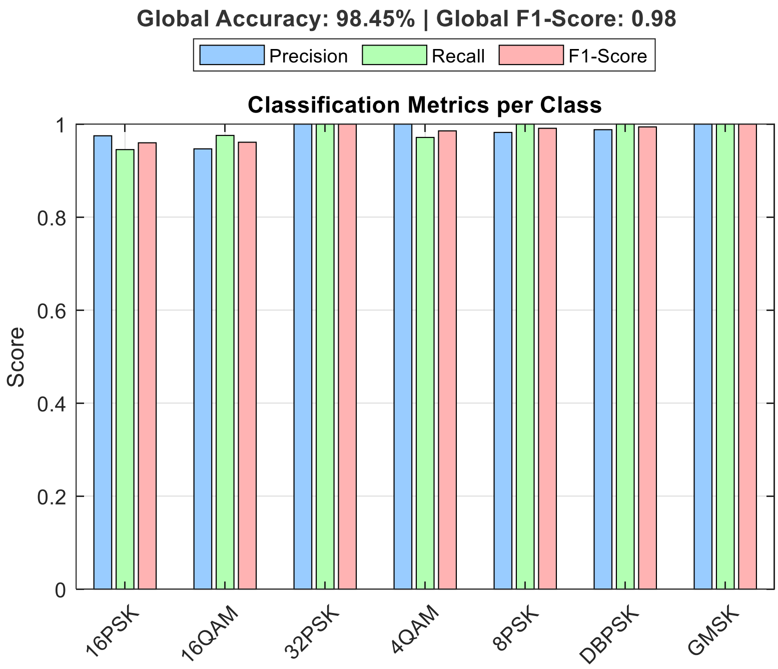

Figure III.38: Precision, Recall, and F1-Score per Class for XGBOOST Model.

III.4.3.2 XGBOOST Confusion Matrix

The confusion matrix presented in Figure III.8 illustrates the performance of a highly accurate multi-class classification model tasked with identifying seven digital modulation formats: 16PSK, 16QAM, 32PSK, 4QAM, 8PSK, DBPSK, and GMSK. The matrix reveals exceptional classification results, with per-class accuracy exceeding 94% across all modulation types. The model achieved perfect classification (100%) for 32PSK, 4QAM, and GMSK, demonstrating its ability to distinguish these formats with complete reliability. Similarly high accuracies were recorded for DBPSK (98.8%), 8PSK (98.2%), 16PSK (97.5%), and 16QAM (94.7%). Correct predictions are concentrated along the diagonal entries of the matrix, indicating a strong alignment between predicted and true labels. Misclassifications are minimal and typically involve closely related modulation types, suggesting slight overlaps in signal characteristics.

For example, 16PSK was misclassified as 16QAM in 9 instances, and conversely, 16QAM was predicted as 16PSK in 4 cases likely due to similarities in constellation structures or phase resolution under noisy conditions. A few misclassifications also occurred within 4QAM, with 3 instances labeled as 8PSK and 2 as DBPSK, potentially reflecting shared phase or amplitude properties in degraded signal environments.

Despite these minor errors, with misclassification rates ranging from only 1.2% to 5.3%, the model exhibits strong robustness and class separability. These results not only validate the effectiveness of the XGBoost classifier but also demonstrate the high discriminative quality of the extracted features, making this approach highly suitable for practical applications in modulation recognition.

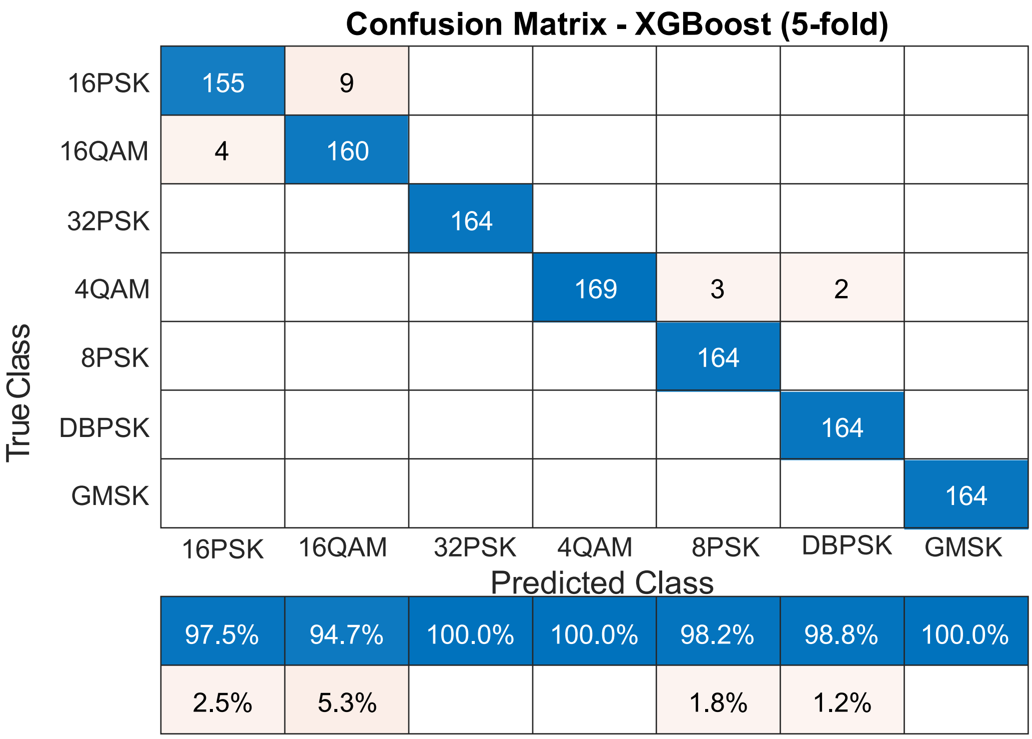

Figure III.39: Confusion Matrix for XGBOOST Classification of Modulation.

III.4.3.3 XGBOOST ROC Curves by Class

Figure III.9 presents ROC curves for the XGBoost model's classification of various modulation schemes, showcasing exceptional performance across all classes. The AUC values further validate the model's discriminative power, with modulation types such as 16PSK, 16QAM, 32PSK, BPSK, DBPSK, and GMSK achieving a perfect AUC of 1.00, indicating flawless separability between classes. The 4QAM class demonstrates an outstanding AUC of 0.99, reflecting only minimal overlap between positive and negative classifications, underscoring the model's ability to distinguish even more subtle differences.

These results reinforce the robustness of the XGBoost model, consistent with the strong performance observed in earlier metrics. The consistently high AUC scores, including for classes with historically challenging similarities, such as 16PSK and 16QAM, highlight the model's reliable discriminative capabilities. This further solidifies XGBoost's superiority in modulation recognition tasks, making it particularly well-suited for high-precision applications in wireless communications and signal processing.

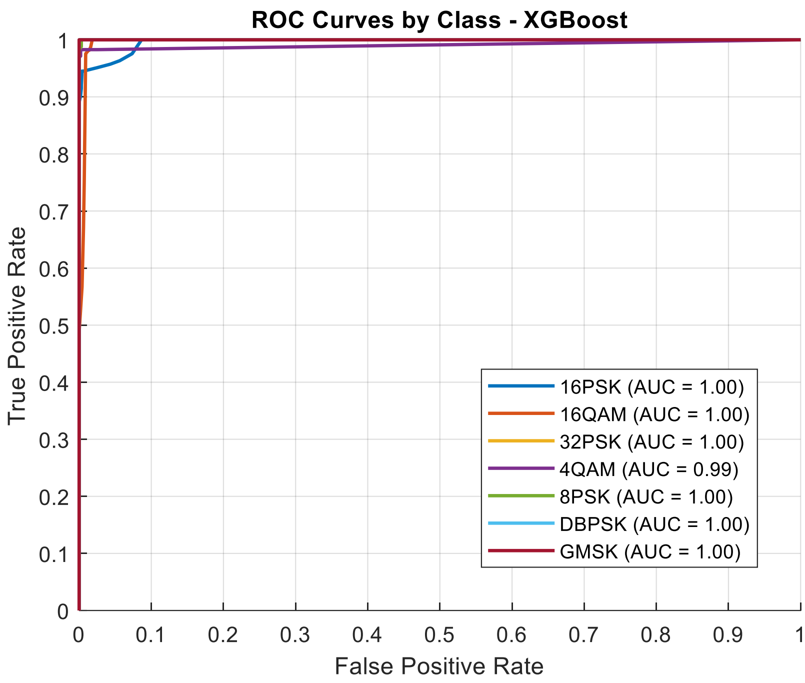

Figure III.40: ROC Curves for XGBOOST Classification of Modulation.

III.4 Comparison of the results of the three models

The evaluation of the CNN, Random Forest, and XGBoost models for digital modulation classification (Table III.1) reveals distinct performance disparities. The CNN model, while effective, demonstrated the weakest results, with an accuracy of 70.64% and an F1-score of 0.69. It faced significant challenges in differentiating between similar modulation schemes, such as 16PSK and 16QAM, where the accuracy dropped to 60.2%. Additionally, the CNN model exhibited inconsistent class separability, as reflected in its AUC values ranging from 0.89 to 0.99. In contrast, the Random Forest model showed a marked improvement, achieving 97.32% accuracy and an F1-score of 0.97. It classified 4QAM, DBPSK, and GMSK perfectly, though it still encountered minor misclassifications between 16PSK and 16QAM, with error rates ranging from 4.9% to 7.6%.

The XGBoost model emerged as the superior performer, with an impressive 98.45% accuracy and an F1-score of 0.98. It achieved flawless classification for four modulation types (32PSK, 4QAM, DBPSK, GMSK) and demonstrated near-perfect AUC scores ranging from 0.99 to 1.00 across all classes. While both ensemble methods (Random Forest and XGBoost) outperformed CNN, XGBoost's gradient boosting framework provided slightly better discrimination, especially for more challenging cases, reinforcing its position as the optimal model for this task.

Table III.1: Comparison table of the three models

Metric	CNN	Random Forest	XGBoost
Global Accuracy	70.64%	97.32%	98.45% **(Best)**
Global F1-Score	0.69	0.97	0.98 **(Best)**
Best Class Accuracy	DBPSK,4QAM,32PSK	DBPSK, GMSK, 8PSK (100%)	DBPSK, GMSK, 8PSK, 32PSK (100%)
Most Confused Class	16PSK misclassified as 4QAM/16QAM	16PSK confused with 16QAM	16PSK ↔ 16QAM (minor errors)
AUC Scores	Mostly high, 0.89–0.99	0.99–1.00 (excellent)	All AUCs = 1.00 or 0.99

The clear performance hierarchy positions XGBoost as the top choice, followed by Random Forest, with CNN significantly trailing behind. This ranking highlights the advantages of ensemble methods in structured signal classification tasks, where their ability to combine multiple weak learners and effectively handle feature interactions proves particularly valuable. The results suggest that while deep learning approaches like CNN hold potential, traditional machine learning methods, such as Random Forest and XGBoost, currently offer more reliable solutions, especially when dealing with spectrally similar signal types.

III.5 Conclusion

The experimental results demonstrate a clear performance hierarchy among the evaluated models. While the CNN model achieves moderate accuracy (70.64%), it struggles with spectrally similar modulation types, such as 16PSK and 16QAM. In contrast, ensemble methods—Random Forest (97.32% accuracy) and XGBoost (98.45% accuracy)—exhibit superior classification capabilities, with XGBoost emerging as the top performer due to its near-perfect AUC scores and minimal misclassifications. The robustness of XGBoost in handling complex signal features underscores its suitability for real-world AMC applications. These findings suggest that, despite the potential of deep learning, traditional ML techniques like XGBoost currently offer more reliable solutions for modulation classification tasks. Future work could explore hybrid architectures or enhanced feature extraction methods to further bridge the performance gap between DL and ensemble approaches.

General Conclusion:

This thesis has presented a comparative study of three advanced models Random Forest, XGBoost, and CNN for the task of AMC using the RadioModRec-1 dataset. The research was driven by the growing need for reliable and adaptive techniques capable of handling the complexity of modern wireless communication systems.

Contrary to the common expectation that DL models outperform classical approaches, the experimental results in this study revealed that Random Forest and XGBoost achieved higher accuracy and more robust performance than CNN in the tested scenarios. These ensemble-based methods demonstrated strong classification capabilities, particularly when combined with carefully engineered features such as normalized I/Q samples and constellation-based metrics. Their efficiency, interpretability, and resilience to noise and signal variation contributed to their superior performance [80].

CNNs, while powerful in many contexts, showed lower classification accuracy in this study, potentially due to factors such as limited dataset size, model complexity, or the specific signal representations used. Despite their ability to learn features directly from raw data, CNNs may require more extensive tuning or data augmentation to match or surpass the performance of structured ML models in certain signal classification tasks.

These findings suggest that traditional ML models remain highly competitive for modulation recognition, especially when appropriate feature engineering is applied. They offer a practical and effective solution for many real-world applications, particularly in environments with limited computational resources or when rapid deployment is needed.

Future work could focus on enhancing CNN performance through optimized architectures, larger and more diverse training datasets, or hybrid approaches that combine the strengths of both ML and DL. Additionally, exploring model generalization across different communication standards and environments would further validate their applicability in real-world systems. As the field of wireless communications continues to evolve, intelligent and adaptive modulation classification will play a crucial role in ensuring efficient, secure, and high-quality connectivity across a range of applications.

References

[1] Savo G. Glisic, Advanced Wireless Communications and Internet: Future Evolving Technologies, Hoboken, New Jersey: John Wiley & Sons, 2011, pp. 1-960.

[2] F.Zhang, C.Luo, J.Xu, Y.Luo and F.Zheng, "Deep learning based automatic modulation recognition: Models, datasets, and challenges," *Digital Signal Processing,* pp. 1_19, 2022.

[3] F.Rehman, Q.Abbas and M.Karam Shehzad, "DL-AMC: Deep Learning for Automatic Modulation Classification," pp. 1_5, 2025.

[4] Gerrard and G.Macangus , Offshore Electrical Engineering Manual, Second Edition ed., Houston, Texas: Gulf Professional Publishing, 2018, pp. 1_541.

[5] K.Wesolowski and H.Wesolowski , Mobile Communication Systems, USA: John Wiley & Sons, Inc., 2001, pp. 1_352.

[6] R.Fagen and A.Hall , Systems Research for Behavioral Science: Definition of System, Routledge, 1968.

[7] S. R. a. A. Saunders, Antennas and propagation for wireless communication systems, Hoboken, NJ: Wiley, 2007, pp. 1_524.

[8] Agrawal and P.Govind , Fiber-Optic Communication Systems, USA: John Wiley & Sons, 2012, pp. 600_627.

[9] Wang,Z.Mu,X.Liu and Yuanwei, "Bidirectional Integrated Sensing and Communication: Full-Duplex or Half-Duplex?," *IEEE Transactions on Wireless Communications,* pp. 8184_8199, 2024.

[10] R. Yarlagadda, Analog and Digital Signals and Systems, USA: Springer New York, 2009, pp. 1_59.

[11] G.Maral, M.Bousquet and S.Zhili, Satellite Communications Systems: Systems, Techniques and Technology, USA: John Wiley & Sons, 2020, pp. 790_804.

[12] E.A.Lee and D.G.Messerschmitt, Digital Communication, USA: Springer Science & Business Media, 2012, pp. 700_727.

[13] F.Tobagi, "Multiaccess Protocols in Packet Communication Systems," *IEEE Transactions on Communications,* pp. 468_488, 1980.

[14] A.K.Nandi and Z.Zhu, Automatic Modulation Classification: Principles, Algorithms and Applications, USA: John Wiley & Sons, 2015, pp. 156_204.

[15] J. D. Gibson, Introduction to Communication Systems, USA: Springer Cham, 2023, pp. 69_90.

[16] S.Faruque, Amplitude Modulation (AM), USA: Springer, Cham, 2021, pp. 147_163.

[17] M.Steer, "Microwave and RF Design I - Radio Systems," Libre Texts, Carolina , 2019.

[18] XB Bian and AD Bandrauk, "Probing Nuclear Motion by Frequency Modulation of Molecular High-Order Harmonic Generation," *Physical Review Letters,* pp. 193_901, 2014.

[19] D.R.Stephens, Phase-Locked Loops for Wireless Communications: Digital and Analog Implementations, USA: Springer, 2012, pp. 1_434 .

[20] J.B. Anderson, T.Aulin and C.Sundberg, Digital Phase Modulation, USA: Springer Science & Business Media, 2013, pp. 1_509.

[21] S.Ghavami,R.Adve and F.Lahouti, "Information Rates of ASK-Based Molecular Communication in Fluid Media," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications,* pp. 277_291, 2015.

[22] F.A.Rakhmadi, Mitrayana and H.Shalihah, "Design of green laser modulation system based on Arduino uno microcontroller," *Conference Proceedings,* 2018.

[23] C. Erdoğan,I. Myderrizi and S. Minaei, "FPGA Implementation of BASK-BFSK-BPSK Digital Modulators [Testing Ourselves]," *IEEE Antennas and Propagation Magazine,* pp. 262_269, 2012.

[24] L.Vangelista, "Frequency Shift Chirp Modulation: The LoRa Modulation," *IEEE Signal Processing Letters,* pp. 1818_1821, 2017.

[25] K. Sushmaja and F.Noorbasha, "Implementation of Binary Shift Keying Techniques," *International Journal of Engineering Trends and Technology,* 2013.

[26] P.Krishan, "Performance Analysis of Hybrid RF/FSO System Using BPSK-SIM and DPSK-SIM Over Gamma-Gamma Turbulence Channel With Pointing Errors for Smart City Applications," *IEEE Access,* pp. 75025_75032, 2018.

[27] Md.S.Faruk and K.Kikuchi, "Compensation for In-Phase/Quadrature Imbalance in Coherent-Receiver Front End for Optical Quadrature Amplitude Modulation," *IEEE Photonics Journal,* pp. 7800110_7800110, 2013.

[28] I.Fady and El-Nahal, "Coherent 16 Quadrature Amplitude Modulation (16QAM) Optical Communication Systems," *Photonics Letters of Poland,* pp. 57_59, 2018.

[29] Z.Liu,J.Kakande,B.Kelly,J.O'Carroll,R.Phelan,D.J. Richardson and K.Slavík, "Modulator-free quadrature amplitude modulation signal synthesis," *Nature Communications,* 2014.

[30] H.Dahrouj, R.Alghamdi, H.Alwazani, S.Bahanshal, A.Alameer Ahmad and A.Faisal, "An Overview of Machine Learning-Based Techniques for Solving Optimization Problems in Communications and Signal Processing," *IEEE,* vol. 9, pp. 74908 _ 74938, 2021.

[31] K.Sun, A.Roy and J.M.Tobin, "Artificial intelligence and machine learning: Definition of terms and current concepts in critical care research," *Journal of Critical Care,* vol. 82, 2024.

[32] O.Ugochukwu Ikechukwu, O.Ogugua Chimezie, A.Adebunmi Okechukwu and A.Temitayo Oluwaseun , "Machine learning in cybersecurity: A review of threat detection and defense mechanisms," *World Journal of Advanced Research and Reviews,* vol. 21, no. 1, pp. 2286_2295, 2024.

[33] Yuji Roh, Geon Heo, Steven Euijong Whang, "A Survey on Data Collection for Machine Learning," *IEEE,* pp. 1_20, 2019.

[34] C.Vladimiro and G.Zelaya, "Towards Explaining the Effects of Data Preprocessing on Machine Learning," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, Macao, China, 2019.

[35] D.Theng and Kishor K. Bhoyar, "Feature selection techniques for machine learning: a survey of more than two decades of research," *Knowledge and Information Systems,* vol. 66, pp. 1575–1637, 2023.

[36] C.Liu, P.Balasubramanian, J.An and F.Li, "Machine learning prediction of ammonia nitrogen adsorption on biochar with model evaluation and optimization," *npj clean water with King Fahd University of Petroleum &Minerals,* pp. 1-12, 2025.

[37] U.Bhatt, A.Xiang, S.Sharma, A.Weller, A.Taly, Y.Jia, J.Ghosh, R.Puri, José M. F. Moura and P.Eckersley, "Explainable machine learning in deployment," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, New York, NY, USA, 2020.

[38] A.E. Márquez-Chamorro, I.A.Nepomuceno-Chamorro, M.Resinas and A.Ruiz-Cortés, "Updating Prediction Models for Predictive Process Monitoring," in *Advanced Information Systems Engineering 34th International Conference, CAiSE 2022*, Leuven, Belgium, 2022.

[39] V.Nasteski, "An overview of the supervised machine learning," researchgate, Bitola, Macedonia, 2017.

[40] C.M. Eckhardt, S.J. Madjarova, R.J. Williams, M.Ollivier, J.Karlsson, A.Pareek and B.U. Nwachukwu, "Unsupervised machine learning methods and emerging applications in healthcare," *Knee Surgery, Sports Traumatology, Arthroscopy,* vol. 31, pp. 376–381, 2022.

[41] Jesper E. van Engelen and Holger H. Hoos, "A survey on semi-supervised learning," *Springer Nature,* vol. 109, pp. 373–440, 2019.

[42] S.Mousavia, M.Schukata and E.Howley, "Deep Reinforcement Learning: An Overview," in *Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016*, Cham, 2017.

[43] B.Jdid, K.Hassan, I.Dayoub, W.Lim, M.Mokayef, "Machine Learning Based Automatic Modulation Recognition for Wireless Communications: A Comprehensive Survey," *IEEE,* vol. 9, pp. 57851 - 57873, 2021.

[44] N.Ravikumar,A.Zakeri,Y.Xia and A.F. Frangi, "Chapter 16 - Deep learning fundamentals," in *Medical Image Analysis*, Academic Press, 2024, pp. 415-450.

[45] Y.Wu and J.Feng, "Development and Application of Artificial Neural Network," *Wireless Personal Communications,* vol. 102, pp. 1645–1656, 2017.

[46] K.Ahamed and S.Akthar, "A Study on Neural Network Architectures," *International Institute for Science, Technology and Education,* pp. 1-7, 2016.

[47] V.Sze, Y.Chen, T.Yang and J.Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," *Proceedings of the IEEE,* vol. 105, pp. 2295-2329, 2017.

[48] X.Zhou, W.Zhang, Z.Chen, S.Diao and T.Zhang, "Efficient Neural Network Training via Forward and Backward Propagation Sparsification," in *Advances in Neural Information Processing Systems*, China, 2021.

[49] LI, M.DOROSLOVAČKI and M.H. LOEW, "Approximating the Gradient of Cross-Entropy Loss Function," *IEEE Access,* vol. 8, pp. 111626-111635, 2020.

[50] F.Mehmood, S.Ahmad and T.Whangbo, "An Efficient Optimization Technique for Training Deep Neural Networks," *Mathematics,* pp. 1000-1360, 2023.

[51] L. N. Smith, "Cyclical Learning Rates for Training Neural Networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Washington, 2017.

[52] L.Guerdan, P.Sun, C.Rowland, L.Harrison, Z.Tang, N.Wergeles and Y.Shang, "Deep Learning vs. Classical Machine Learning: A Comparison of Methods for Fluid Intelligence Prediction," in *Adolescent Brain Cognitive Development Neurocognitive Prediction (ABCD-NP 2019)*, U.S.A, 2019.

[53] K.O'Shea and R.Nash, "An Introduction to Convolutional Neural Networks," arXiv, uk, 2015.

[54] N.Ketkar and J.Moolayil , "Convolutional Neural Networks," in *Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch*, Apress, 2021, pp. 197-242.

[55] JA.AYENI, "Convolutional Neural Network (CNN): The architecture," *Journal of Physical Science,* vol. 4, pp. 42-50, 2022.

[56] C.Kong and S.Lucey, "Take it in your stride: Do we need striding in CNNs?," arXiv, USA, 2017.

[57] L.Bai, Y.Lyu and X.Huang, "A Unified Hardware Architecture for Convolutions and Deconvolutions in CNN," arXiv, USA, 2020.

[58] S.Shadman, A.Nazib, A.J.Kabir and A.Hridon, "An Overview of Convolutional Neural Network: Its Architecture and Applications," Preprints, Bangladesh, 2017.

[59] A.Ziaee and E.Çano, "Batch Layer Normalization A new normalization layer for CNNs and RNNs," in *Proceedings of the 6th International Conference on Advances in Artificial Intelligence*, New York, NY, USA, 2023.

[60] S.Cai, Y.Shu, W.Wang, G.Chen, B.Ooi and M.Zhang, "Effective and Efficient Dropout for Deep Convolutional Neural Networks," arXiv, China, 2020.

[61] S.Tavasoli, "The Machine Learning Algorithms List: Types and Use Cases," *AI & Machine Learning,* pp. 1-2, 2025.

[62] J.Ali, R.Khan, N.Ahmad and I.Maqsood, "Random Forests and Decision Trees," *International Journal of Computer Science,* pp. 272-277, 2012.

[63] Y.Liu, Y.Wang and J.Zhang , "New Machine Learning Algorithm: Random Forest," in *Information Computing and Applications*, Berlin, Heidelberg, 2012.

[64] U.Senturk, K.Polat, I.Yucedag and F.Alenezi, "Chapter 4 - Arrhythmia diagnosis from ECG signal pulses with one-dimensional convolutional neural networks," in *Diagnostic Biomedical Signal and Image Processing Applications with Deep Learning Methods*, Academic Press, 2023, pp. 83-101.

[65] S.Suthaharan, "Random Forest Learning," in *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*, Springer US, 2016, pp. 273-288.

[66] C.Vens and F.Costa, "Random Forest Based Feature Induction," in *2011 IEEE 11th International Conference on Data Mining*, Vancouver, BC, Canada, 2011.

[67] L.Breiman, "Random Forests," *Machine Learning,* pp. 5–32, 2001.

[68] H.Belyadi and A.Haghighat, "Chapter 5 - Supervised learning," in *Machine Learning Guide for Oil and Gas Using Python*, Gulf Professional Publishing, 2021, pp. 169-295.

[69] T.Chen and C.Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2016.

[70] A.Anghel, N.Papandreou and T.Parnell, A.De Palma and H.Pozidis, "Benchmarking and Optimization of Gradient Boosting Decision Tree Algorithms," *arXiv,* pp. 1-7, 2019.

[71] C.Bentéjac, A.Csorgo and G.Martınez-Munoz, "A Comparative Analysis of XGBoost," *researchgate,* pp. 1-20, 2019.

[72] C. D. Sutton, "11 - Classification and Regression Trees, Bagging, and Boosting," in *Handbook of Statistics*, Elsevier, 2005, pp. 303-329.

[73] G.Naidu, T.Zuva and E.M.Sibanda, "A Review of Evaluation Metrics in Machine Learning Algorithms," in *Artificial Intelligence Application in Networks and Systems*, Springer, Cham, 2023, pp. 15–25.

[74] T. a. R. M.Heydarian, "MLCM: Multi-Label Confusion Matrix," *IEEE Access,* pp. 19083 - 19095, 2022.

[75] A.S.Ferreira, A.W.Visser, B.R.MacKenzie and M.R.Payne, "Accuracy and precision in the calculation of phenology metrics," *Journal of Geophysical Research: Oceans,* pp. 8195-9034, 2014.

[76] G.M.Foody, " Challenges in the real world use of classification accuracy metrics: From recall andprecision to the Matthews correlation coefficient," *Journal Pone,* pp. 1-27, 2023.

[77] Z.H.Hoo,J.Candlish and D.Teare, "What is an ROC curve?," *Emergency Medicine Journal,* pp. 357-359, 2017.

[78] J.R.Turner, "Area Under the Curve (AUC)," *Springer Nature,* pp. 1-146, 2020.

[79] Zhang D, Ding W, Zhang B, Xie C, Li H, Liu C and al. , "Automatic modulation classification based on deep learning for unmanned aerial vehicles.," *Sensors Signal Processing and Visual Computing,* pp. 1-15, 2018.

[80] N.Mohapatra, K.Shreya and A.Chinmay , "Optimization of the Random Forest Algorithm," in *Advances in Data Science and Management*, Singapore, 2020.