DEEP LEARNING BASED AUTOMATIC MODULATION CLASSIFICATION FOR SUB-CARRIERS OF OFDM SIGNALS

by

Gökhan Tosun

B.Sc., Electrical & Electronics Engineering, İhsan Doğramacı Bilkent University, 2019

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical & Electronics Engineering
Boğaziçi University
2023

ABSTRACT

DEEP LEARNING BASED AUTOMATIC MODULATION CLASSIFICATION FOR SUB-CARRIERS OF OFDM SIGNALS

Automatic modulation classification (AMC) is automatically identifying and classifying the modulation schemes employed in digital communications. By accurately identifying the modulation scheme, AMC enables communication systems to adapt their parameters, optimizing efficiency, spectral utilization, and overall performance. The majority of the literature on AMC focuses on the single-carrier communications systems. This thesis addresses the gap between the AMC and multi-carrier communications systems. Two architectures are proposed. Both employ a filter bank-convolutional neural network (CNN) complex. The first architecture uses raw features and a maximum operation to perform classification, whereas the second architecture learns feature patterns by employing a fully connected neural network (FNN). It is observed that the raw features are not sufficiently informative for theoretical and practical purposes. It is further observed that putting together the raw features and allowing the transformations on the combinations of the raw features, effectively forming a decision context, improves the performance significantly. The performances of both architectures are analyzed through the accuracy metric and confusion matrices. Finally, the thesis is concluded by summarizing the experiments, results, and implications and mentioning the possible future work.

ÖZET

OFDM SİNYALLERİNİN ALTTAŞIYICILARI İÇİN DERİN ÖĞRENME TEMELLİ OTOMATİK MODÜLASYON SINIFLANDIRMA

Otomatik modülasyon sınıflandırması (AMC), dijital haberleşmede kullanılan modülasyon tekniklerini otomatik olarak tanımlar ve sınıflandırır. AMC, modülasyon tekniğini doğru bir şekilde tanımlayarak iletişim sistemlerinin parametrelerini uyarlamasına, verimliliği, tayf kullanımını ve genel performansı optimize etmesini sağlar. AMC ile ilgili literatürün büyük bir kısmı tek taşıyıcılı iletişim sistemlerine odaklanmaktadır. Bu tez, AMC ile çok taşıyıcılı iletişim sistemleri arasındaki boşluğu ele almaktadır. Önerilen iki mimari vardır. Her ikisi de bir filtre bankası ve bir evrişimli sinir ağından (CNN) oluşan bir bileşik yapı kullanır. İlk mimari, sınıflandırmayı gerçekleştirmek için işlenmemiş özellikleri ve maksimum işlemi kullanırken, ikinci mimari, ek olarak bir sinir ağı (FNN) kullanarak özelliklerde ortak olarak görülen kalıpları öğrenir. İşlenmemiş özelliklerin teorik ve pratik amaçlar açısından yeterince bilgilendirici olmadığı görülmektedir. Ek olarak, işlenmemiş özellikleri bir araya getirmenin ve işlenmemiş özelliklerin kombinasyonları üzerinde dönüşümlere izin vererek etkili bir karar bağlamı oluşturmanın performansı önemli ölçüde artırdığı gözlemlenmiştir. Her iki mimarinin de performansı doğruluk metriği ve hata matrisleri aracılığıyla analiz edilmiştir. Son olarak deneylerin, sonuçların ve çıkarımların özetlenmesi ve gelecekte yapılması muhtemel çalışmalara değinilerek tez sonlandırılmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A	Instantaneous amplitude
A_n	Normalized instantaneous amplitude
A_{nc}	Normalized-centered instantaneous amplitude
$\mathbf{b}_i$	Bitstream vector within the i^{th} training sample
B	Length of a bitstream
BW	Bandwidth of a communications channel
BW_c	Coherence bandwidth
BW_i	Bandwidth of the i^{th} subchannel
C_i	The i^{th} cumulant
f	Instantaneous frequency
f_n	Normalized instantaneous frequency
f_{nc}	Normalized-centered instantaneous frequency
f_X	Probability density function of the random variable X
$\mathcal{F}_N^{-1}(\cdot)$	N-point IFFT operation
g	Pulse-shaping signal
G	Margin of the support vector machine
h	Arbitrary function
$I(\cdot,\cdot)$	Input of convolution operation
$I(m_i)$	Set of symbols in the i^{th} modulation scheme
k	Bits per symbol
K	Number of received symbols
$K(\cdot,\cdot)$	Kernel function
l	Log-likelihood function
L	Lagrangian function *or* frame length
$\mathbb{L}$	Likelihood function
m_i	The i^{th} modulation scheme
M	Number of modulation schemes
$M(i)$	Number of symbols in the i^{th} modulation scheme

$\mathbb{M}_t$	Modulation operation with scheme t
N	FFT length
$\mathbb{P}$	Probability measure
$\mathbf{r}$	Vector of received symbols
r_i	The i^{th} received symbol
r_{Ii}	Real part of the i^{th} received symbol
r_{Qi}	Imaginary part of the i^{th} received symbol
R	Autocorrelation function *or* bitrate
R_i	Bitrate of the i^{th} subchannel
s	Signal
$\mathbf{s}$	Symbol sequence
$\mathbf{s}^P$	Parallelised symbol sequence
S	Power of the signal
$S(\cdot, \cdot)$	Output of convolution operation
T_s	Sampling period
U	Number of subchannels in a multi-carrier
w	Weight of an MLP input
$\mathbf{w}$	Normal vector of a hyperplane
$X[i]$	The i^{th} frequency component of the DFT of $x[n]$
$\mathbf{x}^+$	Positive samples within the training data
$\mathbf{x}^-$	Negative samples within the training data
$\mathbf{x}_i$	The i^{th} training sample
$\mathbf{x}_i^{TX}$	The i^{th} transmitted frame
$x[i]$	The i^{th} item in a discrete-time sequence
y_i	The i^{th} training label
α	Lagrangian multiplier *or* wavelet function parameter
γ_{max}	Maximum normalized-centered instantaneous amplitude
δ	Dirac delta function
δ_i	The i^{th} symbol in a modulation scheme
η	Noise process

θ	Value of the Θ random variable
Θ	Random variable denoting parameters
μ	Mean of a probability distribution
ξ	MLP activation function
σ	Standard deviation of a probability distribution
τ	Wavelet function parameter
ϕ	Instantaneous phase *or* feature transformation
ϕ_{NL}	Non-linear component of the instantaneous phase
$\psi_{a,\tau}$	Wavelet function with parameters a and τ
Ψ	Moment generating function
ω_c	Angular frequency of the carrier signal

LIST OF ACRONYMS/ABBREVIATIONS

AD	Anderson-Darling
ALRT	Average Likelihood Ratio Test
AMC	Automatic Modulation Classification
AWGN	Additive White Gaussian Noise
BPSK	Binary Phase-Shift-Keying
CDF	Cumulative Distribution Function
CFO	Carrier Frequency Offset
CNN	Convolutional Neural Network
CSI	Channel State Information
CWT	Continuous Wavelet Transform
CvM	Cramér–von Mises
DFT	Discrete Fourier Transform
EA	Electronic Attack
EM	Expectation-Maximization
EP	Electronic Protect
ES	Electronic Support
FFT	Fast Fourier Transform
FN	False Negative
FNN	Feed-forward Neural Network
FP	False Positive
GLRT	Generalized Likelihood Ratio Test
GoF	Goodness of Fit
HLRT	Hybrid Likelihood Ratio Test
IDFT	Inverse Discrete Fourier Transform
IFFT	Inverse Fast Fourier Transform
ICI	Intercarrier Interference
IQ	In-phase Quadrature
KNN	k-Nearest Neighbor

KS	Kolmogorov-Smirnov
MFCC	Mel Frequency Cepstral Coeffieicents
MGF	Moment Generating Function
ML	Maximum Likelihood
MLP	Multi-layer Perceptron
OFDM	Orthogonal Frequency-Division Multiplexing
PSK	Phase-Shift-Keying
QAM	Quadrature-Amplitude Modulation
QPSK	Quadrature Phase-Shift-Keying
ReLU	Rectified Linear Unit
RF	Radio Frequency
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristics
SNR	Signal-to-Noise Ratio
STFT	Short-time Fourier Transform
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

1. INTRODUCTION

Wireless communications have become an integral element of modern society, transforming how people interact, exchange information, and conduct business. As the world increasingly relies on seamless connectivity, the significance of wireless communication systems cannot be overstated. A crucial element of modern wireless communication systems is automatic modulation classification (AMC).

AMC is automatically identifying and classifying the modulation schemes employed in wireless transmissions. This capability holds massive importance in today's dynamic communication landscape, where many devices and applications coexist within the limited spectrum resources available. By accurately identifying the modulation scheme, AMC enables communication systems to adapt their parameters, optimizing efficiency, spectral utilization, and overall performance.

1.1. Civil Applications

There are various civil applications of AMC. Instead of exploiting the target communications, civil applications focus on utility and performance.

An example is spectrum monitoring. Given that the electromagnetic spectrum is a scarce resource, various authorities control and allocate the parts of the spectrum to different service providers or users. In the case where there is an unwanted activity within an allocated frequency band, the responsible authority can use AMC to detect the modulation type of the signal in order to demodulate and determine the source of the unwanted activity.

In some communications systems, the transmitter utilizes some adaptive modulation scheme. That is, the transmitter adapts the communications parameters based on the quality of the radio link. In such a scenario, a parameter of interest is the modu-

lation type of the signal. The transmitter might use more robust modulation schemes, such as binary phase-shift-keying (BPSK) and quadrature phase-shift-keying (QPSK) when the link quality is relatively lower and more performant modulation schemes, such as 16 or 64 quadrature amplitude modulation (QAM), when the link quality is relatively higher. In such communication systems, the receiver can utilize AMC to decide on the modulation type of the received signal if such information is unavailable beforehand.

1.2. Military Applications

Modern electronic warfare consists of three major components. Those components are described in [1] as electronic support (ES), electronic attack (EA), and electronic protect (EP). AMC technology is an essential tool within each subset.

ES aims to extract specific information from the collected signals. Some examples of the target features are frequency, modulation type, and bitrate. Automatic modulation classification can be used in electronic support context to extract modulation information of the intercepted signal. Such information is called combat information and, with further processing, can be used for intelligence generation. Electronic support helps maintain communications superiority on the battlefield.

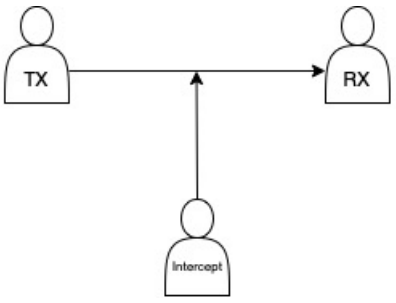

Figure 1.1: Electronic warfare diagram.

EA tries to prevent the adversary from effectively using their communication systems using various techniques. Some examples are jamming and radar deception.

Jamming requires knowledge of modulation type; hence automatic modulation classification can be utilized for electronic attack purposes.

EP seeks to protect friendly communications from adversary electronic attack actions. AMC can be used in electronic protect applications as well. For example, a friendly transmitter can monitor the modulation type of the adversary jammer and switch the friendly communications to another modulation type to render the adversary jamming useless.

Other than the mentioned applications spectrum management is another military application that needs to be mentioned. In military communications, spectrum is limited. AMC can be used to determine the modulation types of friendly communications in the target band. This information, in turn, can be used to efficiently allocate communications frequencies, minimize interference, and optimize the performance on a specific band within the available spectrum.

Overall, AMC technology enhances military capabilities in electronic warfare and spectrum management. With AMC technology, military forces can make informed decisions, effectively utilize available resources, and maintain communication superiority on the battlefield.

1.3. Related Work

AMC literature broadly consists of two parts. They are non-blind AMC and blind AMC. Non-blind AMC methods assume specific information regarding the physical channel, such as scattering, fading, and power decay over the channel, is known at the receiver. The combination of such characteristics of the channel is called channel state information (CSI). On the other hand, blind AMC methods do not assume any knowledge of the channel, making blind AMC a more complex problem than non-blind AMC.

A relatively simple and popular non-blind AMC method is likelihood-based AMC. This methodology consists of finding likelihood functions under different hypotheses regarding the modulation scheme and selecting the one that maximizes the likelihood function. Over time, different likelihood methods emerged based on the available CSI at the receiver.

Maximum-likelihood (ML) classification can be used when complete CSI is available. Different parameters regarding the channel, such as channel gain, noise variance and phase offset, are known or estimated at the receiver. With the parameters, likelihood functions for each candidate modulation type are calculated, and the modulation type that maximizes the likelihood function is selected. ML classification methods are studied in [2–4].

Having complete CSI available at the receiver is not practical. To tackle the problem of incomplete CSI, variations of the ML classification methods are developed. Average likelihood ratio test (ALRT), first introduced in [5], computes the average of the parameters by integrating over the entire parameter space. The proposed method was later used in [6].

While ALRT is an improvement over plain ML classification, it induces further complexity to an already complex problem by applying integration over the entire parameter space. Generalized likelihood ratio test (GLRT), introduced in [7], simplifies the process without compromising the performance. Instead of treating the unknown parameters as random variables, it treats them as deterministic but unknown quantities. It replaces the integration operation by maximization of the likelihood function with respect to the parameters. GLRT can be considered as a mixture of an ML classifier and an ML estimator.

GLRT fails to discriminate between the nested constellations, that is, different orders of the same modulation types, *e.g.*, 8PSK and 16PSK, or 16QAM and 64QAM. Hybrid likelihood ratio test (HLRT) is proposed in [7], right after GLRT, as an im-

provement. HLRT computes the average over the received symbols and maximizes the likelihood function with respect to the carrier phase.

If the received signal is long enough, that is, if the signal has enough samples to construct an empirical probability distribution, that empirical probability distribution can be used to classify the modulation scheme. In such a scenario, distribution is affected mainly by channel parameters and modulation type. If the parameters are known or estimated, the modulation scheme is the primary determinant of the distribution. Various goodness of fit (GoF) tests are developed for such scenarios.

An example GoF test is the Kolmogorov-Smirnov (KS) test, first proposed in [8]. It evaluates the similarity between two empirical probability density functions (PDF). KS test is applied to the AMC field in [9]. In [10, 11], the test is studied further, and complexity is reduced while enhancing the performance by applying various optimizations.

Cramér-von Mises (CvM) test is an alternative to KS test. It evaluates the GoF between a theoretical cumulative distribution function (CDF) and an empirical distribution function. First introduced in [12,13], then generalized into the two-sample case in [14] and applied to AMC for the first time in [15].

Anderson-Darling (AD) test also uses empirical distributions. It is a test that checks whether a sample belongs to a given distribution. It is rather similar to the KS test, with an addition of a weight multiplier. This weighting mechanism emphasizes the tails of the distribution more. It is introduced in [16].

Blind AMC methods utilize various features of the received signal. These features are used to make inferences via various decision mechanisms, such as machine learning models. The mentioned features can broadly be grouped into spectral features, wavelet-based features and higher-order statistics-based features. Spectral features are based on the amplitude, frequency, and phase of the received signal, all affecting the signal's

spectrum, hence the name. They are studied in [17] and [18]. The wavelet-based features utilize wavelet transform to extract meaningful information from the signal. They are introduced in [19], and further adopted in [20], [21]. Higher-order statistics-based methods generally utilize higher-order moments and higher-order cumulants of the received signal. These types of methods are first proposed in [22], further improved in [23] and more recently studied in [24].

After finding useful features containing meaningful information about the received signal, machine learning methods can also be applied to those features. Various methods are explored in the AMC field.

The k-nearest-neighbour (KNN) method is a suitable method for the AMC task. Candidate modulation types, the number of classes, are known at the receiver. Extracted features are used to construct a feature space for the classification task, and the received signal is classified according to the location within the feature space. KNN for the AMC task is introduced in [25].

Support vector machines (SVM) are also applicable. SVM determines a hyper-plane in the feature space and decides on the class based on the location of the sample with respect to the decided hyper-plane. SVMs are inherently binary classifiers but can be used for multi-class classification with particular procedures. SVM applications for AMC are studied in [26, 27].

Logistic regression has applications in AMC as well. In [28], it is used to reduce the dimensionality of the feature space. A designated function of the extracted features is passed to the logistic function to obtain the probabilities for two different modulation types. Logistic regression is inherently a binary classifier as well, but again, there are particular procedures for multi-class classification tasks.

Recently, with the improved computational power and the advent of machine learning techniques, the merit of the feed-forward neural networks (FNN) and the

convolutional neural networks (CNN) in the AMC field have been explored. FNN architectures are studied in [29–31]. There are various architectures with different optimizations. These architectures use engineered features to perform classification tasks. To lighten the feature engineering work, CNNs are utilized. CNNs extract essential features from the inputs by using numerous filters at convolution layers. Some proposed CNN architectures are [32, 33].

The previously mentioned literature focuses on single-carrier communications systems. The multi-carrier communications literature focuses primarily on orthogonal frequency-division multiplexing (OFDM) signals and shows significant similarity considering the methods and features employed.

In [34], the ALRT and HLRT methods are applied to OFDM signals with index modulation. The authors examine the classification performance assuming full CSI and no CSI available at the receiver. On the other hand, in [35], the expectation-maximization (EM) algorithm is proposed for the AMC task. The authors consider various channel impairments, such as clock timing offset, phase noise, and Doppler shift, and find that the classification performance is significant.

In [36, 37], the moments of the received OFDM signal are used as classification features. The former examines a multi-path channel under fading conditions, and the latter considers carrier frequency offset (CFO), which is the root cause of intercarrier interference (ICI).

The wavelet transform-based features are used in the OFDM context in a relatively limited capacity. In [38], the wavelet transform of the received signal is computed two times, only to discriminate between the single-carrier modulations and multi-carrier modulations while omitting the classification of the modulation scheme of the communications.

Cumulants appear in [39–41] within the OFDM context. [39] cumulants to a lesser

extent, classifying the signal as single-carrier or multi-carrier. [40] is a rather realistic study, assuming no CSI is available at the receiver and employs a radio-frequency (RF) testbed for simulations. [40,41] present a machine learning approach employing decision trees and random forests.

In [42], authors employ Mel Frequency Cepstral Coefficients (MFCCs) to extract features from the OFDM signals and feed extracted features to a fully connected neural network to classify the modulation type. Furthermore, in [43], the raw in-phase and quadrature (IQ) samples of the received signals are fed into a CNN for feature extraction, and extracted features are fed into a fully connected neural network for classification of the modulation type, similar to [42].

1.4. Contribution of the Thesis

The AMC literature focuses mainly on single-carrier communications. As a result, there is a gap between the literature for single-carrier and multi-carrier communications. This thesis addresses the mentioned gap by presenting two new architectures for multi-carrier AMC. They can be summarized as follows:

Both architectures employ a filter bank consisting of multiple fast Fourier transform (FFT) filters and use parallel convolutional networks to extract features. Based on the extracted features, a classification is performed, and this is where the two architectures differ.

- The first architecture's classification is based on the most confident vote across the filter bank-convolutional network modules, independent of each other.
- In the second architecture, the outputs of the filter bank-convolutional network modules are combined to create a context. The classification is done by a fully-connected neural network.

1.5. Organization of the Thesis

In Chapter 2, an overview of the concepts that are essential to the proposed architectures is provided. Chapter 3 gives a chronological summary of the AMC literature and comparisons and contrasts across the methods. In Chapter 4, the proposed architectures are described in detail. The signal model and the dataset are introduced. The architecture of the filter bank, the convolutional networks and the classification mechanism are explained. The results for each architecture are discussed. Finally, in Chapter 5, the thesis is concluded by stating the final remarks and possible future work.

2. FUNDAMENTALS

2.1. Multi-Carrier Modulation

2.1.1. Overview

Multi-carrier modulation is a modulation technique where a high-rate bitstream is divided into parallel low-rate substreams and transmitted over multiple channels simultaneously. Given that the bit rates of the substreams are lower than the actual bitstream, the corresponding bandwidths of the subchannels are much less than the bandwidth of the actual bitstream. The main idea is to parcel the total available frequency spectrum into multiple narrowband subchannels and independently modulate the subcarrier associated with each subchannel.

The advantages of multi-carrier modulation are discussed in [44] and are summarized as follows. The division into lower-rate substreams increases the spectral efficiency of the multi-carrier modulation system relative to single-carrier systems. Moreover, the channels over which the substreams are sent are orthogonal to each other, and the number of substreams is determined such that each subchannel has a bandwidth less than the coherence bandwidth of the channel. As a result, subchannels mainly do not interfere with each other, and they experience relatively flat fading under the fading channel conditions, respectively. Multi-carrier modulation also simplifies the equalization process at the receiver by enabling the receiver to apply frequency-domain equalization techniques.

To illustrate the concept, consider a channel with an available bandwidth of BW, bitrate R, and coherence bandwidth $BW_c < BW$, resulting in frequency-selective fading over the available band. With multi-carrier modulation, the available bandwidth is split into U subchannels, each with bandwidth $BW_i = BW/U$, and consequently with bitrate $R_i \approx R/U$. For sufficiently large U, the bandwidth of a subchannel is smaller

than the coherence bandwidth, that is, $BW_i = BW/U \ll BW_c$, resulting in flat-fading within each sub-channel.

There are two possible implementations regarding the spacing of the subchannels. Subchannels can be separated from each other completely, or they can overlap to some extent. Illustrations of the resulting spectra for both implementations can be seen in Figure 2.1.

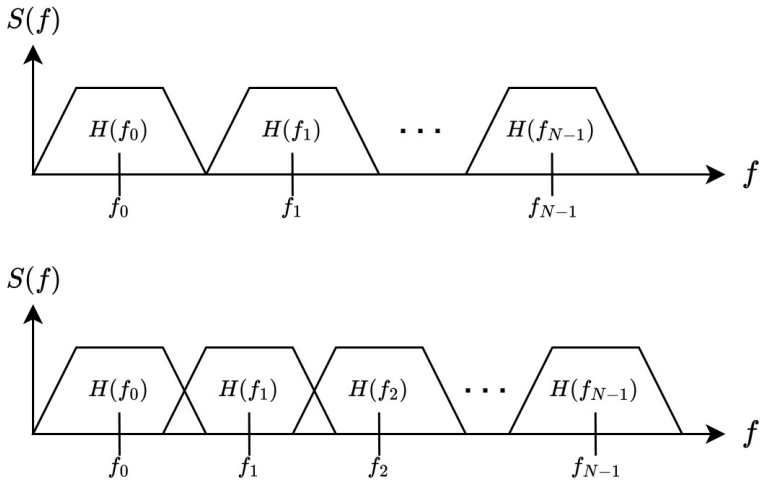

Figure 2.1: Non-overlapping versus overlapping subcarrier spacing.

The former is spectrally inefficient, considering the larger bandwidth requirement and reduced bitrate for each subchannel, together with U near-ideal, expensive filters. Letting the subcarriers overlap while preserving the subcarriers' (approximate) orthogonality improves the spectral efficiency and throughput of the system. The set of signals in the form

$$s_i(t) = \cos\left(2\pi(f_0 + \frac{i}{T_n}t) + \phi_i\right) \quad \text{for } i = 0, 1, \ldots, \tag{2.1}$$

comprises a set of orthogonal basis functions for any subcarrier phase offset ϕ_i, therefore can be used as subcarriers.

Transmitted signal is composed of the sum of the modulated signals associated with each subcarrier. Typically, employed modulation schemes are quadrature-amplitude modulation (QAM) and phase-shift-keying (PSK). The resulting signal is

$$s(t) = \sum_{i=0}^{U-1} s_i g(t) \cos(2\pi f_i t + \phi_i), \tag{2.2}$$

where s_i is the modulated symbol associated with the i^{th} subcarrier, $g(t)$ is the pulse-shaping signal, f_i is the center frequency and ϕ_i is the phase offset of the i^{th} carrier, respectively. An illustration of a multi-carrier transmitter architecture can be seen in Figure 2.2.

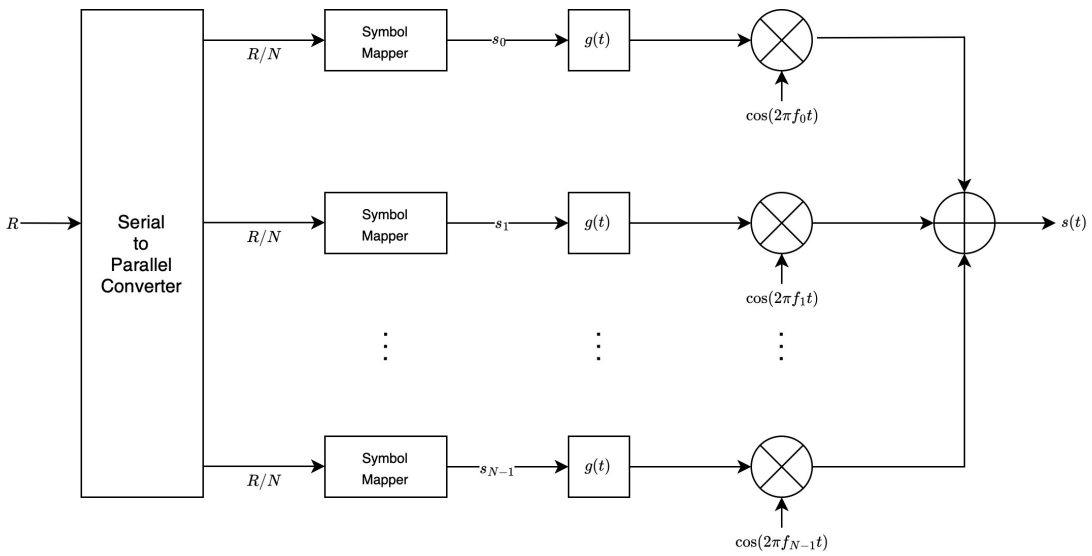

Figure 2.2: Illustration of multi-carrier transmitter.

Each substream is extracted at the receiver by passing the received signal through N near-ideal filters for non-overlapping subcarriers, correlating the signal with the orthogonal basis functions, and sampling the correlator output for overlapping subcarriers. After the substreams are extracted, each is demodulated and mapped into the bits to recover the original bitstream.

The discrete-time implementation of multi-carrier modulation is efficient and is

called orthogonal frequency-division multiplexing. It is widely adopted in modern communication standards.

2.1.2. Orthogonal Frequency-Division Multiplexing

OFDM is a frequency-division multiplexing scheme which implements the multi-carrier modulation using discrete Fourier transform (DFT) and its inverse, inverse discrete Fourier transform (IDFT).

To define DFT, let $x[n]$, $0 \leq i \leq N-1$ be a discrete-time sequence. The N-point DFT is defined as

$$\mathrm{DFT}\{x[n]\} = X[i] \triangleq \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} x[n] e^{-j2\pi i \frac{n}{N}}, \quad 0 \leq i \leq N-1. \tag{2.3}$$

IDFT is similarly defined. For the frequency domain signal $X[i]$, the original sequence is recovered by computing the IDFT. It is defined as

$$\mathrm{IDFT}\{X[i]\} = x[n] \triangleq \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} X[i] e^{j2\pi i \frac{n}{N}}, \quad 0 \leq i \leq N-1. \tag{2.4}$$

In modern computers, DFT and IDFT are efficiently implemented using FFT and inverse fast Fourier transform (IFFT) algorithms, introduced in [45].

In OFDM, the input bitstream is linearly modulated, *e.g.*, using PSK, QAM, producing a complex-valued symbol sequence. The symbol sequence is then passed through a serial-to-parallel converter to stack the desired number of symbols, usually in powers of two, so that each symbol will be transmitted through a subcarrier. In OFDM, it is assumed that the stacked symbols are the individual frequency components of the OFDM symbol. The time-domain signal to be transmitted physically is obtained by computing the IDFT of the stacked symbols. The number of points of the IDFT is equal to the number of stacked symbols.

Let complex-valued symbol stream be denoted by $X[0], X[1], \ldots, X[N-1]$. Then the time-domain OFDM symbol is given by the IDFT of the symbol sequence, that is

$$x[n] = \frac{1}{\sqrt{N}} \sum_{i=1}^{N-1} X[n] e^{j2\pi i \frac{n}{N}}, \quad 0 \leq n \leq N-1. \tag{2.5}$$

After obtaining $x[n]$, cyclic prefix is added to the signal. That is, if p is the length of the cyclic prefix, last p items of the sequence $x[n]$ is prefixed in front of the sequence, resulting in a new sequence with length $N+p$. Finally, obtained sequence is converted back to serial, converted to analog and carried to the target passband. At the receiver, same steps in reverse order applied to recover the original bitstream. A diagram of an OFDM system can be seen in Figure 2.3.

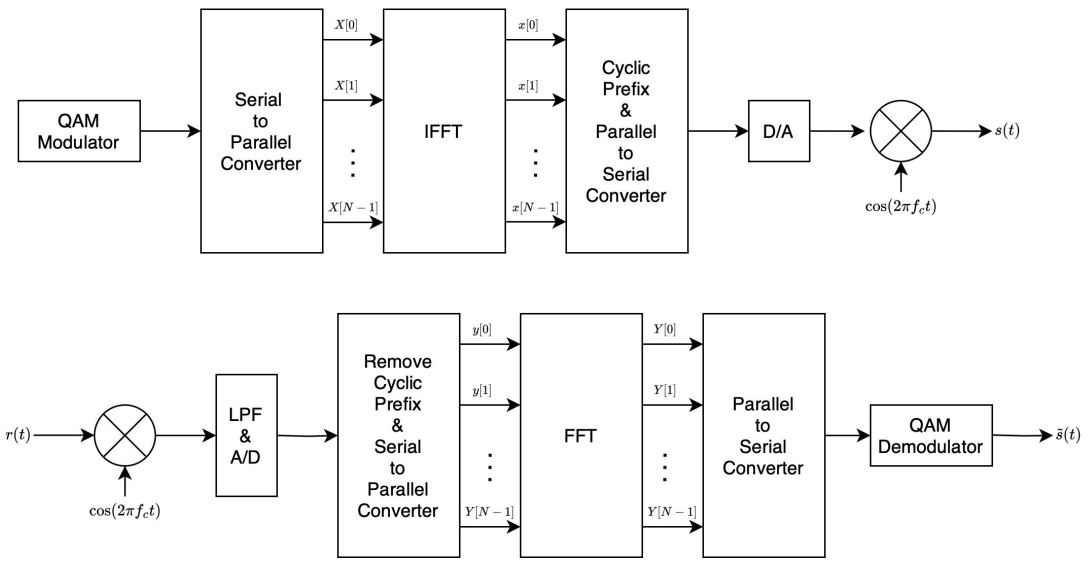

Figure 2.3: An illustration of an OFDM transmitter and receiver, above and below, respectively.

2.2. Deep Learning

2.2.1. Feed-Forward Neural Networks

An overview of the subject based on [46, 47] will be provided in this chapter. The feed-forward neural networks aim to approximate some function h. It does so by defining a mapping $\mathbf{y} = h(\mathbf{x}; \theta)$ and learning the set of hyperparameters θ that results in the best approximation of the function. Feed-forward neural networks are also known as multi-layer perceptron (MLP).

Neurons are the building blocks of the neural networks. They are also called nodes or units. Each neuron represents a vector-to-scalar function. A neuron takes inputs from the other connected neurons and produces its activation value. At each neuron, the weighted sum of the inputs is computed, then a non-linear activation function is applied to the sum, to produce the activation. Non-linear activation functions enable neural networks to learn complex, non-linear relationships between input and outputs by introducing non-linearity to the network. Weights represent the strength of the connections between the neurons, and they are learned throughout the training process. A visual representation of a neuron in an feed-forward neural network is below.

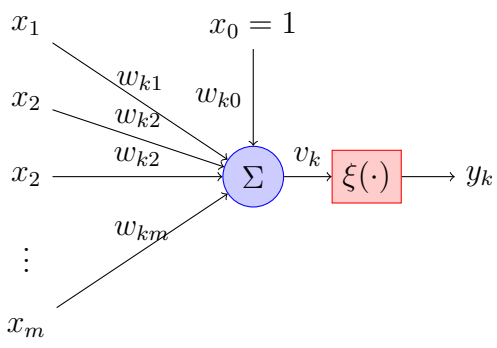

Figure 2.4: Model of a neuron.

In Figure 2.4, inputs to the neuron are denoted by x_i, weights of the incoming connections to the neuron is denoted by w_{kj}. Then, the equations describing a single

neuron can be written as the following two equations [47]:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j, \tag{2.6}$$

$$y_k = \xi(u_k + b_k), \tag{2.7}$$

where u_k is the linear combiner output given the input signals, b_k is the bias term, $\xi(\cdot)$ is the non-linear activation function and y_k is the output of the neuron. Alternatively, bias term can be included in the summation by setting $x_0 = 1$ and $w_{k0} = b_k$. Based on latter formulation, model of a neuron can be summarized in Figure 2.4.

A collection of neurons in the same hierarchy is called a layer. A feed-forward neural network consists of an input layer, an output layer, and a single or multiple hidden layer(s). The input layer takes the input, which can be numeric data or other data types transformed into a numerical representation. The output layer produces the output, $\mathbf{y}$. Hidden layers are the layers between the input and the output layers. The neurons at one layer are connected to the neurons at the next layer through the weighted connections, previously referred to as the weights. A visual representation of a neural network can be seen in Figure 2.5.

The objective of the training process is to adjust the weights so that the network approximates the function h in the best possible way. The goal is achieved by minimizing a cost function, which quantifies the difference between the output of the network and the output of the target function h, by varying the weights at each training step proportional to their gradient with respect to the cost function, utilizing iterative optimization algorithms. An important example of such algorithms is called the backpropagation algorithm. The optimal values of the parameters of a neural network are computed by applying the backpropagation algorithm iteratively.

These neural networks are called feed-forward because the information flows in only one direction, starting from the input layer, through hidden layers, and finally

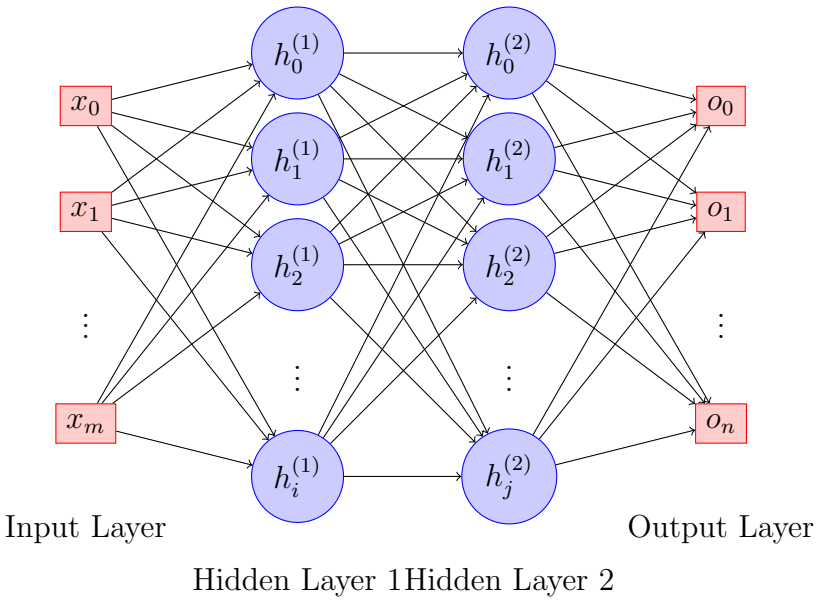

Figure 2.5: Illustration of a neural network.

to the output layer, giving rise to the naming feed-forward. Feed-forward neural networks have been successfully applied to a wide range of tasks, including image and speech recognition, natural language processing, and regression problems. Hence, they significantly important as they lay the ground for more complex architectures, such as recurrent neural networks (RNNs) and CNNs. The latter will be discussed in the upcoming section.

2.2.2. Convolutional Neural Networks

CNNs are neural networks designed explicitly for processing structured grid-like data, such as images, sound, and time series. They have succeeded remarkably in various computer vision tasks, including image classification, object detection, and image segmentation. As the name suggests, they use convolution operation in at least one of its layers. Convolution operation in two dimensions is defined as

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n). \tag{2.8}$$

However, many neural network libraries implement the cross-correlation operation, and call it convolution. The cross-correlation operation is defined as follows. The difference between the convolution and the cross-correlation is that the kernel is flipped in former, whereas the kernel is not flipped in the latter. Since this detail is not of significant importance from a deep learning perspective, there is no problem using the latter.

In most cases, a convolution layer consists of three stages [46]. First stage is convolution operation. Multiple parallel convolutions on input data is performed to produce set of linear activations. Then, these activations are fed through non-linear activation functions, such as rectified linear unit (ReLU). Finally, a pooling operation is applied to the non-linear activations. These three stages are detailed below.

The convolution layer applies the convolution operation to the input using multiple kernels in parallel. A kernel is usually a multidimensional grid of weights, smaller than the input and learned through the training process. The kernel is moved across the input and element-wise multiplied with the overlapping part of the input. The result of the element-wise multiplication is summed, and the sum comprises the result of the convolution operation at that particular input region. This process is repeated for all the input data and results with a matrix, the feature map. The size of the feature map depends on the input size, kernel size, stride, and padding of the input.

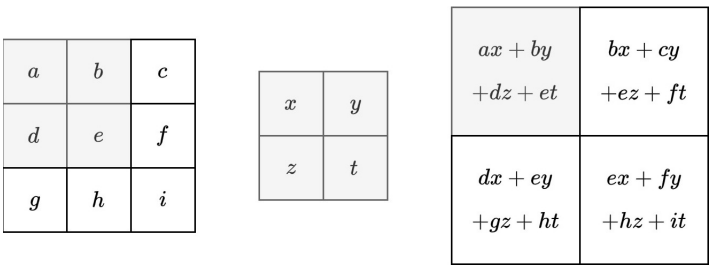

Figure 2.6: Illustration of convolution operation. Input is on the left, kernel is in the middle and the corresponding output is on the right.

Applying kernels to the input differentiates convolutional neural networks from

feed-forward neural networks. In feed-forward neural networks, each output unit interacts with each input unit through matrix multiplications. In contrast, in convolutional neural networks, the connections between the input and output units become sparse due to kernels being smaller than the input. Sparse connectivity helps to detect the local features within the grid-like data while also improving the computational efficiency of convolutional neural networks.

The pooling operation produces the summary statistic of each location on the output of the convolution operation. There are different types of pooling; among them, max pooling, average pooling, and L_2 pooling are widespread. Pooling operation helps to make the learned representations invariant to variations. The invariance to translation concept is particularly important if it is more important whether a feature exists than where it is. Since pooling summarizes the output, it can be applied with a stride greater than one. Pooling results in improved statistical and computational efficiency [46].

Commonly, convolutional layers are used for learning and extracting the discriminative features from the data. After learning the relevant features using a set of convolutional layers, the output is flattened into a column vector in order to remove the spatial nature of the output and fed into a feed-forward neural network for classification or prediction purposes. This architecture allows convolutional neural networks to learn hierarchical representations with increasing abstractions, resulting in an improved classification or prediction performance of the following feed-forward neural network.

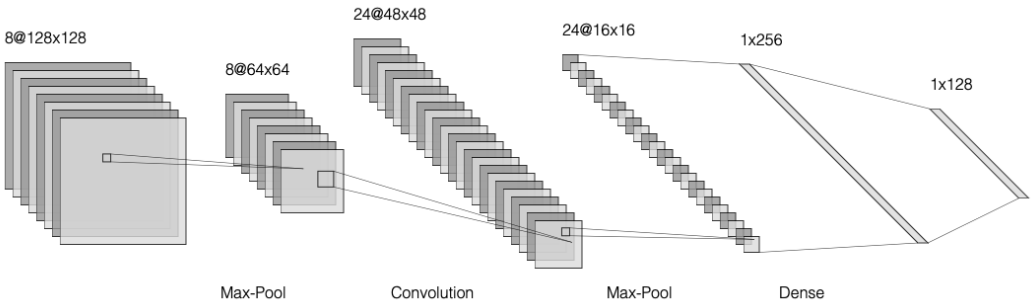

Figure 2.7: Illustration of a convolutional neural network.

3. AMC METHODS

3.1. Likelihood-Based Methods

3.1.1. Maximum Likelihood Classification

ML-based classification is well-studied in the literature. Some example works are [2–7]. ML-based classification models assume that the channel parameters, such as symbol rate, carrier frequency, carrier phase, SNR, and timing offset, are known or estimated. The only unknown parameter is the modulation type of the received signal.

The fundamental ML-based classification is studied in [3]. Consider M possible modulation schemes. Let m_i denote the i^{th} modulation scheme. Furthermore, the i^{th} modulation scheme has $M(i)$ number of, possibly complex, symbols. Symbols of a given modulation scheme are given as

$$I(m_i) = \{\delta_0, \delta_1, \ldots, \delta_{M(i)-1}\}. \tag{3.1}$$

Let r_i denote the i^{th} received symbol after the signal is demodulated and passed through a matched filter. r_i is expressed as

$$r_i = r_{I,i} + jr_{Q,i}, \tag{3.2}$$

where r_I and r_Q denote the in-phase and quadrature components of r, respectively. For a given modulation type m_i, the probability that the received signal belongs to the i^{th} modulation scheme is given by the density

$$\mathbb{P}_R(r_{I,n}, r_{Q,n} \mid M = m_i) = \frac{1}{M(i)} \sum_{k=1}^{M(i)} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(r_{I,n}-\delta_{I,k})^2-(r_{Q,n}-\delta_{Q,k})^2}{2\sigma^2}}, \tag{3.3}$$

where δ_I and δ_Q denote the in-phase and quadrature components of the symbol μ,

respectively. To obtain the joint likelihood function in a compact way, denote the vector of K demodulated symbols as

$$\mathbf{r} = \begin{bmatrix} r_1 & r_2 & \dots & r_K \end{bmatrix}^{\intercal}. \tag{3.4}$$

Since the received symbols are assumed to be independent, joint likelihood function is obtained by multiplying the marginal likelihood functions, that is

$$\mathbb{P}_{\mathbf{R}}(\mathbf{r}_I, \mathbf{r}_Q \mid M = m_i) = \prod_{n=1}^{K} \mathbb{P}_R(r_{I,n}, r_{Q,n} \mid M = m_i). \tag{3.5}$$

Joint likelihood function is evaluated for all values of m_i, and m_i that produces the highest likelihood value is selected.Therefore, the decision rule is given as

$$m = \arg\max_{m} \mathbb{P}_{\mathbf{R}}(\mathbf{r}_I, \mathbf{r}_Q \mid M = m). \tag{3.6}$$

3.1.2. Average Likelihood Ratio Test

The maximum likelihood classifier assumes all the received signal parameters are known. However, in reality, this is rarely the case.

If some of the parameters are unknown, these parameters can be modelled as random variables. Then, these random variables are either known or have hypothesized PDFs. Since there are possibly multiple numbers of unknown parameters, this is a composite hypothesis testing problem and can be solved by applying multiple pairwise likelihood ratio tests.

The average likelihood ratio test (ALRT), introduced in [5], employs the known or hypothesized PDFs to average over all possible values of the unknown parameters.

Resulting likelihood function takes the form of

$$\mathbb{L}_{ALRT}(\mathbf{r} \mid M = m_i) = \int_{\boldsymbol{\Theta}} f_{\mathbf{R}}(\mathbf{r} \mid \boldsymbol{\Theta} = \theta, M = m_i) f_{\boldsymbol{\Theta}}(\theta \mid M = m_i) d\theta, \tag{3.7}$$

where θ is the vector of random variables correspond to unknown parameters, $f_{\boldsymbol{\Theta}}$ is the corresponding known or hypothesized joint PDF. Based on the ALRT likelihood function, a likelihood ratio test can be formed as

$$\frac{\mathbb{L}_{ALRT}(\mathbf{r} \mid M = m_i)}{\mathbb{L}_{ALRT}(\mathbf{r} \mid M = m_j)} \underset{m_j}{\overset{m_i}{\gtrless}} \frac{\mathbb{P}_M(m_j)}{\mathbb{P}_M(m_i)}, \tag{3.8}$$

where $\mathbb{P}_M(m_i)$ is the prior probability of the modulation type m_i. In the case of uniform priors, the test becomes

$$\frac{\mathbb{L}_{ALRT}(\mathbf{r} \mid M = m_i)}{\mathbb{L}_{ALRT}(\mathbf{r} \mid M = m_j)} \underset{m_j}{\overset{m_i}{\gtrless}} 1. \tag{3.9}$$

The tests are applied in a pairwise fashion to exhaust the parameter space. ALRT results in the optimal classification if only the PDFs of the unknown parameter used in likelihood derivation are the true PDFs.

3.1.3. Generalized Likelihood Ratio Test

ALRT is computationally expensive and usually results in non-linear classifier structures. An alternative possible approach to the problem of unknown parameters in the received signal is to estimate the unknown parameters and use the estimates as if they are the actual parameters. This approach is proposed in [7], and called the generalized likelihood test (GLRT).

The ML-estimate of the unknown parameters are given by

$$\hat{\theta}_{ML} = \arg\max_{\theta} f_{\mathbf{R}}(\mathbf{r} \mid \boldsymbol{\Theta} = \theta). \tag{3.10}$$

Using the estimated parameters, likelihood function for GLRT is

$$\mathbb{L}_{GLRT}(\mathbf{r} \mid M = m_i) = f_{\mathbf{R}}(\mathbf{r} \mid \mathbf{\Theta} = \hat{\theta}_{ML}, M = m_i), \tag{3.11}$$

and the likelihood ratio test is given by

$$\frac{\mathbb{L}_{GLRT}(\mathbf{r} \mid M = m_i)}{\mathbb{L}_{GLRT}(\mathbf{r} \mid M = m_j)} \underset{m_i}{\overset{m_j}{\gtrless}} \frac{\mathbb{P}_M(m_j)}{\mathbb{P}_M(m_i)},$$

where $\mathbb{P}_M(m_i)$ is the prior probability of the modulation type m_i. In the case of uniform priors, the test becomes

$$\frac{\mathbb{L}_{GLRT}(\mathbf{r} \mid M = m_i)}{\mathbb{L}_{GLRT}(\mathbf{r} \mid M = m_j)} \underset{m_j}{\overset{m_i}{\gtrless}} 1. \tag{3.12}$$

Again, likelihood ratio test are conducted in a pairwise fashion. A favorable results of the GRLT is that the ML-estimates of the unknown parameters are known and can be used after the modulation classification stage in the receiver.

3.1.4. Hybrid Likelihood Ratio Test

In some cases, it is useful to treat some parameters as random and some others as unknown but deterministic. This approach is called hybrid likelihood ratio test (HLRT).

Let θ_R denote the vector random parameters, and θ_D denote the vector of deterministic but unknown parameters. In this setup, the likelihood function is derived by averaging over the random parameters, while using the ML-estimate of the unknown parameter, that is

$$\mathbb{L}_{HLRT}(\mathbf{r}|M = m_i) = \max_{\theta_{\mathbf{D}}} \int_{\mathbf{\Theta}_R} f_{\mathbf{R}}(\mathbf{r} \mid \mathbf{\Theta}_R = \theta_R, M = m_i) d\theta_R, \tag{3.13}$$

and the hybrid likelihood ratio test is formed as

$$\frac{\mathbb{L}_{HLRT}(\mathbf{r}|M=m_i)}{\mathbb{L}_{HLRT}(\mathbf{r}|M=m_j)} \underset{m_i}{\overset{m_j}{\gtrless}} \frac{\mathbb{P}_M(m_j)}{\mathbb{P}_M(m_i)}, \tag{3.14}$$

where $\mathbb{P}_M(m_i)$ is the prior probability of the modulation type m_i. In the case of uniform priors, the test becomes

$$\frac{\mathbb{L}_{HLRT}(\mathbf{r} \mid M=m_i)}{\mathbb{L}_{HLRT}(\mathbf{r} \mid M=m_j)} \underset{m_j}{\overset{m_i}{\gtrless}} 1. \tag{3.15}$$

3.2. Feature-Based Methods

3.2.1. Spectral Features

There are numerous features stemming from the spectral properties of the signal that can be used for modulation classification tasks. These features are mainly derived from characteristics of the signal, such as instantaneous amplitude $A(t)$, instantaneous phase $\phi(t)$, and the instantaneous frequency $f(t)$. Some of these features are introduced in [17] and [18], then are used to build a decision tree for modulation classification task. Mentioned features are detailed below.

The first feature is called maximum normalized-centered instantaneous amplitude. To define this feature, let m_a denote the average value of the instantaneous amplitude over a single frame, and be defined as

$$m_a = \frac{1}{N} \sum_{i=1}^{N} A(i). \tag{3.16}$$

Furthermore, let $A_n(i)$ denote the normalized instantaneous amplitude at time instants

$t = i/f_s$, where $i = 1, 2, \ldots, N$, and be defined as

$$A_n(i) = \frac{A(i)}{m_a}. \tag{3.17}$$

Also, let $A_{nc}(i)$ denote the normalized-centered instantaneous amplitude at time instants $t = i/f_s$, where $i = 1, 2, \ldots, N$, and be defined as

$$A_{nc}(i) = A_n(i) - 1. \tag{3.18}$$

Finally, define the first feature, maximum normalized-centered instantaneous amplitude as

$$\gamma_{max} = \max \left| \mathrm{DFT}(A_{nc}(i)) \right|, \tag{3.19}$$

where DFT operation denotes the discrete Fourier transform. This feature carries meaningful information when the amplitude of the signal changes based on the utilized modulation type.

The second feature is the standard deviation of the absolute value of the non-linear component of the instantaneous phase. Let $\phi_{NL}(t)$ be the value of the non-linear component of the instantaneous phase at $t = i/f_s$, where $i = 1, 2, \ldots, N$. Let C be the number of samples in the set $\{\phi(i)\}$ for which $A_n(i) > a_t$, where a_t is the threshold in amplitude below which the estimation of the instantaneous phase is too sensitive to the noise. Then, the second feature is defined as

$$\sigma_{ap} = \sqrt{\left(\frac{1}{C} \sum_{A_n(t) > a_t} \phi_{NL}^2 \right) - \left(\frac{1}{C} \sum_{A_n(t) > a_t} |\phi_{NL}| \right)^2}. \tag{3.20}$$

This feature carries meaningful information when the phase of the signal changes based on the utilized modulation scheme.

The third feature is the standard deviation of the non-linear component of the instantaneous phase. Using the definitions given above, the feature is defined as

$$\sigma_{dp} = \sqrt{\left(\frac{1}{C}\sum_{A_n(t)>a_t}\phi_{NL}^2\right) - \left(\frac{1}{C}\sum_{A_n(t)>a_t}\phi_{NL}\right)^2}. \tag{3.21}$$

The difference compared to the previous feature is that the value of the instantaneous phase is used instead of its absolute value in the second term inside the square root. Similarly, this feature is useful when the instantaneous phase of the signal carries information.

The fourth feature is the standard deviation of the absolute value of the normalized-centered instantaneous amplitude. Again, using the definitions given above, the fourth feature is given by

$$\sigma_{aa} = \sqrt{\left(\frac{1}{N}\sum_{i=1}^{N}A_{nc}^2(i)\right) - \left(\frac{1}{N}\sum_{i=1}^{N}|A_{nc}(i)|\right)^2}. \tag{3.22}$$

Again, this feature is significant when the amplitude of the signal carries information. Specifically, this feature is useful for distinguishing between ASK-2 and ASK-4. The fifth feature is the standard deviation of the absolute value of the normalized-centered frequency. Let m_f denote the average value of the instantaneous frequency over a single frame and be defined as

$$m_f = \frac{1}{N}\sum_{i=1}^{N}f(i). \tag{3.23}$$

Furthermore, let $f_c(i)$ denote the centered instantaneous frequency and be defined as

$$f_c(i) = f(i) - m_f. \tag{3.24}$$

Moreover, let $f_{nc}(i)$ denote the normalized-centered frequency of the signal and be

defined as

$$f_{nc}(i) = \frac{f_c(i)}{r_b}, \tag{3.25}$$

where r_b is the bitrate of the communications. Then, the fifth feature is given by

$$\sigma_{fa} = \sqrt{\left(\frac{1}{C}\sum_{A_n(i)>a_t} f_{nc}^2(i)\right) - \left(\frac{1}{C}\sum_{A_n(i)>a_t} |f_{nc}(i)|\right)^2}. \tag{3.26}$$

This feature is meaningful when the information is carried on frequency of the signal. Specifically, this feature is useful for distinguishing between frequency-shift-keying (FSK) modulations, especially FSK-2 and FSK-4.

The sixth feature is the spectral symmetry around the carrier frequency. It is given by

$$P = \frac{\sum_{i=1}^{f_{nc}} |X_c(i)|^2 - \sum_{i=1}^{f_{nc}} |X_c(i) + f_c + 1|^2}{\sum_{i=1}^{f_{nc}} |X_c(i)|^2 + \sum_{i=1}^{f_{nc}} |X_c(i) + f_c + 1|^2}, \tag{3.27}$$

where $X_c(i)$ is the discrete time Fourier transform of the signal and f_c+1 is the sample number corresponding to the carrier frequency. This feature is useful for distinguishing between numerous amplitude modulations.

The seventh feature is the standard deviation of the normalized-centered instantaneous amplitude. Using the definitions given above, this feature can be written as

$$\sigma_a = \sqrt{\left(\frac{1}{C}\sum_{A_n(i)>t_a} A_{nc}^2(i)\right) - \left(\frac{1}{C}\sum_{A_n(i)>t_a} A_{nc}(i)\right)^2}. \tag{3.28}$$

This feature is useful for distinguishing analog and digital amplitude modulations.

The eight feature is the kurtosis of the normalized-centered instantaneous ampli-

tude. Based on the definitions given above, the feature is given by

$$\mu_{42}^{A} = \frac{\mathbb{E}[A_{nc}^{4}(i)]}{\left(\mathbb{E}[A_{nc}^{2}(i)]\right)^{2}}. \tag{3.29}$$

The ninth and the final feature is the kurtosis of the normalized instantaneous frequency. Based on the definitions given above, this feature is given by

$$\mu_{42}^{f} = \frac{\mathbb{E}[f_{nc}^{4}(i)]}{\left(\mathbb{E}[f_{nc}^{2}(i)]\right)^{2}}, \tag{3.30}$$

and it is useful for distinguishing between analog and digital frequency modulations.

Overall, the nine mentioned features are used to build a decision tree classifier in [17] and [18]. The mentioned approaches are similar to feature-based machine learning approaches utilized today.

3.2.2. Wavelet Transform-Based Features

The utilization of wavelet transform on a digital modulation signal yields unique patterns for various types, facilitating straightforward identification through simplified processing. As a result, wavelet transform-based features comprise a set of efficient and useful features for modulation classification tasks. Some of these features for PSK, FSK and QAM signals are discussed in [6], [20], and [21], and summarized below.

The continuous wavelet transform (CWT) of a signal $s(t)$ is defined as

$$\mathrm{CWT}(s; a, \tau) = \int_{-\infty}^{\infty} s(t)\psi_{a,\tau}^{*}(t)dt, \tag{3.31}$$

where ψ is the wavelet function, a and τ are the parameters of the wavelet function and $*$ denotes the complex conjugation operation.

There are numerous wavelet functions developed over time. One of the examples

is the Haar Wavelet, which is defined as

$$\psi(t) = \begin{cases} 1 & 0 \leq t \leq \frac{1}{2}, \\ -1 & \frac{1}{2} \leq t \leq 1, \\ 0 & \text{otherwise.} \end{cases} \tag{3.32}$$

Haar wavelet is a fundamental and computationally simple wavelet function. It is also pretty useful for modulation classification tasks. To illustrate, consider the Haar wavelet transform of PSK and FSK signals. The amplitude of the transformation of PSK signal is

$$|\mathrm{CWT}_{PSK}(a,\tau)| = \frac{4\sqrt{S}}{\omega_c\sqrt{a}} \sin^2\left(\omega_c \frac{aT_s}{4}\right), \tag{3.33}$$

where ω_c denotes the angular frequency of the carrier and S denotes the power of the signal, and T_s is the sampling interval. On the other hand, the amplitude of transformation of FSK signal is

$$|\mathrm{CWT}_{FSK}(a,\tau)| = \frac{4\sqrt{S}}{\sqrt{a}(\omega_c+\omega_i)} \sin^2\left((\omega_c+\omega_i)\frac{aT_s}{4}\right), \quad 1 \leq i \leq M, \tag{3.34}$$

where the above definitions hold and ω_i denotes the frequency of the i^{th} symbol in the symbol book. Furthermore, the amplitude of the transformation of a QAM signal is given by

$$|\mathrm{CWT}_{QAM}(a,\tau)| = \frac{4S_i}{a\sqrt{\omega_c}} \sin^2\left(\frac{aT_s}{4}\right), \tag{3.35}$$

where S_i is the amplitude of the i^{th} symbol in the symbol book.

These transformations are distinct for different types of modulations. These distinctions are used effectively for modulation classification tasks.

3.2.3. Cumulant-Based Features

Cumulants are set of quantities that are related to the shape of a PDF, similar to moments. The first-order, second-order and the third-order cumulants are the same as the respective order moments, but higher-order cumulants are polynomials of the moments. The moment-generating function (MGF) or the characteristic function of a probability distribution is given by

$$\Psi(t) = \mathbf{E}[e^{ixt}] = \sum_{n=0}^{\infty} \frac{1}{n!}(\mu_n(it))^n. \tag{3.36}$$

Using the MGF definition given above, writing the Taylor expansion of the expression

$$\ln \Psi(t) = \sum_{n=0}^{\infty} \frac{1}{n!}(\kappa_n(it))^n, \tag{3.37}$$

gives the cumulants as the coefficients of the expansion. Therefore, the i^{th} cumulant can be obtained by taking the i^{th} order derivative of the logarithmic-MGF given above and evaluating its value at $t = 0$, provided that the MGF exists, *i.e.*,

$$C_i = \ln \Psi^{(i)}(0). \tag{3.38}$$

The higher-order cumulants can be written in terms of the lower orders. The expression is

$$C_{mn} = \text{cum}(r[n], \ldots, r[n], r^*[n], \ldots, r^*[n]), \tag{3.39}$$

where $*$ denotes the complex conjugation operation. The subscript mn means that there are $m - n$ terms in the not-conjugated part, and n term in the conjugated part

within the brackets. The cumulants up until the 6^{th} order are:

$$C_{20} = M_{20}, \tag{3.40}$$

$$C_{21} = M_{21}, \tag{3.41}$$

$$C_{40} = M_{40} - 3M_{20}^2, \tag{3.42}$$

$$C_{41} = M_{41} - 3M_{20}M_{21}, \tag{3.43}$$

$$C_{42} = M_{42} - |M_{20}|^2 - 2M_{21}^2, \tag{3.44}$$

$$C_{60} = M_{60} - 15M_{20}M_{40} + 30M_{20}^3, \tag{3.45}$$

$$C_{61} = M_{61} - 5M_{21}M_{40} - 10M_{20}M_{41} + 30M_{20}^2M_{21}, \tag{3.46}$$

$$C_{62} = M_{62} - 6M_{20}M_{42} - 8M_{21}M_{41} - M_{20}M_{40} + 6M_{20}^2M_{20} + 25M_{21}^2M_{20}, \tag{3.47}$$

$$C_{63} = M_{63} - 9M_{21}M_{42} + 12M_{21}^3 - 6M_{20}M_{41} + 18M_{20}M_{21}M_{20}, \tag{3.48}$$

where M_{pq} is defined as

$$M_{pq} = \mathbf{E}[(r[n])^{p-q}(r^*[n])^q]. \tag{3.49}$$

In the AMC context, the cumulants form complex features that carry significant information about the modulation type of the signal. Such informative features can be fed into a classification algorithm, either a statistical model or a machine learning algorithm, to achieve the AMC task.

3.3. Machine Learning Based Methods

3.3.1. K-Nearest Neighbours

K-nearest neighbors (KNN) is a supervised learning method utilized in AMC tasks. It is a relatively simple approach. Training and test samples consist of designated features. Dimensionality of the feature space depends on the number of designated features. Overall, the process can be summarized as:

i Store training samples and corresponding labels.

ii Upon receiving test sample, compute the distance between the test sample and each training sample.

iii Sort the distances in ascending order. Call the sorted list L.

iv Truncate L such that it has length K. Call the new list L'.

v Select the class with largest number of occurrences in L'.

The parameter K is significant. A smaller value of K results in sharp decision boundaries and makes the algorithm prone to noisy samples. In contrast, a larger value results in smooth decision boundaries and may overlook the local patterns within the feature space. Therefore, an optimal value of K should be determined using validation methods. For example, setting $K = 1$ causes the algorithm to classify the test sample as same class as the closest training sample. On the other hand, with N training samples, setting $K = N$ causes the algorithm to classify the test sample as the majority class within the training samples. Various features can be used for AMC task in KNN. These feature include but not limited to the features discussed in the previous section. Cumulant-based features are an example.

In the second step given in the algorithm above, distances between the samples are computed. There are numerous available distance functions. Some examples are Euclidean distance, Manhattan distance and cosine distance. As the number of samples grow, distance computations might become expensive. As a result, numerous feature reduction methods are utilized together with KNN algorithm.

Keeping the mentioned points in consideration, KNN is a simple and effective algorithm for AMC task.

3.3.2. Support Vector Machine

Support vector machines (SVMs) are another machine learning tool that is used for classification tasks. SVMs can be preferred where the samples in the feature space

are not linearly separable. They produce non-linear decision boundaries, hyperplanes, by generating linear boundaries in a transformed, higher-dimensional version of the original feature space. The hyperplane is located such that the margin that separates the classes is maximized.

Let $\mathbf{w}$ denote the normal vector of the hyperplane. Denoting the positive and negative samples by $\mathbf{x}^+$, $\mathbf{x}^-$, respectively the constraints can be written as

$$\mathbf{w} \cdot \mathbf{x}^+ + b \geq 1, \tag{3.50}$$

$$\mathbf{w} \cdot \mathbf{x}^- + b \leq -1. \tag{3.51}$$

Let y_i be a variable that takes value 1 if the sample is positive, and value -1 if the sample is negative. Then, the constraint given above can be combined into a single equation given by

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1. \tag{3.52}$$

$$\tag{3.53}$$

For the samples on the margins it is required that

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 = 0. \tag{3.54}$$

Furthermore, the goal is to maximize the margin. The margin is given by the distance between the two classes. It can be written as the projection of the distance between two classes onto the normal of the hyperplane, *i.e.*,

$$G = (\mathbf{x}^+ - \mathbf{x}^-) \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|}. \tag{3.55}$$

Using the constraints above, the margin can be written as

$$M = \frac{2}{\|\mathbf{w}\|}. \tag{3.56}$$

Maximizing M corresponds to minimizing $1/M = \|\mathbf{w}\|$, and that is equivalent to minimizing $\frac{1}{2}\|\mathbf{w}\|^2$. Using the approach, and minding the constraints, Lagrangian can be written as

$$L = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_i \alpha_i (y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1), \tag{3.57}$$

where α_i are the Lagrange multipliers. Solving the optimization problem yields the normal vector of the hyperplane as

$$\hat{\mathbf{w}} = \sum_i^N \hat{\alpha}_i y_i \mathbf{x}_i. \tag{3.58}$$

Then, denoting the predicted class for the unknown sample u by $\hat{c}$, the decision rule can be written as

$$\hat{c} = \begin{cases} +1, & \mathbf{w} \cdot \mathbf{u} + b \geq 0, \\ -1, & \mathbf{w} \cdot \mathbf{u} + b \leq 0. \end{cases} \tag{3.59}$$

It is important to highlight that the results is related to the samples only through the inner product.

The discussed mechanism works well with linearly separable classes. When the samples are not linearly separable, a transformation of the feature space is useful. By transforming the feature space, a linearly separable form is obtained. Suppose $\phi(\mathbf{x})$ is the transformed feature. Then, the hyperplane equation becomes

$$\hat{\mathbf{w}} = \sum_i^N \hat{\alpha}_i y_i \left\langle \phi(\mathbf{x}), \phi(\mathbf{x}_i) \right\rangle, \tag{3.60}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product operator. As a result, knowledge of the result of the inner product in the transformed feature space is sufficient, instead of knowing the transformation itself. The function which gives the result of the inner product in the transformed feature space is called the kernel function an denoted by $K(\mathbf{x_i}, \mathbf{x_j})$. There are numerous kernel functions. Some are polynomial, radial basis and neural network kernel functions.

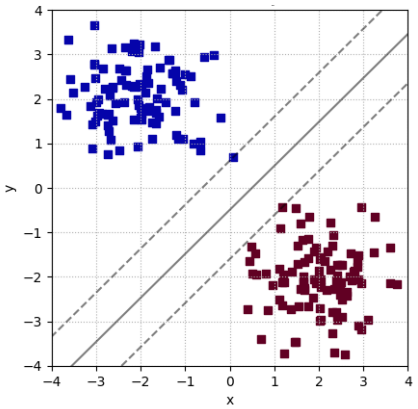

Figure 3.1: Illustration of a linear SVM classifier.

Since the mechanism relies on a separating hyperplane, SVMs are binary classifiers by nature. However, it is possible to combine multiple SVMs for non-binary classification tasks. One approach for such cases is called one-versus-rest. In this approach a single class is selected and all the remaining classes are combined into a single class. Then, classification is performed. This process is repeated for all classes, and the class with highest score is selected as the predicted class. This approach might become computationally expensive when there are large number of training samples or classes.

For AMC tasks, all the mentioned features can be used stand-alone or as a set for utilizing the SVM classifier.

3.3.3. Logistic Regression

Logistic regression is widely used for binary classification as well. For AMC tasks, it can be used as well, provided that a proper feature set is selected. In logistic regression, the aim is to determine the probability of a sample belonging to one of the classes, by expressing the logarithm of the logits of the event in terms of a linear combination of the independent variables. Here, logit denotes the ratio of probability of an event happening versus not happening. That is

$$\text{logit}(p) = \ln \frac{p}{1-p}. \tag{3.61}$$

For a training sample (x_i, y_i), and parameter set θ, probability of x_i belonging to class 1 is denoted as

$$h_\theta(x_i) = \mathbb{P}(y_i = 1 \mid x_i; \theta), \tag{3.62}$$

then, the probability that the same sample belongs to class 0 is denoted as

$$\mathbb{P}(y_i = 0 \mid x_i; \theta) = 1 - h_\theta(x_i). \tag{3.63}$$

These two equations can be written in single equation as follows.

$$\mathbb{P}(y_i \mid x_i; \theta) = h_\theta(x_i)^{y_i}(1 - h_\theta(x_i))^{(1-y_i)}. \tag{3.64}$$

To obtain the optimal set of parameters, maximum-likelihood estimation can be used. Likelihood function is given by

$$\mathbb{L}(\theta) = \prod_{i=1}^{N} \mathbb{P}(y_i \mid x_i; \theta) \tag{3.65}$$

$$= \prod_{i=1}^{N} h_\theta(x_i)^{y_i}(1 - h_\theta(x_i))^{(1-y_i)}, \tag{3.66}$$

then, the log-likelihood function, denoted by $l(\theta)$ is

$$l(\theta) = \log\left(\mathbb{L}(\theta)\right) \tag{3.67}$$

$$= \sum_{i=1}^{N} y_i \log h_\theta(x_i) + (1 - y_i) \log 1 - h_\theta(x_i). \tag{3.68}$$

The ML-estimate for θ is then,

$$\hat{\theta}_{ML} = \arg\max_{\theta} l(\theta). \tag{3.69}$$

Unfortunately, there is no closed-form solution. For maximizing the likelihood function, iterative methods are used. Some widely used methods are gradient ascend and Newton-Raphson method.

3.3.4. Neural Networks

An overview of the neural networks was provided at the end of Chapter 2. Hence, further discussion is omitted in this chapter.

The FNNs can learn from custom features, given that the features contain discriminative features regarding the target classes. In this vein, for AMC tasks, most of the features mentioned previously in this chapter might comprise a proper input for a suitable FNN architecture.

The previously mentioned features are derived through feature engineering, and CNNs are quite instrumental from the feature engineering viewpoint. CNNs can extract the relevant features by applying numerous filters and pooling operations to the raw input data. Then, the extracted features can be used to perform the classification task through a proper FNN architecture. In the AMC context, raw data might comprise IQ samples of the received signal or some relevant transformation of the received signal.

Overall, neural networks have a wide range of applications, and the AMC is also among them. In fact, the contribution of the thesis is based on the FNN and CNN architectures.

4. AMC FOR SUB-CARRIERS OF OFDM SIGNALS

4.1. Signal Model

In this section, a discrete baseband signal s_i is considered. Given the similar nature of the problem for both baseband and band-pass signals, and for the sake of simplicity, the analysis focuses on the baseband signals. The signal is assumed to be affected by an additive-white-Gaussian-noise (AWGN) process. In practice, such noise processes can result from imperfect hardware or interference between the signal of interest and the other signals within the channel. AWGN process is complex-valued and defined as

$$\eta_i = R_i + jI_i, \tag{4.1}$$

where the PDFs of the real and imaginary parts of the process are given by zero mean normal distribution, that is

$$f_{R_n}(x) = f_{I_n}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x_i^2}{2\sigma^2}}. \tag{4.2}$$

Therefore, the signal model is given as

$$r_i = s_i + \eta_i, \quad i = 0, 1, \ldots. \tag{4.3}$$

The quality of the channel is chiefly determined by the standard deviation of the noise process. Higher standard deviations result in a lower signal-to-noise ratio (SNR), hence lower signal quality. On the other hand, lower standard deviations result in higher SNR and, hence, higher signal quality.

4.2. Evaluation Metrics

Since the problem of interest consists of classifying the modulation type, the accuracy metric is a natural choice. Accuracy is defined as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{4.4}$$

where TP, TN, FP and FN are true positive, true negative, false positive and false negative, respectively. It can be described as the percentage of correct predictions among all predictions. The higher the accuracy score, without overfitting, the better the model.

Other than the accuracy, confusion matrices will also be analyzed. A confusion matrix is a compact representation the performance of a classification algorithm. The labels are located on the rows, and the predictions are in the columns. Examining a confusion matrix can help determine if the algorithm confuses a class with another, hence the name.

Finally, receiver operating characteristics (ROC) curves will be examined. The ROC curve illustrates a binary classifier's performance at varying TP and FP rates. An ideal classifier would achieve a TP rate of 1 while attaining an FP rate of 0, and a classifier with no skill would have equal TP and FP rates equal to 0.5. By examining the ROC plots, the relative performances of the models will be assessed. Given that there are three classes in the problem of interest, models will be evaluated for each class in a one-versus-rest scheme.

4.3. Dataset

The dataset consists of IQ samples of the received signal as input and the modulation type as the output. For a single sample, input to the model is generated by the following procedure.

i Generate random bits. For a training sample $\mathbf{x}_i$,

$$\mathbf{b}_i = \begin{bmatrix} b_{i0} & b_{i1} & \dots b_{iB-1} \end{bmatrix}, \quad b_{ij} \in \{0, 1\} \quad \text{and} \quad j < B, \tag{4.5}$$

where b_{ij} denotes the value of the j^{th} bit in the i^{th} training sample, and B denotes the length of the bitstream.

ii Modulate the bitstream to obtain the symbol sequence

$$\mathbf{s}_i = \mathbb{M}_t(\mathbf{b}_i), \quad t \in \{\text{4PSK}, \text{8PSK}, \text{16QAM}\}, \tag{4.6}$$

where $\mathbb{M}$ denotes the modulation operation and t denotes the modulation type.

iii Apply serial-to-parallel conversion to the symbol sequence

$$\mathbf{s}_i^P = \text{S2P}(\mathbf{s}_i) = \begin{bmatrix} \mathbf{s}_i^{(0)} & \mathbf{s}_i^{(1)} & \dots & \mathbf{s}_i^{(\frac{L}{N}-1)} \end{bmatrix}. \tag{4.7}$$

Here, S2P denotes the serial-to-parallel conversion operator, L is the length of the symbol sequence and N is the length of the IFFT operation for the next step.

iv Generate the OFDM symbol by computing the IFFT of the symbol sub-sequence.

$$\mathbf{x}_i^{TX} = \begin{bmatrix} \mathcal{F}_N^{-1}\left(\mathbf{s}_i^{(0)}\right) & \mathcal{F}_N^{-1}\left(\mathbf{s}_i^{(1)}\right) & \dots & \mathcal{F}_N^{-1}\left(\mathbf{s}_i^{(\frac{L}{N}-1)}\right) \end{bmatrix}, \quad N \in \{256, 512, 1024\}, \tag{4.8}$$

where $\mathcal{F}_N^{-1}(\cdot)$ denotes the N-point IFFT operation, and $\mathbf{x}_i^{TX}$ denotes the transmitted frame.

v Add AWGN to the OFDM symbol.

$$\mathbf{x}_i = \mathbf{x}_i^{TX} + \eta_i, \tag{4.9}$$

where η_i is the noise vector, defined as

$$\eta_i = \begin{bmatrix} \eta_{i0} & \eta_{i1} & \dots & \eta_{ij-1} & \dots & \eta_{iL-1} \end{bmatrix}, \quad j < L, \tag{4.10}$$

with $\eta_{ij} \sim N(0, \sigma)$, and autocorrelation function satisfying

$$R_{\eta\eta}(i,j) = \delta_{ij}\sigma^2 = \begin{cases} \sigma^2, & i = j, \\ 0, & i \neq j. \end{cases} \tag{4.11}$$

In the first step, the length of the bitstream depends on the modulation scheme that will be used in the second step. Denote bits-per-symbol of the modulation scheme by k and denote the number of samples in a frame by L. Then, the length of the bitstream is Lk. For example, if the modulation scheme is 8PSK and $L = 2048$, then $k = 3$ and the bitstream length is $Nk = 2048 \times 3 = 6144$.

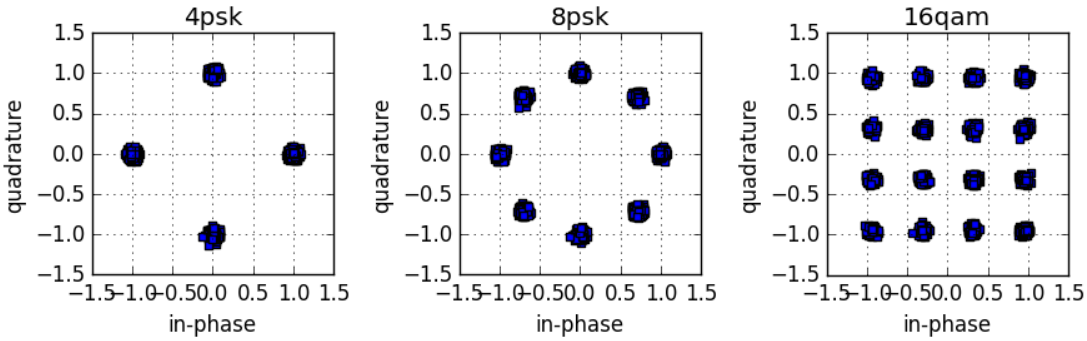

Figure 4.1: Constellation diagrams.

In the second step, possible modulation schemes are 4PSK, 8PSK, and 16QAM. In the fourth step, possible IFFT lengths are 256, 512, and 1024. One transmit frame is composed of 2048 samples. As a result, if the IFFT is 256-point, there are 8 OFDM symbols; if the IFFT is 512-point, there are 4 OFDM symbols, and if the IFFT is 1024-point, then there are 2 OFDM symbols within a frame. Formally, denote frame length by L and IFFT length by N; then, a frame will contain $\frac{L}{N}$ OFDM symbols. To obtain such a structure, L complex symbols are generated through steps 1 and 2. Then through step 3, the symbol sequence is chopped into $\frac{L}{N}$ subsequences, and through step 4, the IFFT of each subsequence is computed, resulting in $\frac{L}{N}$ OFDM symbols. Then, the generated OFDM symbols are concatenated to obtain length L complex-valued

OFDM signal. The final signal has the dimensions 2×2048, the first row being the in-phase component, where the second row is the quadrature component. Finally, as the last step, noise with varying power levels is added to the generated OFDM signal to impose channel conditions. Denoting training samples as $(\mathbf{x}_i, y_i)$, the procedure described above produces $\mathbf{x}_i$. The corresponding label, y_i, is the modulation type used in Step ii.

4.4. Architectures

4.4.1. Filter Bank

Both models that will be discussed use FFT and IFFT to perform classification tasks. The exact size of the transformation is unknown at the receiver beforehand, yet the possible size values are known. Therefore, to perform the task efficiently, multiple FFT filters are required. To satisfy the requirement, a filter bank concept is used. A filter bank is a container for multiple filters, initialized with a set of filter sizes. In the scope of the thesis, a fixed set of sizes is used. The set is $\{256, 512, 1024\}$. The filters within the filter bank perform transforms in a specific way. The process is described in four steps below. L denotes the number of symbols in the received frame.

i Divide the input into $\frac{L}{N}$ pieces with size N.
ii Compute N-point FFTs of each piece.
iii Concatenate the transformed pieces in the original order.
iv Return the transformed and concatenated samples.

An example from the filtering process is the following. Denote received OFDM symbol by $\mathbf{r}$ and a filter with F_N where N is the FFT size. There are three filters, F_{256}, F_{512}, and F_{1024}. Each filter consumes a copy of $\mathbf{r}$ to produce a transformed symbol sequence, denoted by $\tilde{\mathbf{r}}_{256}$, $\tilde{\mathbf{r}}_{512}$ and $\tilde{\mathbf{r}}_{1024}$, respectively. This operation is effectively a short-time Fourier transform (STFT). An illustration of the filter bank concept can be seen in Figure 4.2. After the filtering operation, linearly modulated signals belonging to the

individual subcarriers are obtained.

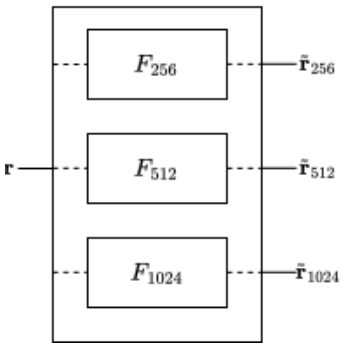

Figure 4.2: Filter bank illustration.

Filtering operation is a pre-processing step for learning and classification processes. An illustration of the real part of the filtered signal can be seen in Figure 4.3. The correct FFT size can be determined even with the naked eye using filtered signals in high SNR values. However, the task becomes more complicated when the SNR value decreases.

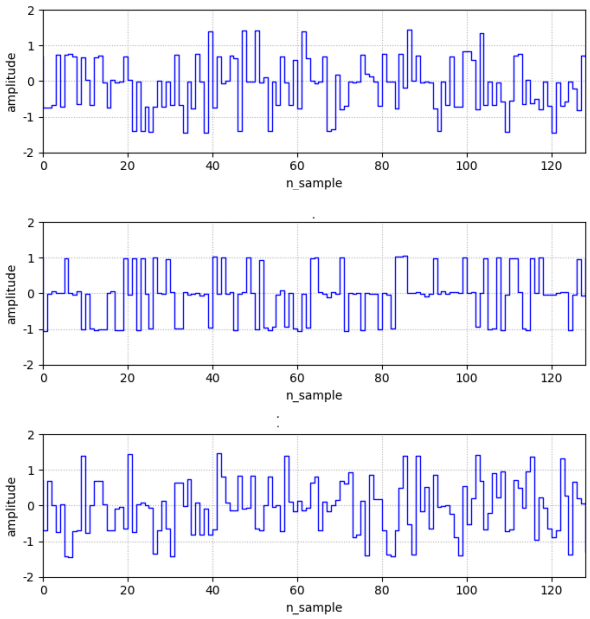

Figure 4.3: Real part of the output of the filter bank. 256-point is at the top, 512-point is in the middle and 1024-point is at the bottom.

4.4.2. Baseline Model

4.4.2.1. Model Description. This architecture includes a filter bank, consisting of 3 filters, multiple CNNs and numerous pooling layers, followed by a sigmoid function. The CNNs act as feature extractors for a given input. The sigmoid function maps the output of the model between 0 and 1, essentially converting the output vector into probabilities. The model contains a total of 5943 trainable parameters. A layout of the model can be seen in the Table 4.1.

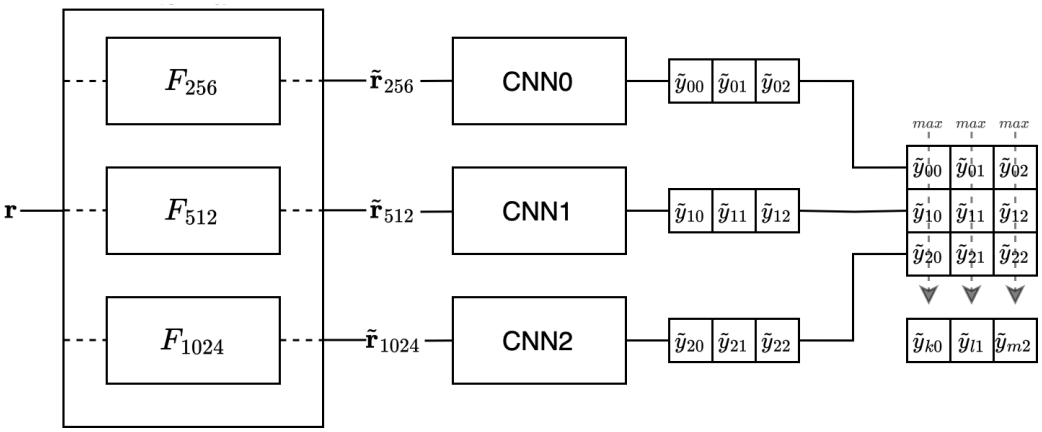

Figure 4.4: Diagram of the baseline.

The i^{th} input vector, $\mathbf{x}_i$, consists of received IQ samples. The vector is passed through the filter bank, meaning that the vector is filtered with three different filters, each with different FFT sizes. The output of each filter is then fed into an individual CNN; there are 3 different CNNs in this case. Outputs of the CNNs are stacked on top of each other, each row corresponding to a CNN's output with size 1×3, producing a 3×3 matrix. Then, the maximum of each column is taken to produce the classification output. The final maximum operation causes the final probability of a class to be the maximum value of all three probabilities produced by individual networks for the same class. The described architecture is illustrated in Figure 4.4.

The main idea behind this architecture is to select the highest CNN output for

a given class among all the CNN outputs for that given class. This concept is almost equivalent to selecting the output of the most confident CNN for each class. After selecting the most confident CNNs for each class, the maximum value among the previously selected activations is selected as the final class decision. Described mechanism sounds exciting, considering it selects the highest probable class each time.

Table 4.1: Layout of the baseline model.

	Output Dimensions		
Layer	**FilterBank1**	**FilterBank2**	**FilterBank3**
Input	$1 \times 2 \times 2048$	$1 \times 2 \times 2048$	$1 \times 2 \times 2048$
Convolution1	$32 \times 3 \times 2048$	$32 \times 3 \times 2048$	$32 \times 3 \times 2048$
MaxPool1	$3 \times 2 \times 1025$	$3 \times 2 \times 1025$	$3 \times 2 \times 1025$
Convolution2	$4 \times 3 \times 1026$	$4 \times 3 \times 1026$	$4 \times 3 \times 1026$
MaxPool2	$4 \times 2 \times 514$	$4 \times 2 \times 514$	$4 \times 2 \times 514$
Convolution3	3	3	3
RowStack	3×3		
Maximum	3		
Softmax	3		

4.4.2.2. Experiments & Results. The model is implemented in PyTorch [48] and trained using a built-in backpropagation algorithm. The implemented model has three convolutional layers and two max-pooling layers. The model uses Leaky ReLU as the activation function with a negative slope of 0.05. The convolution operations result in 32, 4, and 1 feature maps in the given order. Max pooling kernel has dimensions 2×2. At each convolution and pooling layer, inputs are padded such that the output of the convolution or pooling layer has the exact dimensions as the input, except for the last convolution layer. There are three CNNs, one for each filter; each produces a 1×3 output. Then, these vectors are stacked on each other, and the previously described maximum operation is applied. For training the model, the previously described dataset is used. The dataset has different SNR values, modulation types and

FFT lengths. There are 1000 items for each unique triplet of SNR value, modulation type and FFT length, each consisting of 2048 IQ samples. The dataset contains 72000 items, corresponding to approximately 150 million data points. Adam [49] optimizer with learning rate $1e-4$, together with cross-entropy loss function is used. The batch size is 1 due to computational constraints. The samples are shuffled, and the complete dataset is iterated three times; that is, the training process uses 3 epochs.

While the described classification-based-on-confidence mechanism works as intended, the learning behaviour of the network performs under expectations. The implemented model achieves 50% accuracy for 3-class classification task. Given that a uniform random guess classification would achieve close to 33.3% accuracy, it is safe to say that the model adds value. It learns the patterns in the data and performs better than random guessing. While beating a random guess is not nearly a significant milestone, the results inspire to improve the model.

The confusion matrices for different SNR values can be seen in Figure 4.5. The effect of the SNR can be observed from -4 dB to 8 dB plots. After 8 dB, the classification performance seems to converge; and it is observed that the model mostly confuses 8PSK samples with 16QAM samples and vice-versa. The reason behind the limited performance is the following. All the networks are being trained independently from each other; furthermore, the extracted features are immediately used for making a decision instead of forming a context for the decision. One individual CNN is unaware of other CNNs' outputs. As a result, overall decision performance is limited. Based on this finding, a new hypothesis which claims that the performance would improve if the CNNs were to collaborate, is formed, and in the next section, such an architecture is explored.

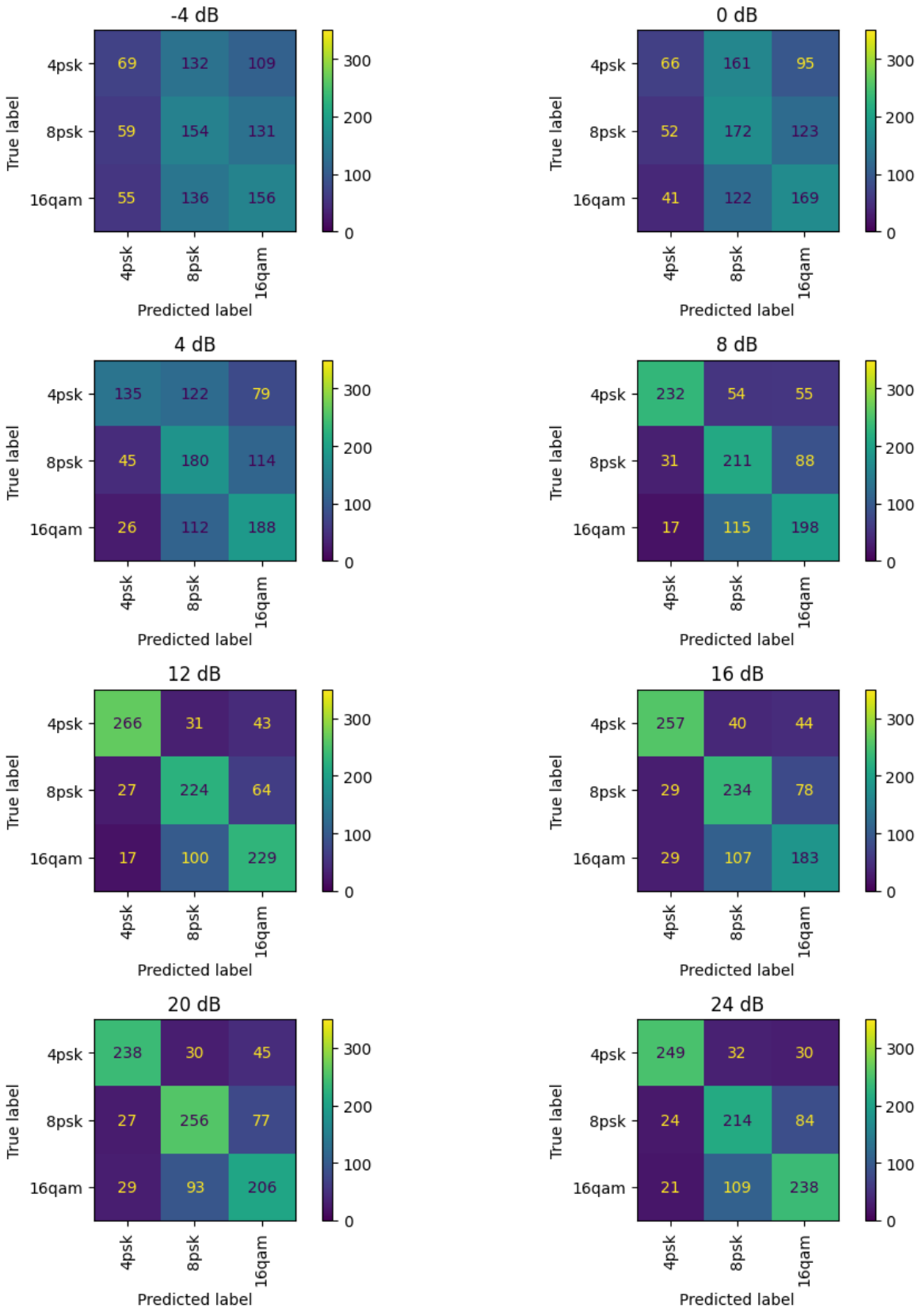

Figure 4.5: Confusion matrices for the baseline model.

4.4.3. Alternative Model

4.4.3.1. Model Description. The model again includes a filter bank, consisting of 3 filters, multiple CNNs and additional FC layers and a sigmoid layer at the end. The CNNs act as feature extractors for a given input, and the FC layers act as the classifier based on the extracted features. The sigmoid function maps the output of the last FC layer between 0 and 1, essentially converting the output vector into probabilities. The model contains 10287 trainable parameters. A layout of the model can be seen in the Table 4.2.

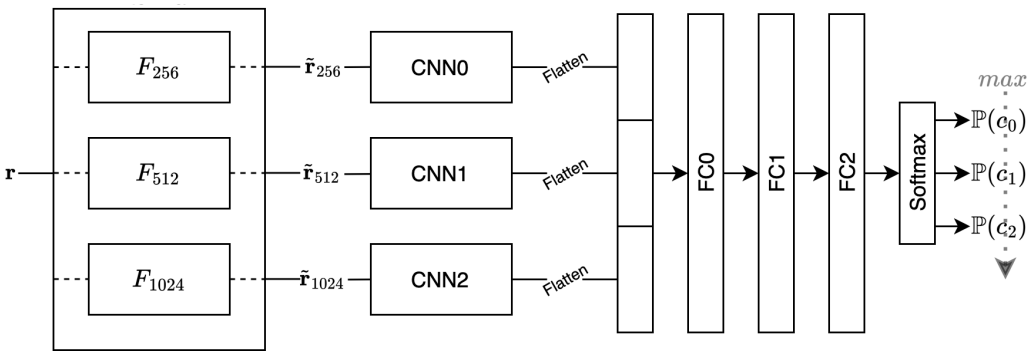

Figure 4.6: Diagram of the alternative model.

Again, the input vector consisting of IQ samples is passed through the filter bank. The output of each filter is then fed into an individual CNN; there are three different CNNs in this case as well. Outputs of the CNNs are then flattened into single vectors. Then, the resulting three vectors are concatenated to produce a single input, the input of the classifier FC network. The produced vector is fed into the feed-forward network. The network output is passed through a sigmoid layer to convert the class outputs into probabilities. Finally, the class with the maximum probability is selected as the prediction. The described architecture is illustrated in Figure 4.6. The main idea behind this architecture is to collect the extracted features into a single vector and use all of them together to make a better-informed decision compared to the previous architecture. Previously, the decision mechanism was based on a simple maximum

operation, whereas in this architecture, the features are used to create a context for a better-informed classifier network. This mechanism can be interpreted as CNNs collaborating via a feed-forward network to make a better decision.

Table 4.2: Layout of the alternative model.

	Output Dimensions		
Layer	**FilterBank1**	**FilterBank2**	**FilterBank3**
Input	$1 \times 2 \times 2048$	$1 \times 2 \times 2048$	$1 \times 2 \times 2048$
Convolution1	$32 \times 3 \times 2048$	$32 \times 3 \times 2048$	$32 \times 3 \times 2048$
MaxPool1	$3 \times 2 \times 1025$	$3 \times 2 \times 1025$	$3 \times 2 \times 1025$
Convolution2	$4 \times 3 \times 1026$	$4 \times 3 \times 1026$	$4 \times 3 \times 1026$
MaxPool2	$4 \times 2 \times 514$	$4 \times 2 \times 514$	$4 \times 2 \times 514$
Flatten	4112	4112	4112
Concatenate	12336		
Dense1	128		
Dense2	64		
Dense3	3		
Softmax	3		

4.4.3.2. Experiments & Results. The model is implemented in PyTorch [48] as well and trained using a built-in backpropagation algorithm. The implemented model has two parts, CNN part and feed-forward part. In CNN part, there are 3 separate CNNs, each with two convolutional layers and two max-pooling layers. CNNs use leaky ReLU as the activation function with a negative slope of 0.05. Optimal value for the negative slope parameters was decided based on empirical evaluations based on grid search. The convolution operations result in 32 and 4 feature maps in the given order. Max pooling kernel has dimensions 2×2. At each convolution and pooling layer, inputs are padded such that the output of the convolution or pooling layer has the exact dimensions as the input. In feed-forward part, there are three dense layers, which implements 12336×128, 128×64 and 64×3 transformations, where the last transformation produces outputs

for each class.

The same dataset, with same SNR, modulation type and FFT size triplets, is used for training the implemented model. Adam optimizer, [49], with learning rate 0.0001, together with cross-entropy loss function, is used. The optimal learning rate was determined based on the empirical evaluations based on grid search. Again, the batch size is 1, due to computational constraints. Training process used 3 epochs.

The results indicate that the collaboration between the CNNs indeed improve the performance significantly; and it is safe to say that the shortcoming of the previously described model was determined correctly. This architecture achieves 91% accuracy for the same 3-class classification task. Not only the result is much better than random guessing, that is 33%, it is also much better than the previous architecture, which achieved 50% accuracy. This improvement is based on the combined usage of the extracted features via a concatenation operation and a feed-forward network. The confusion matrices for a range of SNR values can be seen in the Figure 4.7.

In -4dB setting, model just outputs random guesses. Given that it is a significantly low SNR value, this can be considered as expected behavior. Moreover, it can be seen in the lower SNR region the model confuses between the PSK modulations only. This observation might be attained to the fact that the 4PSK constellation is a direct subset of 8PSK constellation, and model finds it hard to distinguish between them, when the received IQ samples are spread around the constellation points further. In the case where SNR is higher, model accuracy gets significantly better, achieving around 98.5% accuracy score at and above 16dB.

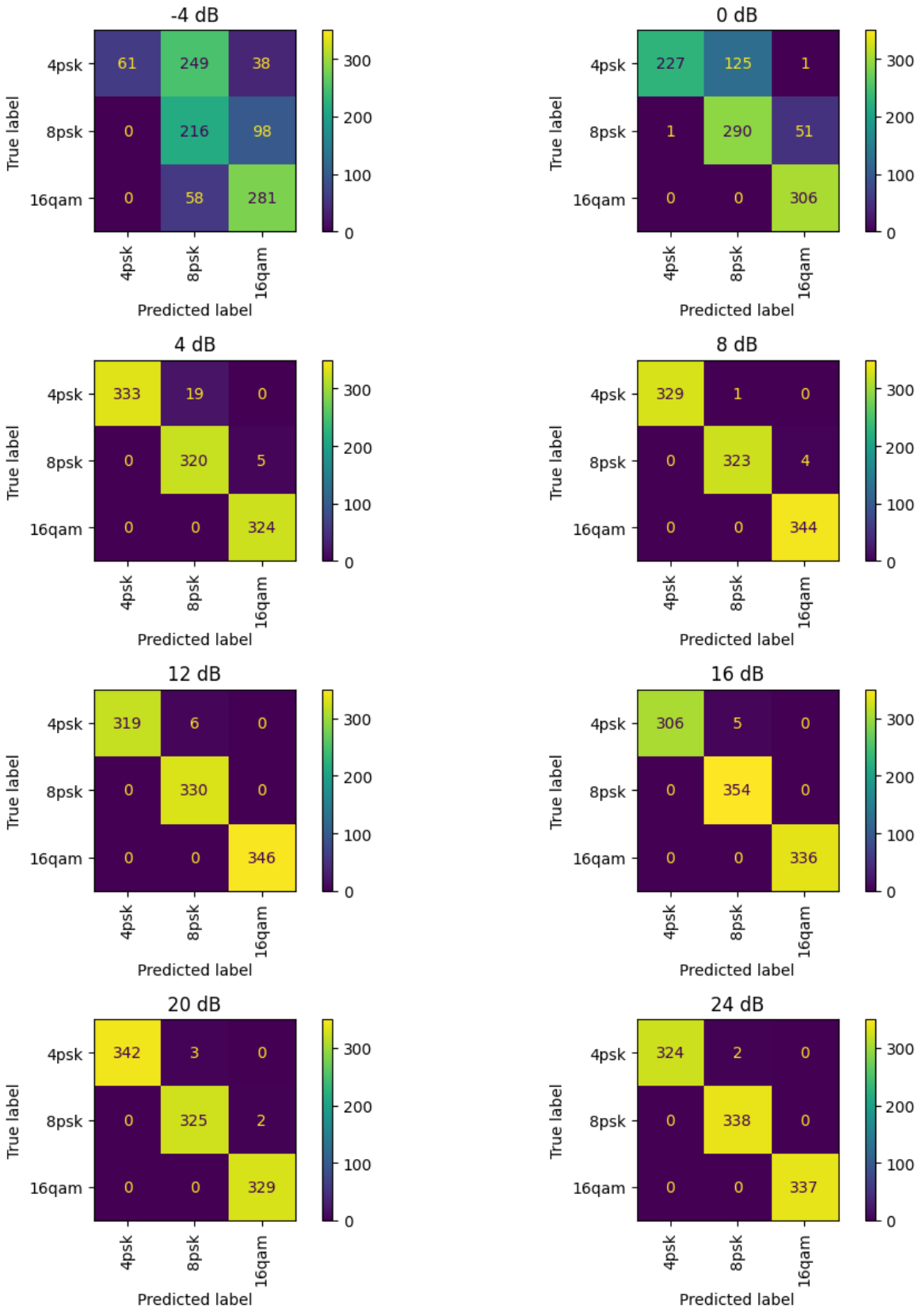

Figure 4.7: Confusion matrices for the alternative model.

The SNR versus accuracy plots of both models can be seen in Figure 4.8. As mentioned, the performance of the baseline model peaks at a relatively low SNR value, whereas the performance of the alternative model improves rapidly, achieving almost 100% accuracy from 4dB onwards.

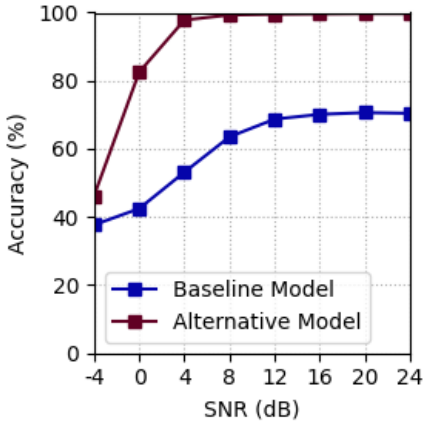

Figure 4.8: SNR versus accuracy curves.

Furthermore, ROC curves of the models are provided in Figure 4.9. Furthermore, ROC curves of the models are provided in Figure 4.9. Given that the ROC curves are used for analyzing binary classifiers, the classification problem is represented as a binary classification problem by employing a one-versus-rest scheme.

For a given TP rate, the corresponding FP rate of the model can be seen in the plots. The ideal point on the ROC plot is the top-left corner, $(0, 1)$. The ROC curve's closest point to the top-left corner represents the optimal decision threshold. It was previously stated that the baseline model added value by performing better than a random guess; however, it was not sufficiently performant. This observation is clearly illustrated in the ROC plots; the curve is above the chance line at all times; however, it does not get sufficiently close to the top-left corner of the plot. On the other hand, the improved alternative model performs better for each class at all admissible decision thresholds, as expected.

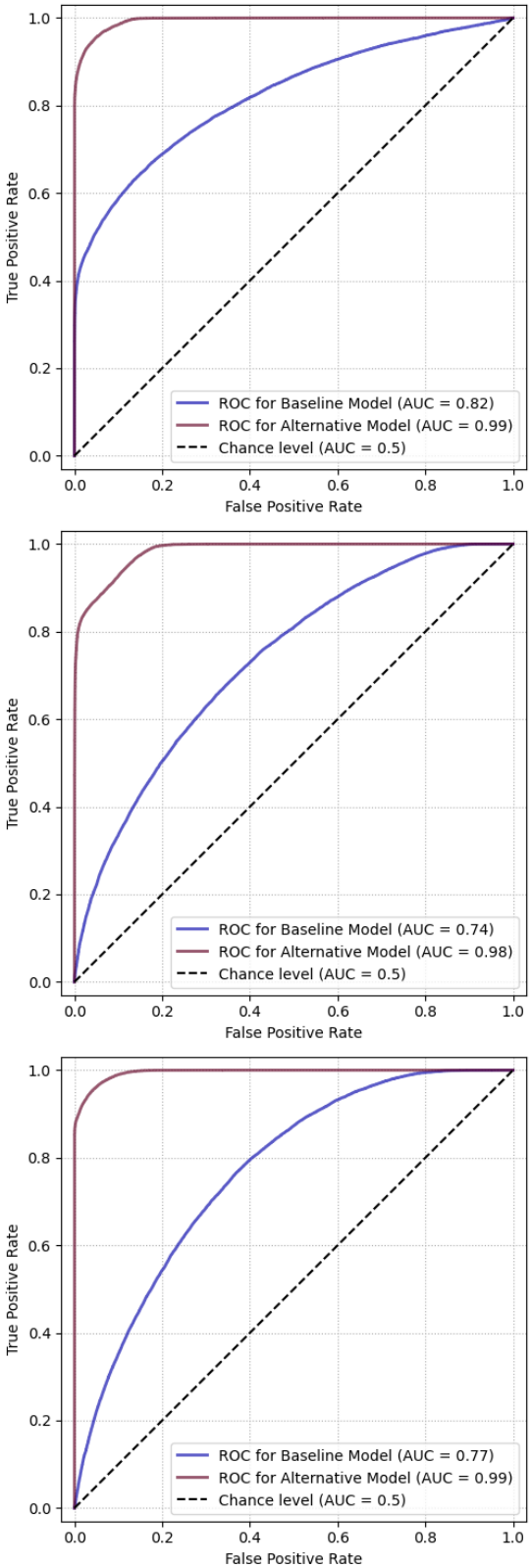

Figure 4.9: ROC curves using one-versus-rest scheme. QAM is given at top, 8PSK in the middle and 16QAM at the bottom.

5. CONCLUSION

With the increasing usage of wireless communications technologies, efficient utilization of the frequency spectrum has become a keystone in the digital communications field. AMC is a crucial element in improving the utilization of the frequency spectrum. Accurately identifying the modulation scheme enables systems to optimize their parameters accordingly and improve overall performance.

In this thesis, a chronological overview of the literature on AMC is provided. Various forms of likelihood-based methods, feature-based methods employing statistical and spectral features, machine learning, and deep learning methods are examined. Their emergence, strengths, and weaknesses are highlighted. Each solution served better under different conditions and must be selected accordingly.

The thesis focused on the blind AMC task. The only knowledge available at the receiver is the pool of the available modulation types. The methods proposed tried to eliminate high computational complexity, feature engineering, and impractical assumptions regarding the channel. The thesis proposes two similar but distinct architectures. Both architectures employ a filter bank-CNN complex to extract features from the received signal. The first architecture uses raw features to form a decision, effectively selecting the maximum activation for each filter bank-CNN complex for each class. The second proposed architecture improves the described process. It takes raw features and learns patterns relevant to the modulation type through an FNN. Then, it forms a decision using all the information together, forming a decision context. The FNN takes the concatenated outputs of the filter bank-CNN complexes, meaning the FNN can compare and contrast the features coming from different OFDM symbol lengths. It is observed that the decision context improves the classification performance significantly.

The first architecture adds value; however, the performance could be improved for theoretical and practical applications. It achieves 50% accuracy for three classes

with only 5943 trainable parameters on the out-of-sample test set, which illustrates the added value; it is significantly better than the 33% accuracy, corresponding to a random guess, yet insufficient. On the other hand, after analysing and updating the architecture accordingly, the second architecture performs much better. It achieves 91% accuracy for three classes with 10287 parameters on the same out-of-sample test set. The improvement illustrates the added value from the decision context concept.

There is still room for improvement. The proposed architectures contain relatively few trainable parameters compared to modern architectures. Enlarging the models might improve the accuracy of the second architecture. Furthermore, the thesis focused on only three OFDM symbol lengths and three modulation types. Expanding the scope of the model might be explored in the future.

REFERENCES

1. Poisel, R., *Foundations of Communications Electronic Warfare*, Artech House, 2008.

2. Huan, C.-Y. and A. Polydoros, "Likelihood methods for MPSK modulation classification", *IEEE Transactions on Communications*, Vol. 43, No. 2/3/4, pp. 1493–1504, 1995.

3. Sills, J. A., "Maximum-likelihood modulation classification for PSK/QAM", *MILCOM 1999. IEEE Military Communications. Conference Proceedings*, Vol. 1, pp. 217–220, IEEE, 1999.

4. Wei, W. and J. Mendel, "Maximum-likelihood classification for digital amplitude-phase modulations", *IEEE Transactions on Communications*, Vol. 48, No. 2, pp. 189–193, 2000.

5. Polydoros, A. and K. Kim, "On the detection and classification of quadrature digital modulations in broad-band noise", *IEEE Transactions on Communications*, Vol. 38, No. 8, pp. 1199–1211, 1990.

6. Hong, L. and K. Ho, "BPSK and QPSK modulation classification with unknown signal level", *MILCOM 2000 Proceedings. 21st Century Military Communications. Architectures and Technologies for Information Superiority*, Vol. 2, pp. 976–980, IEEE, 2000.

7. Panagiotou, P., A. Anastasopoulos and A. Polydoros, "Likelihood ratio tests for modulation classification", *MILCOM 2000 Proceedings. 21st Century Military Communications. Architectures and Technologies for Information Superiority*, Vol. 2, pp. 670–674 vol.2, 2000.

8. Massey, F. J., "The Kolmogorov-Smirnov Test for Goodness of Fit", *Journal of*

the American Statistical Association, Vol. 46, No. 253, pp. 68–78, 1951.

9. Wang, F. and X. Wang, "Fast and Robust Modulation Classification via Kolmogorov-Smirnov Test", *IEEE Transactions on Communications*, Vol. 58, No. 8, pp. 2324–2332, 2010.

10. Urriza, P., E. Rebeiz, P. Pawelczak and D. Cabric, "Computationally Efficient Modulation Level Classification Based on Probability Distribution Distance Functions", *IEEE Communications Letters*, Vol. 15, No. 5, pp. 476–478, 2011.

11. Zhu, Z., M. Waqar Aslam and A. K. Nandi, "Genetic algorithm optimized distribution sampling test for M-QAM modulation classification", *Signal Processing*, Vol. 94, pp. 264–277, 2014.

12. Cramér, H., "On the composition of elementary errors", *Scandinavian Actuarial Journal*, Vol. 1928, No. 1, pp. 13–74, 1928.

13. Mises, R. v., *Wahrscheinlichkeit Statistik und Wahrheit*, Vol. 7, Springer-Verlag, 2013.

14. Anderson, T. W., "On the Distribution of the Two-Sample Cramér-von Mises Criterion", *The Annals of Mathematical Statistics*, Vol. 33, No. 3, pp. 1148–1159, 1962.

15. Honda, C., I. Oka and S. Ata, "Signal detection and modulation classification using a goodness of fit test", *2012 International Symposium on Information Theory and its Applications*, pp. 180–183, 2012.

16. Anderson, T. W. and D. A. Darling, "A Test of Goodness of Fit", *Journal of the American Statistical Association*, Vol. 49, No. 268, pp. 765–769, 1954.

17. Azzouz, E. and A. Nandi, "Automatic identification of digital modulation types", *Signal Processing*, Vol. 47, No. 1, pp. 55–69, 1995.

18. Azzouz, E. and A. Nandi, "Procedure for automatic recognition of analogue and digital modulations", *IEE Proceedings - Communications*, Vol. 143, pp. 259–266, 1996.

19. Hong, L. and K. Ho, "BPSK and QPSK modulation classification with unknown signal level", *MILCOM 2000 Proceedings. 21st Century Military Communications. Architectures and Technologies for Information Superiority*, Vol. 2, pp. 976–980, 2000.

20. Ho, K., W. Prokopiw and Y. Chan, "Modulation identification by the wavelet transform", *Proceedings of MILCOM '95*, Vol. 2, pp. 886–890 vol.2, 1995.

21. Hassan, K., I. Dayoub, W. Hamouda and M. Berbineau, "Automatic modulation recognition using wavelet transform and neural networks in wireless systems", *EURASIP Journal on Advances in Signal Processing*, Vol. 2010, pp. 1–13, 2010.

22. Hipp, J. E., "Modulation Classification based on Statistical Moments", *MILCOM 1986 - IEEE Military Communications Conference: Communications-Computers: Teamed for the 90's*, Vol. 2, pp. 20.2.1–20.2.6, 1986.

23. Soliman, S. and S. Hsue, "Signal classification using statistical moments", *IEEE Transactions on Communications*, Vol. 40, No. 5, pp. 908–916, 1992.

24. Orlic, V. D. and M. L. Dukic, "Automatic modulation classification algorithm using higher-order cumulants under real-world channel conditions", *IEEE Communications Letters*, Vol. 13, No. 12, pp. 917–919, 2009.

25. Aslam, M., Z. Zhu and A. Nandi, "Automatic digital modulation classification using Genetic Programming with K-Nearest Neighbor", pp. 1731–1736, 12 2010.

26. Gang, H., L. Jiandong and L. Donghua, "Study of modulation recognition based on HOCs and SVM", *2004 IEEE 59th Vehicular Technology Conference*, Vol. 2, pp. 898–902, 2004.

27. Zhang, W., "Automatic modulation classification based on statistical features and Support Vector Machine", *2014 XXXIth URSI General Assembly and Scientific Symposium*, pp. 1–4, 2014.

28. Zhu, Z., A. K. Nandi and M. W. Aslam, "Robustness enhancement of distribution based binary discriminative features for modulation classification", *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2013.

29. Mendis, G. J., J. Wei and A. Madanayake, "Deep learning-based automated modulation classification for cognitive radio", *2016 IEEE International Conference on Communication Systems*, pp. 1–6, 2016.

30. Meng, F., P. Chen, L. Wu and X. Wang, "Automatic Modulation Classification: A Deep Learning Enabled Approach", *IEEE Transactions on Vehicular Technology*, Vol. 67, No. 11, pp. 10760–10772, 2018.

31. Wang, Y., M. Liu, J. Yang and G. Gui, "Data-Driven Deep Learning for Automatic Modulation Recognition in Cognitive Radios", *IEEE Transactions on Vehicular Technology*, Vol. 68, No. 4, pp. 4074–4077, 2019.

32. Peng, S., H. Jiang, H. Wang, H. Alwageed and Y.-D. Yao, "Modulation classification using convolutional Neural Network based deep learning model", *2017 26th Wireless and Optical Communication Conference*, pp. 1–5, 2017.

33. Huynh-The, T., C.-H. Hua, Q.-V. Pham and D.-S. Kim, "MCNet: An Efficient CNN Architecture for Robust Automatic Modulation Classification", *IEEE Communications Letters*, Vol. 24, No. 4, pp. 811–815, 2020.

34. Zheng, J. and Y. Lv, "Likelihood-Based Automatic Modulation Classification in OFDM With Index Modulation", *IEEE Transactions on Vehicular Technology*, Vol. 67, No. 9, pp. 8192–8204, 2018.

35. Marey, M. and H. Mostafa, "Turbo Modulation Identification Algorithm for OFDM Software-Defined Radios", *IEEE Communications Letters*, Vol. 25, No. 5, pp. 1707–1711, 2021.

36. Pambudi, A. D., S. Tjondronegoro and H. Wijanto, "Statistical properties proposed for blind classification OFDM modulation scheme", *2014 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology*, pp. 89–93, 2014.

37. Shimbo, D. and I. Oka, "A modulation classification using amplitude moments in OFDM systems", *2010 International Symposium On Information Theory and Its Applications*, pp. 288–293, 2010.

38. Zhang, J. and B. Li, "A New Modulation Identification Scheme for OFDM in Multipath Rayleigh Fading Channel", *2008 International Symposium on Computer Science and Computational Technology*, Vol. 1, pp. 793–796, 2008.

39. Chen, J., Y. Kuo and X. Liu, "Modulation identification for MIMO-OFDM signals", *2007 IET Conference on Wireless, Mobile and Sensor Networks*, pp. 1013–1016, 2007.

40. Pathy, A. K., A. Kumar, R. Gupta, S. Kumar and S. Majhi, "Design and Implementation of Blind Modulation Classification for Asynchronous MIMO-OFDM System", *IEEE Transactions on Instrumentation and Measurement*, Vol. 70, pp. 1–11, 2021.

41. Zhang, Y., G. Wu, J. Wang and Q. Tang, "Wireless signal classification based on high-order cumulants and machine learning", *2017 International Conference on Computer Technology, Electronics and Communication*, pp. 559–564, IEEE, 2017.

42. Al-Makhlasawy, R. M., M. M. A. Elnaby, H. A. El-Khobby and F. E. A. El-Samie, "Automatic modulation recognition in OFDM systems using cepstral analysis and a fuzzy logic interface", *2012 8th International Conference on Informatics and*

Systems, pp. CC–56–CC–62, 2012.

43. Hong, S., Y. Zhang, Y. Wang, H. Gu, G. Gui and H. Sari, "Deep Learning-Based Signal Modulation Identification in OFDM Systems", *IEEE Access*, Vol. 7, pp. 114631–114638, 2019.

44. Goldsmith, A., *Wireless Communications*, Cambridge University Press, 2005.

45. Cooley, J. W. and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series", *Mathematics of Computation*, Vol. 19, pp. 297–301, 1965.

46. Goodfellow, I., Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016.

47. Haykin, S. S., *Neural Networks and Learning Machines*, Third Edition, Pearson, 2011.

48. Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library", *arXiv Computing Research Repository (CoRR)*, 2019.

49. Kingma, D. P. and J. Ba, "Adam: A Method for Stochastic Optimization", *arXiv Computing Research Repository (CoRR)*, 2017.