Imperial College London
Department of Electrical and Computer Engineering

# Neural-Network-Aided Automatic Modulation Classification

Pavlos Triantaris

Supervised by Dr. Deniz Gündüz and Dr. Woon Hau Chin

A thesis submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy in Electrical and Electronic Engineering of Imperial College London
August 2021

# Abstract

Automatic modulation classification (AMC) is a pattern matching problem which significantly impacts divers telecommunication systems, with significant applications in military and civilian contexts alike. Although its appearance in the literature is far from novel, recent developments in machine learning technologies have triggered an increased interest in this area of research.

In the first part of this thesis, an AMC system is studied where, in addition to the typical point-to-point setup of one receiver and one transmitter, a second transmitter is also present, which is considered an interfering device. A convolutional neural network (CNN) is used for classification. In addition to studying the effect of interference strength, we propose a modification attempting to leverage some of the debilitating results of interference, and also study the effect of signal quantisation upon classification performance.

Consequently, we assess a cooperative setting of AMC, namely one where the receiver features multiple antennas, and receives different versions of the same signal from the single-antenna transmitter. Through the combination of data from different antennas, it is evidenced that this cooperative approach leads to notable performance improvements over the established baseline.

Finally, the cooperative scenario is expanded to a more complicated setting, where a realistic geographic distribution of four receiving nodes is modelled, and furthermore, the decision-making mechanism with regard to the identity of a signal resides in a fusion centre independent of the receivers, connected to them over finite-bandwidth backhaul links. In addition to the common concerns over classification accuracy and inference time, data reduction methods of various types (including "trained" lossy compression) are implemented with the objective of minimising the data load placed upon the backhaul links.

## Declaration of Originality

I hereby certify that this thesis is the result of my own work under the guidance of my Ph.D. supervisors, Dr. Deniz Gündüz, Dr. Filippo Tosato, Dr. Evgeny Tsimbalo, and Dr. Woon Hau Chin, and that any ideas or quotations from the work of other people are appropriately referenced.

Imperial College London
London, United Kingdom
10 August 2021

Pavlos Triantaris

# Copyright Declaration

# Acknowledgements

It is meet and right to express my gratitude to all them that made this journey of four years in research possible, and contributed thereto throughout those years:

- My supervisor at Imperial College, Dr. Deniz Gündüz, as well as my industrial supervisors at Toshiba Research Europe Ltd., Dr. Filippo Tosato, Dr. Evgeny Tsimbalo, and Dr. Woon Hau Chin. From my meek first steps into the world of research, and still until now, they have been a source of constant inspiration and guidance, and have guided me through the long and arduous process of completing each step towards completing the PhD course with immense patience and understanding. Additionally, it should be mentioned that, in both institutions mentioned above, I was privileged to have been part of highly competent research teams, and to have been granted the opportunity to learn from their admirable example.

- The Sustainable Cellular Networks Harvesting Ambient Energy project (SCAVENGE), of which I was a part along with thirteen other early-stage researchers from all around Europe. All the people involved in the project provided for each other an inspiring and friendly research environment, and pleasant company overall, while through our meetings over the course of the project, we were lucky enough to visit and experience some of the most beautiful cities of the continent. Additionally, I acknowledge the generous financial support which I received from European Union‘s H2020 research and innovation programme under the Marie Sklodowska-Curie Action SCAVENGE (grant agreement no. 675891).

- The parish churches of Chingford and Brentwood, not only for offering me the opportunity to serve as their organ scholar and to enjoy even some extensive cathedral visits, but also for providing their selfless and wise spiritual guidance and support throughout the recent years, especially a particularly daunting year of strict lockdowns. My particular thanks to Fr. Matthew Austen of Brentwood, and Mr. Michael Emerson FRCO of Chingford.

- The Bristol and District Organists' Association, which provided a most enjoyable and fun extracurricular schedule, with ample doses of English countryside beauty, during my

placement in Bristol; particularly Mr. Chris Mansfield and Mr. Adrian McInnes.

- My family and friends, whether they be based in Greece or in the UK, for their continual support in these years when I was called to carry out demanding research work in a country which was completely new to me, and for all the fun, travels, and relaxation which we have shared. My deepest graditude especially to Mr. Chris Paraskevopoulos LRSM, whom I have the honour of calling my mentor.

- Last, and certainly far from least, my wife, who has always been at the forefront of supporting me even when things got tough and I thought I had hit rock bottom, always an inspiration and a welcoming shoulder to cry on. Cheers to our next adventure, "for richer or for poorer"!

# Dedication

To the memory of my father.

Who sayes that fictions onely and false hair
Become a verse? Is there in truth no beauty?
Is all good structure in a winding stair?
May no lines passe, except they do their dutie
Not to a true, but painted chair?

Is it no verse, except enchanted groves
And sudden arbours shadow course-spunne lines?
Must purling streams refresh a lovers loves?
Must all be vail'd, while he that reades, divines,
Catching the sense at two removes?

Shepherds are honest people; let them sing:
Riddle who list, for me, and pull for Prime:
I envie no mans nightingale or spring;
Nor let them punish me with losse of rime,
Who plainly say, *My God, My King.*

*George Herbert, "Jordan (I)"*

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Signal Identification - Problem Statement

Physical-layer signal identification, a subset of the methods of device identification, is a scientific field which has commanded varying degrees of interest in communications literature throughout the years, and has been marked by an increased focus in recent times.

In light of the latest developments concerning 5G networks, one of the pressing factors which must be taken into account when designing hardware and software for them are the presence of multiple Radio Access Technologies (RATs) and the integration of many different categories of devices into a greater, more unified framework [BCP+03]. As such, the technology of identification and monitoring of active devices within a frequency spectrum becomes a highly desirable goal for a modern communication network due to its ability to offer:

1. Reduction of latency for devices with limited capabilities (e.g. RFID tags and machine-type devices) or low latency requirements
2. Reduction of power consumption for devices such as the aforementioned ones
3. Optimisation of network resources, e.g. by means of interference classification
4. Accurate tracking of device activity

5. Enhanced network security

Representative examples worth mentioning are certain techniques of spectrum allocation and intrusion detection [DZC12], which can benefit any communicating element (e.g. a base station) of a network. The importance of these technologies, however, becomes especially pronounced with multi-RAT environments in mind, considering e.g. the development of unlicensed spectrum usage (with WiFi, LAA [CCGZ17] and other methods).

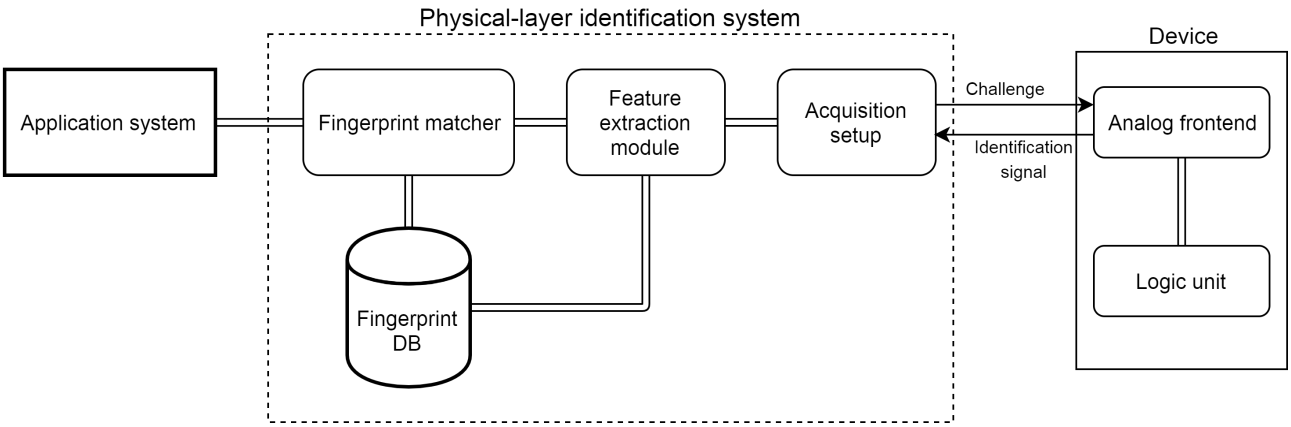



Figure 1.1: Generalised visualisation of a fingerprint-based device identification mechanism.

The identification of wireless devices and their emissions or other activity, as well as the scanning of a spectrum for signals, as traditionally practised in various deployments, follows an approach relying on predetermined data, unique to each communicating device, thus mainly employing methods such as public identifier sharing or secret keys, "handshake" procedures etc. Problems in these methods, however, are expected to arise as a result of the imposition of latency or power restrictions, as well as the great increase of device types cooperating and coexisting in a given spectrum. By contrast, an alternative family of methods is named fingerprinting, of which the key characteristic is leveraging features which can be inferred from the signal of a device under recognition, thus relocating the area of interest to the physical layer instead of higher ones [BCP+03][DZC12]. Fig. 1.1 depicts a generalised form of a wireless device identification system based on fingerprinting.

It should be noted here, for the sake of clarity, that the term "device identification" should not be interpreted as to strictly denote methods of detecting the precise identity of a given communications device (that sub-category of methods is known as specific emitter identification

(SEI)). Consequently, many research works dealing with device ID, including this thesis, are concerned with other sub-areas of the same.

In the context of signal identification (both radio-frequency (RF) and otherwise), the areas of application which have been recognised even in earlier literature [DZC12] include the following (with representative examples):

- Intrusion detection (Hall et al. [HBK04] propose the classification of amplitude, phase, and frequency features from transients, arguing that physical, non-malleable characteristics of transceivers are manifest in the transient and are thus difficult to impersonate)
- Access control (Brik et al. [BBGO08] rely on similar presuppositions about physical impairments, their algorithm based mainly on error metrics such as offsets)
- Malfunction detection (Wang et al. [WOA05] classify defects in physical transmission devices by extracting feature vectors from wavelet decompositions of acoustic signals)
- Secure localisation (Tippenhauer et al. [TRPC09] assess the sensitivity of the Skyhook positioning system to impersonation attacks and propose fingerprinting methods as a way of rendering such systems more robust)
- Wormhole detection (Rasmussen and Capkun [RC07] bring attention to the great difficulty with which wormhole attacks can be dealt with, and suggest the usage of physical-layer fingerprints for an additional security check, as these are hard to replicate)

With regard to RF applications of signal identification, two of the most prominent sub-categories of this field are automatic modulation classification (AMC) and specific emitter identification (SEI).

Modulation classification refers to the process of the accurate detection of the modulation scheme of a given telecommunication signal, and constitutes part of a communication system flowchart between the processes of signal detection and demodulation (a symbolic representation is offered in the flowchart of Fig. 1.2). Although military applications have traditionally acted

as the main catalyst for research in this area, civilian applications also constitute an expanding area of interest, as we shall explain further ahead in this chapter [DABNS07] [ZHC16].

By way of example, one may mention that AMC can aid in discerning the source of a detected signal, which is a desirable feature for autonomous wireless systems (e.g. Internet of Things (IoT) setups). Likewise, a modulation detector is a component of high importance for software-defined radio (SDR) setups, for the sake of fast and intelligent adaptation to spectrum alterations. From the standpoint of military contexts, on the other hand, a potential area of application is the creation of jamming transmissions within a recognised modulation type, as well as the decoding of intercepted signals [RJY+19].

A reason which has led to increased interest in AMC in recent years is to be located in the planning and structuring of Beyond-5G (B5G), where the popularisation of massive multiple-input-multiple-output (MIMO) systems, among other factors, will lead to the complication of an already heavily crowded electromagnetic spectrum. Furthermore, even under current 5G deployments, the presence of multipath fading increases the difficulty of the signal recognition process [HGKL20].

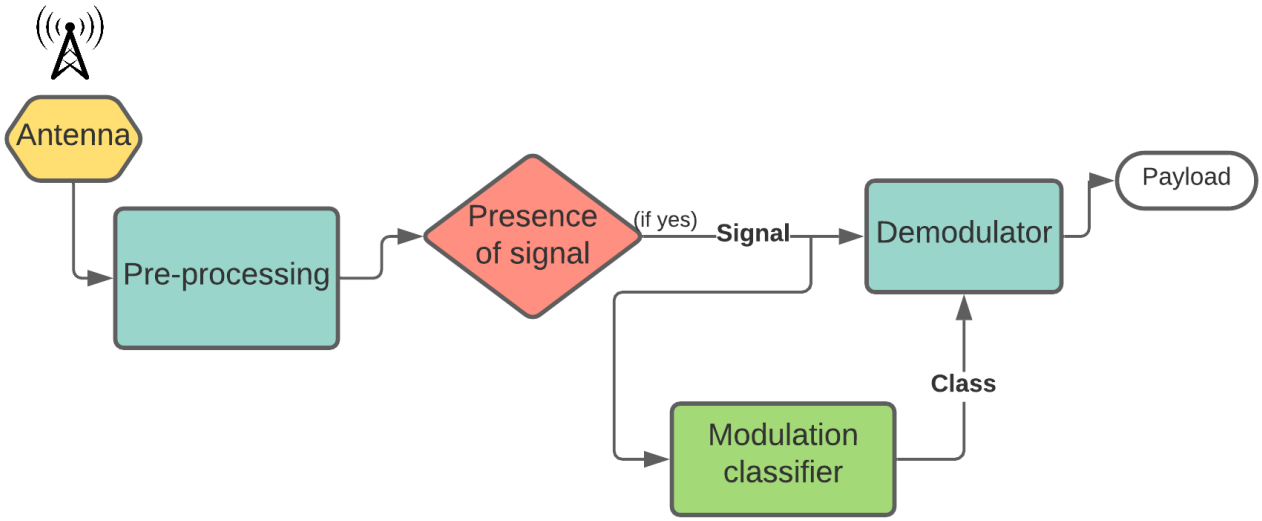



Figure 1.2: Function of AMC in a telecommunication setup.

Traditional approaches to the problem of AMC can be claimed to largely belong to one of two categories of methods: likelihood-based (LB) and feature-based (FB). The former consists of methods where an analytical expression of different signal classes is used to compute

thresholds, and a decision is made based on comparison against the same; while in the latter category, the quantities utilised for reaching a decision are statistical features computed from signals [RJY+19]. Problems and drawbacks arising from both methods, however, have led to the popularisation of a third family of methods, namely those relying on artificial neural networks (ANNs).

While methods involving ANNs may, in certain senses, be considered a sub-category of FB, its proliferation has certainly been catalysed by the rapid development from which neural network technology has benefitted in recent years, with an increasing number of applications in wireless network scenarios [CCS+19]. A most important key feature of this newly-emerging direction of signal ID research lies in its data-driven nature; in particular, while traditional approaches would typically consist e.g. of estimation of features, determining of thresholds, and classification based on the same, neural nets bear the extra requirement of a training stage, which, in turn, requires the presence of potent datasets which are to be used for parameter and hyperparameter tuning and ought to be separate from data used for evaluation.

Our experimentation for the purposes of this work has been conducted exclusively with usage of neural network technology, in such a way that no reliance on aspects of earlier methods is necessary. That is, using a given signal as raw input, we attempt to solve problems of AMC while relegating the process of analysis and feature extraction to the inner workings of the neural classifier itself. In addition, the inclusion of auxiliary information inputs (where they are available) is also considered as a possibility for improving the classifier performance.

## 1.2 Common Challenges

### Datasets

RadioML, as described in [OC16], is one of the most commonly used datasets for purposes of evaluating deep-learning-based AMC setups [TBLS19][RJY+19]. Benefitting from the ability to leverage great numbers of samples (in the order of hundreds of thousands, potentially millions),

it has lead to improvements in the performance for AMC methods through the creation of very large training and testing datasets. Furthermore, the proliferation of relatively small numbers of comprehensive datasets such as RadioML also creates a degree of standardisation, which in turns allows for the comparison of different approaches and the creation of benchmarks.

Most of the experiments outlined in the current work use RadioML to a limited degree, and have mostly relied on locally and newly created datasets with smaller numbers of frames both for the training and testing phases. The foremost reason which led to this decision is that RadioML consists of signals recorded at the receiving end, which means that they already bear the influence of channel effects such as noise and fading; as such, they do not allow for any change in the propagation parameters, and cannot be adapted e.g. to multi-user environments, which constitute one of the main areas of interest of this work.

### Multi-User Environments

As the majority of works proposing AMC solutions are focused mainly upon the optimisation of classifiers and preprocessing methods [DABNS07][RJY+19][MCWW18], the physical setup is assumed to be a point-to-point system, i.e., consisting of one transmitter and one receiver only, without any other devices being present. This assumption, however, is not always justifiable in actual wireless telecommunication systems.

As a result of the above, a decision was made that multi-user environments would factor in our work in two ways:

- Firstly, in a case where the point-to-point system is disrupted by the presence of an interferer.
- Then, in cases where data from multiple receiving entities can be used to improve performance.

### Operation Times

In the context of the development of 5G and B5G technologies, one of the most significant challenges has been the improvement of overall quality of service (QoS) in terms of operating speed, with low-latency services being considered essential in most applications. By way of example, massive MIMO and unmanned aerial vehicle (UAV) technologies are projected to rely extensively upon fast real-time modulation classification systems [HGKL20]. It has been observed, however, that the majority of works concerned with AMC are primarily focused upon the optimisation of classification accuracy, and do not include an assessment of running time as a deployment issue.

Throughout the later phases of the experimentation pertaining to the present work, the speed of the designed algorithms has been assessed as well as the raw effectiveness at correct classification. As a result of that, an additional theme which has accompanied our research has been the incorporation of data reduction techniques (such as quantisation) in the interest of reducing inference time, as well as the impact of each technique upon the performance of algorithms in terms of inference accuracy.

### Burdens on Data Transfer

In addition to operation times, another dimension which is directly affected by data volume in AMC contexts has to do with bandwidth consumption and data transfer in limited telecommunication channels. Especially in cooperative, multiple-receiver cases, typical setups can be considered in which the processing node is not also the receiver, but is instead connected to the receiving devices via connections of a potentially limited capacity. As such, a further issue worth studying which arises from the above observation has been the search for optimal solutions with regard to the volume and type of data communicated between different nodes of a distributed classification setting, as well as the trade-offs between bandwidth consumption and final performance.

### 1.2.1 Objectives of thesis

The overarching purpose of the work described in this thesis has been the identification of potentially promising areas of research pertaining to signal recognition, and pursuit of enhancements to the state of the art. The particular emphasis was upon automatic modulation classification (let it be noted that assessments of other kinds of signal recognition, such as SEI, were not pursued in the context of this work), thus it constituted an area of focus through which this research was carried out. More specifically, all of the aforementioned challenges served as incentives behind the research described here, so we have sought to tackle problems of AMC in ways which seek to:

- Benefit from smaller but comprehensive datasets.
- Acknowledge multi-user setups and their nuances.
- Improve upon baseline operation times for AMC algorithms.
- Take the issue of data transfer between nodes into account, and implement compression techniques with the aim of efficiently reducing data transfer volumes.

## 1.3 Thesis Overview

### Modulation Classification in the Presence of Interference

The earliest concept detailed in the present work considers a telecommunication system where a simple point-to-point transmission between two (single-antenna) devices is negatively impacted by the presence of a third transmitter, which by its activity constitutes a further undesired influence upon the reception (in addition to Gaussian noise and other environmental factors).

As described in Chapter 3, a convolutional neural classifier is trained on the part of the receiving device, and the effect of the interference is considered an additional factor during the analysis of results. Concurrently, in a parallel experiment, the class of the secondary signal is considered

available to the classifier, and a case is made that the incorporation of information on the identity of that signal into the neural net is possible in such a way that it leads to performance improvements.

### Cooperative Model for AMC in a Single-Input-Multiple-Output Setting

This is the first instance during our experiments where a cooperative approach to AMC was proposed, and is detailed in Chapter 4. In particular, it is assumed during the creation of a new, independent AMC dataset on MATLAB that transmission occurs over a single-input-multiple-output (SIMO) channel, and the end device receives different versions of the signal on a number of multiple antennas. Consequently, classifiers are trained with the number of participating receivers as an additional modifiable parameter, and the usefulness of leveraging multiple available versions of a given signal as a way of ensuring better detection accuracy is demonstrated.

In this segment, training and evaluation times are included in all iterations of the experiment. With this factor being taken into consideration, data compression methods are introduced as a way of reducing the intervals required for signal class detection, and the trade-off between speed and accuracy is explored.

### AMC in Geographically Distributed Setups

The approximation of real-world scenarios and environments for signal identification applications is pursued to an increased degree in the concluding part of this current work (Chapter 5). More specifically, the setup under consideration consists of four receiving stations positioned at the vertices of a square covering an expanded geographical area, and a transmitter which moves at a non-zero velocity and might be located anywhere within the square area.

Unlike the previous cooperative experiment, however, it is not necessarily considered a given

that the node hosting the neural classifier always receives all the data in their complete form; on the contrary, it is assumed that the receivers transmit information to the node in question over a limited communication channel, thus the data offered for classification are, in most cases, not available in the form of the full version of a received signal. In this process, various forms of compression and dimensionality reduction are assessed, and their performance (in terms both of accuracy and running time) compared to an ideal scenario. Experiments in this framework include end-to-end training.

## 1.4 Statement of Originality

I, Pavlos Triantaris, hereby declare that the thesis titled "Neural-Network-Aided Automatic Modulation Classification" is my own work, as is the research work presented therein. I confirm the following:

- This work was done wholly or mainly while in candidature for the degree of Doctor of Philosophy at Imperial College of Science, Technology and Medicine.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

# Chapter 2

# Background

In this chapter, we aim to present device ID and automatic modulation classification both in their historic development and in their state-of-the-art context. This, in turn, shall help clarify some of the motivations and inspiration behind the current work.

## 2.1 Physical-Layer Device Identification

Automatic modulation classification forms part of the wider family of methods of physical-layer device identification.

The identification of wireless devices and their activity, as traditionally practised in various deployments, would follow an approach relying on predetermined data, unique to each communicating device, thus mainly employing methods such as public identifier sharing or "handshakes". These methods, however, may become quite cumbersome when restrictions of latency and power consumption are imposed, and a multitude of types of devices is present.

By contrast, an alternative, more recently proposed family of methods is called **fingerprinting**, and leverages not features prescribed to a device in higher levels, but instead relocates the focus onto the behaviour of devices with regard to the physical layer [BCP+03] [DZC12]. More specifically, the idea for a physical-layer approach was primarily based upon the knowledge that

each device is characterised by a practically unique set of hardware imperfections (impairments) which are manifest in the transmitted signal as minor deviations from the ideal waveform, such as carrier frequency offset, I/Q imbalances and phase noise.

The 2012 work by Danev et al. [DZC12] is a seminal paper which aims to familiarise researchers with the area of PHY-layer identification of devices, provide a general framework with common aspects and potential issues, and present a comprehensive list of "families" of techniques used within that framework at the time. Therein it is demonstrated that, at its core, every PHY-ID system can be reduced to three components which are the main *identification system*, an *application* to which identification information is forwarded, and naturally, the *wireless communication device* under consideration. Furthermore, the identification system itself can usually be broken down into three separate components, which are the *signal acquisition setup* (which ought to acquire and digitalise the input signals without adding noise or otherwise degrading them), the *feature extraction module* (also responsible for storing the fingerprints in a database), and a *fingerprint matcher* (which returns information regarding the active device by accessing the already existing database); or generalised versions thereof (refer again to Fig. 1.1).

Features extracted from a signal can be either predefined (*i.e.*, directly measurable characteristics of the signal) or inferred (*i.e.* when they are extracted from an intermediate representation of the signal, such as a Discrete Wavelet Transform, or the aforementioned representation itself is considered a feature map). In the former case, the feature extraction function directly translates an input into a set of values, whereas in the latter it acts as a reducer or expander of dimensionality.

With regard, e.g., to SEI, hardware imperfections (impairments) are leveraged under the assumption that they are unique to each device, and thus a fingerprint might consist of their manifestations in the signal, such as:

- Carrier frequency offset.
- I/Q imbalances (e.g. in PSK/QAM modulations).
- Phase noise.

- Nonlinearities from power amplifiers

The need for automatic intelligent decision-making is inherent in the design of a PHY-ID, in that there exists (almost inevitably) a processor and matcher of fingerprints which reaches a final decision regarding the nature of a signal. This introduces the need for techniques ranging from simple decision trees to higher-order machine learning (with the latter spanning from k-Nearest Neighbour and Support Vector Machine (SVM) algorithms [BBGO08] to more sophisticated ML techniques such as artificial neural networks (ANNs) [OWVC17a][WJV+15]).

## Literature on PHY-ID

Physical-layer identification in general, and modulation recognition in particular, constitutes a field already spanning several decades, but has underwent major developments and garnered substantial interest in the past decade, not least due to the evolution of both hardware and software technologies (e.g., the proliferation of graphics processing units (GPUs) and frameworks such as TensorFlow).

A representative, if early, work on the recognition of electromagnetic devices based on physical-layer characteristics is the paper of Remley et al. [RGJ+05], where WLAN cards of different manufacturers are tested. The approach presented here leverages both frequency- and time-domain characteristics of a signal, without necessarily employing any sort of classification algorithm. By way of example, it is noticed that characteristics such as waveform smoothness and null depth (in the time domain) or periodogram symmetricity differ according to the origin/manufacturer of the card under test. As such, this can be considered a typical early example of a feature-based method of device identification.

Another work in the same category is by Polak and Goeckel [PG14], where the approach was motivated by an interest in enhancing the security of wide-area networks (WANs). Focusing upon the behaviour of phase locked loops (PLLs) of wireless transmitters, a PLL's phase noise is considered as a possible source of a PHY-layer fingerprint, as the former is caused solely by physical variations and impairments in the separate electronic components. In short, the

analysis models a PLL via an autocorrelation function, and consequently a set of three values depending on internal parameters is chosen as a feature vector. The authors then proceed to devise a probabilistic comparative algorithm based on the envelopes of estimated discrete autocorrelation functions computed from captured PLL output records.

Cobb et al. [CGT+10] propose a method named RF distinct native attribute fingerprinting as a system of identity assessment for embedded devices based on the passive observation of collateral radio-frequency (RF) emissions. The feature extraction process there consists of the calculation of a 4-value vector of statistical features over different areas of the signal, with regard to its instantaneous phase, frequency, and amplitude, so that the actual fingerprint vector is the result of their concatenation. The implemented classification tools are multiple discriminant analysis and maximum likelihood classifiers, in conjunction with Monte Carlo simulation for the assessment of the influence of different signal-to-noise ratio (SNR) values upon the effectiveness of the algorithm.

A more recent work, not primarily focused on the fingerprinting aspect, is the article of O'Shea et al. [OWVC17b], which focuses on sparse representation of radio signals and also predicts more recent trends in DL-based signal identification. More specifically, the method presented there attempts to project input signals onto a space where greater or smaller similarity of different inputs is ultimately translated into smaller and greater spatial differences respectively. Eschewing the usual practices of principal component analysis and independent component analysis for sparse representation tasks, the algorithm is instead implemented with the aid of a convolutional neural network (CNN) architecture with nonlinear features. While the CNN structure is trained to function as a classifier and trained accordingly, the desired representation itself is extracted from one of the intermediate outputs of the multilayered function.

## 2.2 Automatic Modulation Classification

Within the wider research area of PHY-ID, automatic modulation classification (AMC) in particular has proven itself useful in a variety of ways, of which some example are the rein-

forcement of situational awareness in software-defined radio systems, as well as communication surveillance [ZYWZ21]. Additionally, spectrum sensing and interference identification constitute important areas of practical application. A key driving factor for the development of AMC has always been found in military applications (e.g. for decoding of intercepted signals, creation of jamming signals in a recognised modulation type), but numerous civilian deployments also exist (e.g. deployments of adaptive modulation and coding) [ZN15][ZYWZ21].

Two broad families of methods which can be distinguished within the broader scope of AMC are likelihood-based (LB) [WJV+15][HDCP10] and feature-based (FB) methods [KKC+16][JPO+18]; their fundamental difference consisting in the fact that the recognition algorithms operate upon probability distributions estimated from the analytical signal expressions in the former case, and upon features extracted from a signal in the latter. A strong incentive behind the decision to focus upon FB experiments has been their advantage of offering near-optimal performance along with practically acceptable deployment times, whereas the fact remains that LB methods, while optimal in the Bayesian sense, incur a burdensome cost of high computational expenses and also do not adapt well to factors such as real channel imperfections and hardware impairments [DABNS07].

Conventional FB approaches for AMC (and time-series classification in general) commonly rely on *a priori* defined metrics (these can range from simple statistical quantities, such as variance or signal power, to higher-order moments and cumulants), with a form of concatenation of the same acting as a representation of each individual signal; the problem, in such a case, consists in determining an optimal selection of features, as well as the spatial grouping thereof, where the latter is, in most cases, achieved through the use of comparative likelihood algorithms, machine learning (ML) methods (e.g. K-nearest neighbour, support vector machines), or an ensemble combination of such methods. Since, however, this course of action necessitates difficult and potentially computationally expensive procedures of data pre-processing and optimal feature choice (incl. for a wide array of sub-ideal conditions), artificial neural networks have been proposed and used as a workaround method (in some cases even considered as a separate third category, distinct from LB and FB) [OWVC17a][OH17a][KKMP17].

In this work, we first sought to extend the scope of such methods to AMC in the presence of interference, seeing how cognitive telecommunication is becoming particularly important with the increasing number of wireless devices sharing the same spectrum in 5G networks. In addition to spectrum sensing aimed at seeking spectrum holes for interference-free communications [ZLHZ10], more advanced cognitive techniques would allow interference identification and cancellation to improve the rate and reliability of communication [WLY12]. Similarly to [OWVC17a],[OH17a],[KKMP17] we will primarily be relying on the implementation of a deep convolutional neural network for AMC, and evaluate its performance in the presence of interference.

### 2.2.1 AMC - Problem Statement

Assuming the existence of a single transmitter, and potentially multiple receivers, in a given telecommunication environment, the baseband form of the received signal can be described as such:

$$\mathbf{r}(t) = A\mathbf{h}^T \sum_{k=0}^{K-1} x_k^m p(t-(k+c)T_s) + \mathbf{A}_n w(t) \tag{2.1}$$

Where $A$ is the amplitude which models transmission power, $\mathbf{h}$ is a vector comprising of (Rician- or Rayleigh-based) coefficients for different channels, $x_k^m$ is the k-th symbol out of K which are to be transmitted during a given time slot, drawn from the constellations of modulation scheme $m$, $p(t)$ is the pulse-shaping function, $c$ is the timing offset, and $w(t)$ is additive white Gaussian noise (with $\mathbf{A}_n$ being an amplitude vector modelling noise power at different receivers).

The goal of a modulation classification algorithm is to be able to solve the following generalised optimisation problem for any given $\mathbf{r}(t)$:

$$\arg\min_{\tilde{m}} \mathcal{L}(\mathbf{r}(t); F(\theta, \tilde{m})) \tag{2.2}$$

Where $\mathcal{L}$ is an objective function which ought to be minimised, and $F$ is a signal processing/transformation unit with parameters $\theta$. As already mentioned before, it is typical for $F$ to assume the form of an ANN, parametrised by its weights, biases, and activations, while in general it might include certain pre-processing steps (which are, as a rule, not tunable).

Commonly, works concerning AMC are focused on devising expressions of $F$, and methods of training $\theta$, aimed at optimising performance accuracy or operation time. In our case, however, we mainly worked with a largely constant architecture and training method (albeit with some novelties in certain cases) which were evidenced to exhibit optimal behaviour when operating upon our datasets, and assessed how realistic data augmentations and preprocessing affect accuracy and speed of ANN-based AMC.

## 2.3 Machine Learning for the Physical Layer

In conjunction with the aforementioned challenge of the presence of multiple RATs, 5G communication networks also face further hurdles in that self-sufficiency and intelligence are required of them in multiple forms, including optimal spectral and energy efficiency, service learning and "smart" transmission protocols. As a result of this, the family of methods know as machine learning has been proposed as a promising solution for the expansion of the functionality of communication networks, since it can greatly assist any existing framework in intelligent decision making, database organisation and adaptive learning [JZR+17].

Generally defined, a machine learns the execution of a particular task $\mathbb{T}$, with the goal of maintaining a specific performance metric $\mathbb{P}$, based on a particular experience $\mathbb{E}$, where the system aims to reliably improve its performance $\mathbb{P}$ while executing $\mathbb{T}$, again by leveraging $\mathbb{E}$. Thus, in a sense, a machine acts within a system as a pseudo-intelligent agent facilitating

automation and self-dependency.

These tasks can be broken down into further sub-categories according to their functionality and structure. A non-exhaustive list would include regression algorithms (such as support vector machines [MW01] and $k$-nearest neighbour algorithms [Pet09]), decision trees, Bayesian procedures, clustering, and deep learning (which includes artificial neural networks). Figure 2.1 presents a concise picture of the process of setting up and tuning a machine learning process.

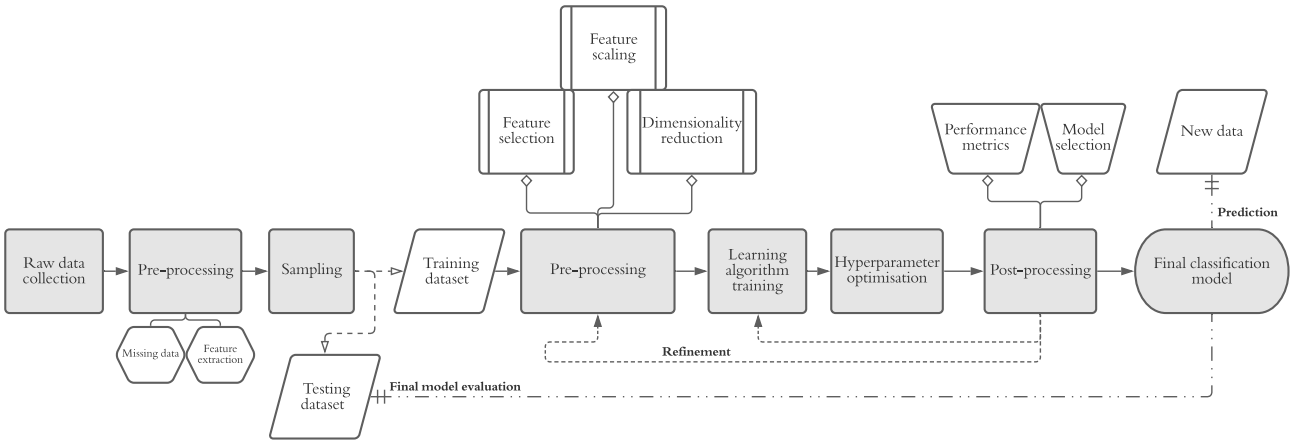



Figure 2.1: Supervised machine learning model setup

Accordingly, some of the factors which would allow us to expect machine learning to exhibit robust, possibly highly competitive performance and usefulness in the contexts considered are the following:

- Real communication systems are characterised by numerous imperfections and non-linearities, an aspect which makes the production of tractable models and methods based on them especially difficult; whereas DL systems can be easily optimised for different architectures and environments.
- Artificial neural networks in particular exhibit great capacity for algorithmic learning and function approximation, and their execution can be highly parallelised, leading to faster execution times and lower energy requirements.
- State-of-the art DL techniques are capable of efficient resource utilisation on massively parallel architectures like GPUs, with little requirements of application-specific modification.

### Literature on ML for the Physical Layer

The work of K. Tsagkaris et al. [TKD08] is a seminal paper concerned with the proper management of the electromagnetic radio spectrum and envisioning the further usage of neural networks in communication systems. The paper is mainly focused on the function of cognitive radio (CR) , which by definition involves intelligent decision-making processes. Any CR is equipped with a storage base of prior information and knowledge, along with a reasoning engine which determines the course of action depending on the measured state of the environment; thus a third part, a learning engine, is proposed as a way of reducing computational complexity and time (modulation classification is mentioned as an important application). In the experiments described in the paper, a typical neural network (NN) configuration is presented where the input is a time-series of measured data rates and the output is a prediction of the data rate achievable by a given IEEE WLAN 802.11g CR configuration.

Also with a focus on cognitive radio is the work of Tumuluru et al. [TWN10]. Presupposing the existence of primary and secondary users within a cognitive radio network (CRN), a reliable spectrum sensing mechanism is deemed necessary at the end of a secondary user, so that interference to the primary ones is minimised. Earlier proposed spectrum predictors were based upon hidden Markov models (HMMs), but the computational complexity and memory space required rendered them impractical, in addition to their requiring continuous training. On the other hand, NNs significantly reduce complexity after their training. The models proposed here succeed at characterising a channel as busy or idle (in regard to activity of a primary user) based on the output of a multilayer perceptron (MLP) which accepts as an input a vector of features describing the channel.

O'Shea et al. in [OCM16] introduce a NN-based method for anomaly detection, also noticing that past methods of addressing the problem have relied on expert features or environmental factors. The main contribution is a scheme based on a neural network as a time-series predictor producing a distribution which is then compared to metrics from anomaly-free reference signals. Various architectures are used for training, including dense, convolutional and long short-term memory (LSTM) networks.

The work of Gandetto et al. [GGR04] takes into account software radio (SR) technology, where RF aspects and transmission/reception functions are defined as software processes. Recognising mode identification (MI) as a significant challenge in modern SDR applications, it is stated that future signal processing techniques for RF applications should be able to identify all present communication modes in a post-A/D version of the incoming electromagnetic signal. The feature selection is based on a time-frequency representation of the signal, via the Wigner-Ville and Choi-Williams transforms and includes features such as maxima of instantaneous frequency. These are subsequently tested on a support vector machine (SVM) and a feedforward backpropagation NN.

The applications of machine learning in state-of-the-art telecommunication networks and directly related areas can span a very broad spectrum [JZR+17][OH17b], which is essentially defined by the areas and sub-areas of research in which "intelligent" decision-making agents are a desirable or required element. As such, these would include, but not be limited to:

- Channel estimation [BDA20]
- Channel state feedback [MYG19][YMG19]
- Channel code design [BDBF+19]
- HetNet clustering [XOIH12]
- Resource optimisation [MS15]
- Multimedia compression [WSK18]
- Energy demand prediction [DOP+14]
- Energy modeling for energy harvesting applications [AMM13]

### 2.3.1 Neural Networks in Telecommunications

A direction of research described in [JZR+17] as relatively recently emerging and highly promising, and has, indeed, been experiencing dramatic increases in usage and development, lies in

paradigms of *computational intelligence*, among which deep learning and its sub-category of neural networks (NNs) are prominently featured. Deep learning, as a family of data mining methods based on learning data representations, is a considerable improvement upon simpler, task-specific machine learning algorithms.

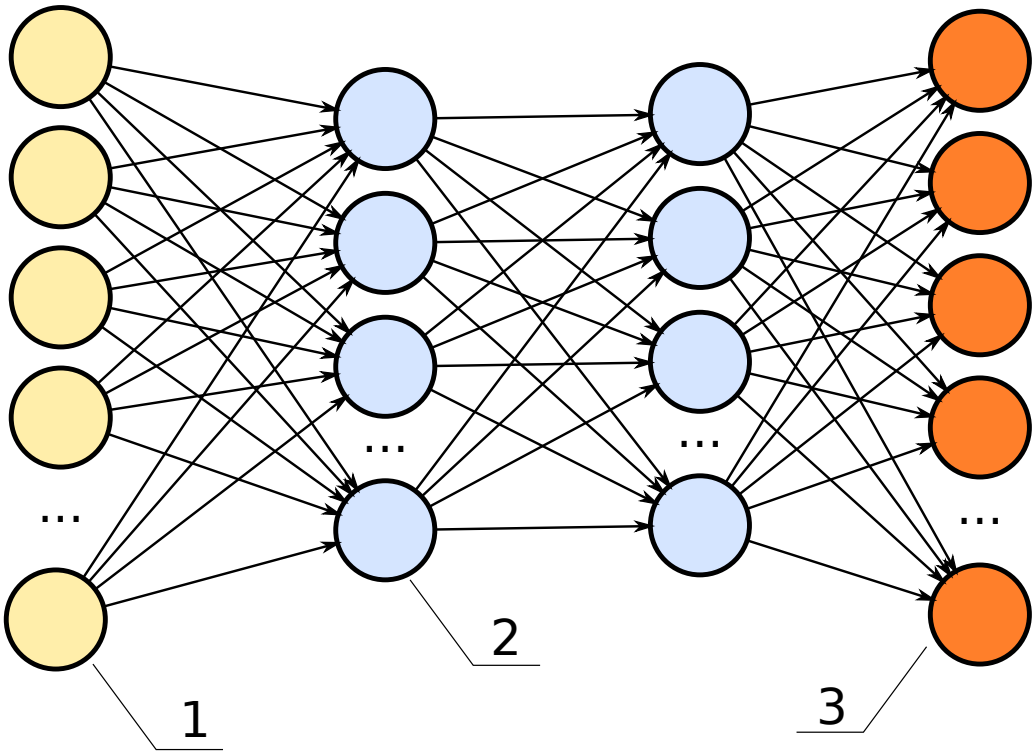



Figure 2.2: Schematic representation of a feedforward Neural Network.

We can define an artificial neural network (ANN) as a trainable information processing machine which is loosely modelled after the biological neural systems of the human brain. The main constituents of a neural network are units called artificial neurons, generally interdependent entities featuring multiple inputs and a single output each, as well as their connecting functions, which are usually weighted and non-linear. In this sense, each neuron can be described as representing a variable and each connection (synapse) as a parameter. Fig. 2.2 presents a feedforward network, one of the simplest ANN forms.

Each neuron is characterised by two potential states of function: training and usage. In the former case, data are supplied along with an instruction to "fire" or not, depending on the received input. In the latter, new data are presented and the neuron enters activation or stays idle based on the similarity of the input pattern to those for which the neuron was trained. Via

the generalisation of this function, a neural network is also able to function in either of those two states. Whether the data presented during the training stage are labelled or unlabelled characterises the training method as one belonging to supervised or unsupervised learning, respectively. A simplified representation of the function of an neuron is presented in Fig. 2.3.

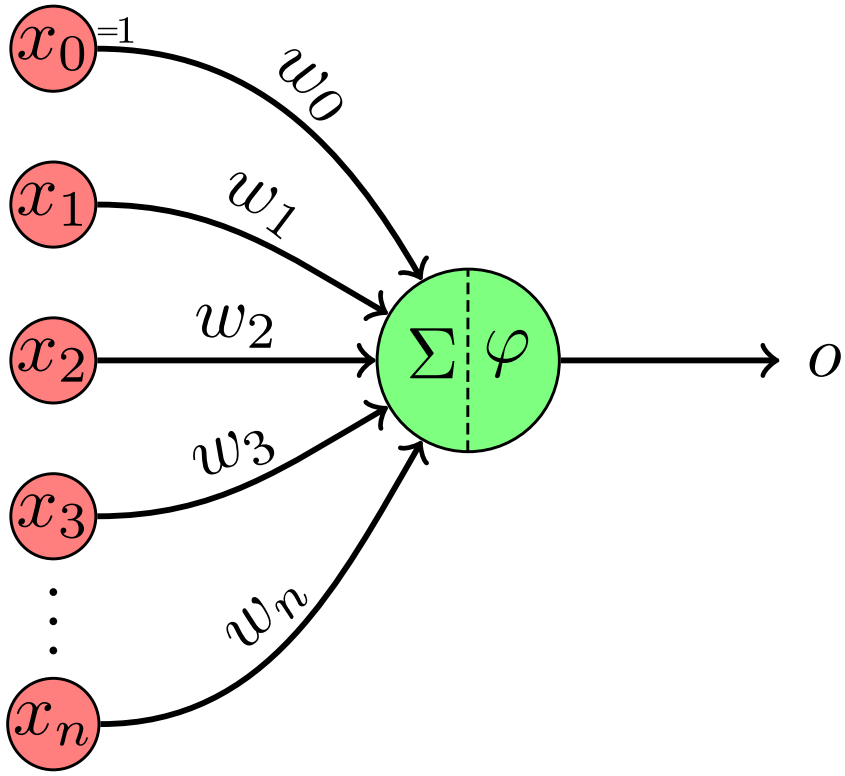



Figure 2.3: The function of an artificial neuron.

Some of the most important merits of artificial neural networks can be outlined in the following [Cla99]:

1. Ability to solve data-intensive problems
2. Rapid prototyping
3. Learning and intelligent adaptation
4. Scalability
5. Nonlinearity
6. Advantage over statistical models (*a priori* knowledge of underlying distributions not needed)

Consequently, three major application areas emerge based on the strengths of ANNs:

- Where an explanation of the network's decisions is unnecessary.
- Where there is prominence of randomness and stochastic behaviour.
- Where conventional, lower-complexity processes are insufficient or difficult to determine, or fail to adequately capture complexity in the data.

It is easy to deduce that neural networks are especially apt to handle nonlinear problems traditionally addressed by various other methods, including pattern recognition, signal processing (including image and sound), unsupervised clustering and data compression. From the field of pattern recognition alone, sub-applications such as fingerprint recognition, signature verification and secure entry systems [Cla99] are worth mentioning.

### 2.3.2 Convolutional Neural Networks (CNNs) and their Usage

While conventional (i.e. feedforward) NNs roughly function in the way described in Fig. 2.2 or Fig. 2.3 (usually with minor differences centred around structure size, method of training and activation functions), the novelty inherent in CNNs is that each neuron is not merely a weight operation followed by an activation function, but rather a filter of three dimensions (height, width, depth) which executes a convolution operation upon the input data and gives an output called a *feature map*, which may or may not be further used as input to an activation function. E.g. for a real one-dimensional input $\boldsymbol{y} \in \mathbb{R}^n$ and a filter $\boldsymbol{q} \in \mathbb{R}^W$, the convolution operation is described as such:

$$z_i = \sum_{j=1}^{W} y_{i+j-1} q_j, \quad i = 1, ..., n - W + 1 \tag{2.3}$$

The filters employed in CNNs are also commonly known as *kernels*. As Eq. 2.3 demonstrates,

kernels perform an operation which is geared towards distinguishing certain patterns or features in the input.

The idea for the employment of convolutional layers in neural networks arose from the problems inherent in the processing and classification of large files such as images and audio. By way of example, an RGB image of 256 pixels in both width and height is described by 196,608 arithmetical values; processing such a file with a conventional NN would require a number of parameters in the order of $10^{10}$, and the resulting model would constitute a great burden in terms of computation complexity and time. On the contrary, CNNs exploit invariances and patterns in input files so as to result in a sort of dimensionality reduction.

As is clearly shown in Fig. 2.4, convolutional layers exploit local correlation in their inputs. In a sense, they can largely be considered pattern detectors; indicatively, convolution blocks in image processing are associated e.g. with edge and surface detection.

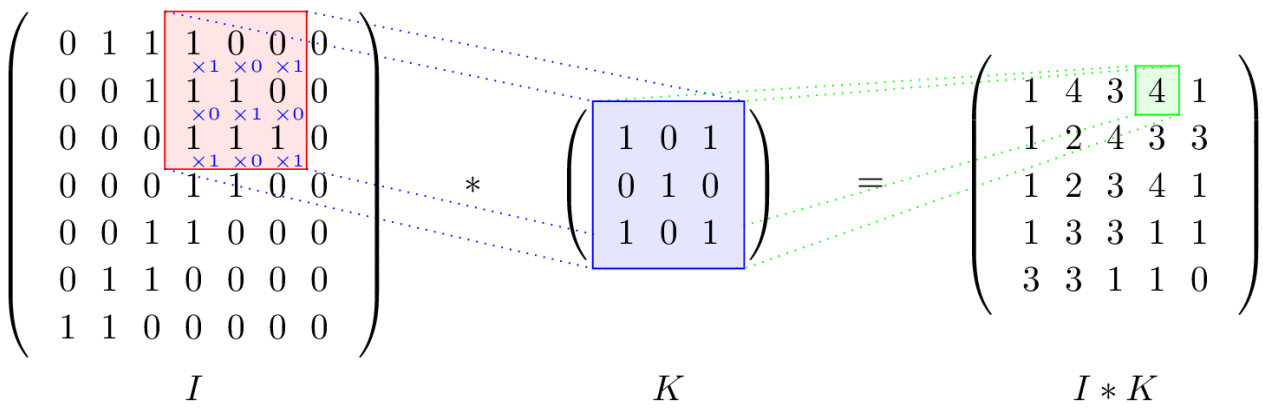



Figure 2.4: Visualisation of convolution function in a CNN

It bears mentioning that other structures which are also commonly found in CNNs are:

- Pooling layers. Pooling is a downsampling operation in which the input data is scanned in areas corresponding to a given volume (similarly to a convolutional kernel) and a single value (which can be e.g. a maximum or average) is extracted from each area.

- Fully connected (or dense) layers. They constitute the basic building block of feedforward

nets, and their operation can be succinctly described in the following equation:

$$\mathbf{z} = f(\mathbf{W}\mathbf{y} + \mathbf{b}) \tag{2.4}$$

Which means that an input $\mathbf{y}$ is mapped into an output of a different dimension $\mathbf{z}$ via a set of weights $\mathbf{W}$ and biases $\mathbf{b}$, while a nonlinearity is introduced by function $f$.

- Softmax layers. This variety of layer maps a $K$-dimensional input vector to a $K$-dimensional output representing a probability distribution, which is achieved with a softmax function:

$$\sigma(z_k) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}, \quad k = 1, ..., K \tag{2.5}$$

A typical CNN is comprised of a combination of the above elements in a logical succession, and the training procedure consists of the imposition of an optimisation algorithm in such a way that the output will optimally approximate a desired value. In the case of supervised learning, the output of a neural network can be e.g. a numerical value or a confidence vector of probabilities indicating the class to which an input data structure most likely belongs (see Eq. 2.5).

Given Equation 2.2, it is easy to comprehend why CNNs would be widely applied in this area of research, as a mathematical solution for such optimisation problems can be very computationally expensive and complex (if not completely intractable), especially since the distributions of the input data are exceedingly difficult to model; CNNs, on the other hand, establish themselves as strong alternatives precisely because they rely on step-wise fine-tuning of parameters for training (based e.g. on methods such as gradient descent). Furthermore, an enhanced degree of effectiveness would be reasonably expected from CNN classifiers for these problems, given certain properties [MCWW18] of modulated radio-signals, such as:

- Time-invariance of statistical features
- Compositional nature of signal likelihood functions

### 2.3.3 Other NN Paradigms

A brief mention may be made here of other varieties of neural networks which have been gaining ground in AMC research ([LYG17][RJY+19]) and have concerned us to a certain degree.

#### Long Short-Term Memory (LSTM) Networks

The LSTM unit was first introduced by Hochreiter and Schmidhuber in [HS97] as a specialised modification of the concept of recurrent neural networks (RNNs). RNNs are neural nets with memory units, designed for the purpose of extracting salient features from time-series data. They are widely used in the fields of hand-written character recognition and speech recognition.

At each step $t$, the function of an LSTM unit is characterised by three gates (input gate $i_t$, forget gate $f_t$, output gate $o_t$) and a state (parameterised by $c_t$ and $h_t$) according to the following equations:

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\
c_{in_t} &= tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
c_t &= f_t c_{t-1} + i_t c_{in_t} \\
h_t &= o_t tanh(c_t)
\end{aligned}
$$

Where the matrices $W$ and $b$, with different subscripts, are weights and biases, $\sigma$ is the sigmoid function, and $tanh$ the hyperbolic tangent function. As is evident from the above, this architecture enables the LSTM cell to retain or forget information from previous states, and thus facilitates the learning of dependencies and features over longer periods of steps.

LSTM networks were first proposed for AMC problems by Rajendran et al. in [RMG+18].

**Residual Networks (ResNets)**

ResNets were proposed by He et al. in [HZRS16] and purport to resolve problems resulting in poor generalisation, such as vanishing and exploding gradients.

The core entity of ResNets is called a residual block. Any traditional ANN layer or series of layers, whose function we denote here as $F$ and can include fully-connected projections, convolution, activations etc. can be turned into a residual block through the usage of a shortcut connection:

$$x_{out} = F(x_{in}) + x_{in}$$

Provided, of course, that $F$ is designed in such a manner that the dimensionality of $x_{in}$ is retained, and this input variable can be added to the output.

### 2.3.4 The State of the Art in AMC

Recent developments in the field of AMC have tackled a variety of issues and managed to solve a number of problems which have been recognised in this area of research over time, while exploring different trade-offs in the process.

Solutions have been offered for competitive reductions of the time required for the execution of algorithms [HGKL20], though certain problems with regard to the performance have persisted (e.g. with regard to QAM and AM modulations). In [DDB20], the issue of dropout usage is assessed, and the authors present a novel architecture called Dense Dropout Convolutional Neural Network (DDrCNN), which claims reduced network size and improved performance on 8 digital modulation classes; among others, the importance of reducing the number of dense layer weights and of avoiding dropout in convolutional layers is highlighted.

Zhang et al. in [ZDL+20] are mainly concerned with the issue of the reduction of the volume of data required for the training of DL-based AMC. Two novel varieties of ANN layers are

proposed for this purpose, one based on the convolution-like autocorrelation function, and the other on modulation filters. The results indicate a promising direction of research in ML with small training data sizes, but still fall short of most recent DL methods for modulation recognition.

The 2020 work by Ramjee et al. [RJY+20] constitutes one of the most important recent developments in AMC research. The main contribution consists in the production of an ensemble subsampling method based on a three-step algorithm incorporating an $\epsilon$-greedy decision tree, which attempts to rank samples from an input signal by order of importance for the classifier. The data volume reduction is significant and yet the classification accuracy manages to reach a saturation point of 97% at only $0dB$ SNR in three out of six assessed cases (with regard to subsampling rate), but the algorithm is reliant on foreknowledge of the SNR at the receiver.

Kumar et al. [KSJY20] present one of the most competitive state-of-the-art developments in the field of AMC. The proposed method relies on the extraction of constellation density matrices (CDMs) from the input signals, which are subsequently converted into RBG images through a series of filters and finally subject to classification via an ensemble of specialised CNNs with residual blocks. The algorithm is trained and tested on a set of 8 digital modulation classes (2-ASK, 4-ASK, QPSK, 8-PSK, 8-QAM, 16-QAM, 32-QAM and 64-QAM) and results in high-SNR regions ($SNR \geq 8dB$) surpass 95% and outperform most state-of-the-art classifiers. Nevertheless, a problem which is still present (and shall also be discussed in the research presented in this publication) is the strong resilience of QAM modulations against classification attempts; by way of example, even with the authors' proposed improved Inception Residual Network architecture, all 64-QAM signals were incorrectly classified for $SNR \leq 3dB$, and their classification accuracy reaches 80% or more only at $SNR = 10dB$.

Finally, it is worth mentioning that a number of non-DL-reliant AMC methods are still under consideration. In [HZST20], a clustering method is proposed in conjunction with an 8-step deterministic decision tree for the classification of four QAM modulation schemes. Additionally, it is evident e.g. from [AYSP20] that methods based on high-order statistical features (such as cumulants) have not become entirely obsolete; the publication in question suggests

improvements to the classic method via the introduction of the Kolmogoroff-Smirnoff test and a preprocessing phase consisting of constellation reflections.

A common factor which is easily detected in the aforementioned state-of-the-art approaches is that they are singularly concerned with simple, point-to-point communication systems, namely settings where a single receiver and a single transmitter interact without consideration of the presence of other devices or of data transfer (though our previous work [TTCG19] considered the cumulative effect of noise and interference, and the possible amelioration of effects of the latter). Likewise, the geographical aspects of a modulation recognition system are, to the best of our knowledge, not covered to any noticeable degree in relevant literature. As a result, our work's focus falls on both these aspects, while at the same time, optimisation is sought under usage of a modest dataset size (whereas, until recently, successful methods such as [HGKL20] would commonly rely on an excess of 1 million samples).

# Chapter 3

# AMC in the Presence of Interference

The subject of cognitive communications commands great interest in fifth-generation (5G) telecommunication networks, owing in great part to the increasing number of wireless devices sharing the same spectrum. In addition to spectrum sensing for seeking spectrum holes for interference-free communications [ZLHZ10], more advanced cognitive techniques would allow interference identification and cancellation to improve the rate and reliability of communication [WLY12].

The goal of the algorithms presented in the current chapter is to detect the modulation scheme of an interference signal, a step which is commonly leveraged for cancelling the same or reducing its effect. This will have to be carried out in the presence of a signal whose modulation scheme is typically known. Note that, from the AMC point of view, this is equivalent to AMC in the presence of interference, with or without known interference modulation.

## 3.1 Conception and Formulation of Setup

The formulation of the problem for the experiments considered in this chapter regards a scenario in which a point-to-point transmission is degraded by the presence of a second transmitting device, the signal of which is deemed undesirable for the receiver.

Let $y[n]$ be the discrete baseband received signal given by:

$$y[n] = h_1[n]b[n] + h_2[n]s[n] + w[n], \quad n = 1, \ldots, N, \tag{3.1}$$

where $b[n]$ are the samples of the main signal whose modulation scheme we wish to detect, $s[n]$ is the secondary signal, whose modulation is considered known, $h_1$ and $h_2$ are the corresponding complex channel gains, and $w[n]$ denotes an additive noise term. Here we aim to build and train a deep-learning-based AMC method which shall successfully identify the modulation scheme $C_b \in \{1, \ldots, M_b\}$ of signal $s[n]$. Furthermore, it is desired to assess whether the knowledge of the modulation class of the secondary signal, $C_s \in \{1, \ldots, M_s\}$ can be incorporated as an input into the classifier and improve the overall performance. Figure 3.1 is a visual representation of the problem as stated above.

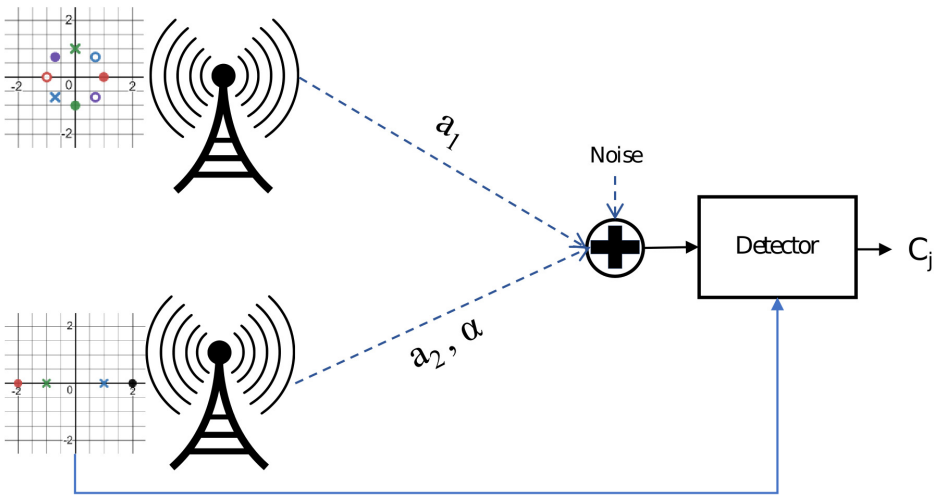



Figure 3.1: AMC in the presence of interference.

Since various reasons, described above, led towards the choice of a CNN as the core of the setup, a data-driven approach was followed, and the deep-learning AMC algorithm trained via the utilisation of signals from known modulation constellations.

## 3.2 RadioML Dataset

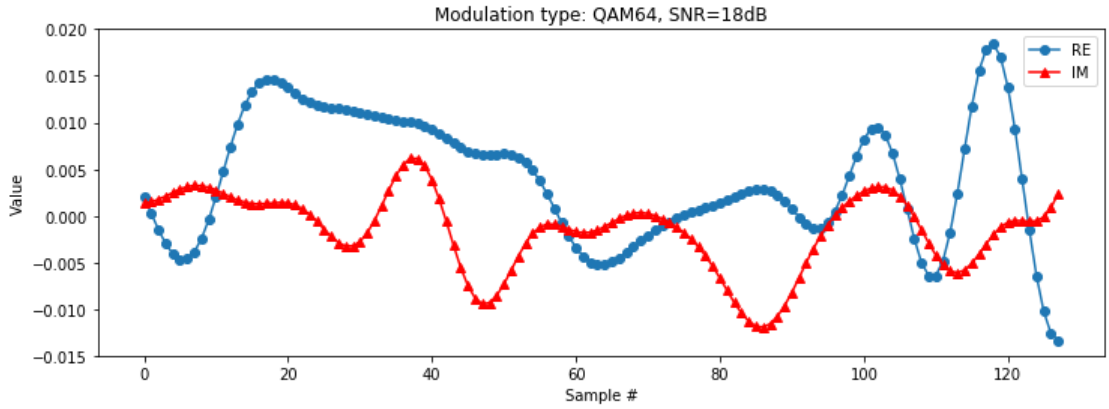
A dataset widely used in modulation-detection-related experimentation at the time of these first experiments was RadioML 2016, proposed and described by O'Shea et al. in [OC16]. For the purposes of this study, an extended version of the RadioML dataset was used, with the following features:

- 1.2 million samples (separated into training, validation, and testing sets),
- 10 different modulation schemes (BPSK, QPSK, 8PSK, 16QAM, 64QAM, GFSK, CPFSK, and PAM4 as digital, and WB-FM and AM-DSB as analogue modulation schemes),
- Sample format: $2 \times 128$ vectors (two channels corresponding to in-phase and quadrature components),
- Signal-to-noise ratio (SNR) values $\in [-20, -18, \ldots, 16, 18]$ dB.

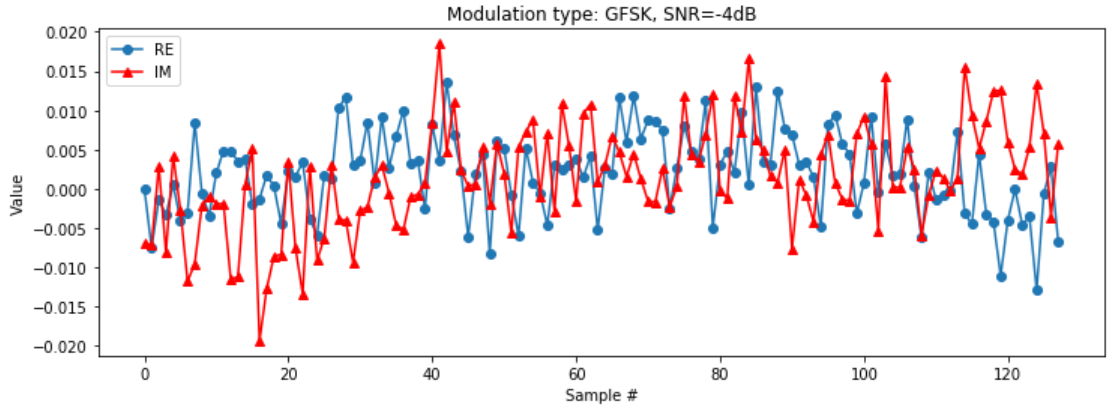
In the interest of comprehensive visualisation, Fig. 3.2 presents the time plots of the in-phase and quadrature (I/Q) components from two example signals from the augmented RadioML dataset, both from digital modulation schemes, but with significantly different noise power levels (with SNR values at 18 and -4 dB, respectively).

Two settings of the problem were considered based on the data features detailed above: a simplified scenario, in which samples for $s[n]$ and $b[n]$ were selected from a subset of the available modulation schemes, and consequently, a more complicated one where all modulations participate as candidates for both signals. For this purpose, samples are selected as follows:

- Only data for SNR values $\geqslant 6$ dB retained for $b[n]$ and $\geqslant 16$ dB for $s[n]$,
- In the easier setting, five classes for the interference (8PSK, PAM4, QAM64, QPSK, WBFM) are considered, and four for the desired signal (AM-DSB, BPSK, CPFSK, QAM16).




(a)




(b)

Figure 3.2: Examples of signals from RadioML 2016.

Through combinations of repetition and shifting, the selected samples were combined so as to create a dataset of 600000 frames for training and 100000 frames for validation. In particular, the sets which were used for the experiment were created through the following procedure:

Let $y_i$ be an instance of the final dataset, $b_i$ and $s_i$ sample sequences from the earlier isolated primary and secondary sets corresponding to the same index $i$. Each $y_i$ is created using Eq. 3.2:

$$y_i[j] = b_i[j] + \alpha_i s_i[j] \,, \quad i = 1, \ldots, M \,, \quad j = 1, \ldots, N, \tag{3.2}$$

where $M$ is the dataset size ($M = 6 \times 10^5$ for training, $M = 10^5$ for validation), $N = 128$ is the number of samples in each sequence as mentioned above, and $\alpha$ is a factor of attenuation or amplification which adjusts the signal-to-interference ratio, selected randomly for each instance,

assigned one of the following values:

$$\alpha \in [0, 0.1, \frac{\sqrt{10}}{10}, 0.5, \frac{\sqrt{2}}{2}, 1, \sqrt{2}, 2, \sqrt{10}], \tag{3.3}$$

or, equivalently,

$$\alpha \in [-\infty, -20, -10, -6, -3, 0, 3, 6, 10] \text{ dB}. \tag{3.4}$$

From Eq. 3.2, the reason due to which the highest available SNR values were chosen for forming the subset of $s_i$ signals is made apparent: the added noise will also be multiplied by the $\alpha$ factor during the process, so signals with an SNR of 16 and 18 dB were selected due to the unavailability of noiseless signals in RadioML.

In [OH17a] it has been demonstrated that feeding raw I/Q values to a CNN architecture results in competitive detection accuracy during testing, even surpassing known feature-based techniques that have been developed over many years. It remained to be explored whether this result would still hold in the presence of an interference source, as well as whether alternative data representations, instead of raw I/Q symbols as available in RadioML, could be of any use in improving classification accuracy (by way of example, the experiments conducted in [KKMP17] include the conversion of the I/Q format into amplitude and phase values, and a small improvement in performance is reported).

## 3.3 Exploiting Desired Signal Modulation Information

It is inherent in the system model detailed above that the modulation scheme used for the secondary signal is known beforehand; as a result, it was worth investigating whether this

additional information could be exploited in order to improve detection accuracy, and how significant this improvement would turn out to be.

The most straightforward manner to exploit this information would be to train a separate classifier for each class of modulation; *i.e.*, a separate neural network to be trained so as to detect the interference modulation for each kind of desired signal modulation. Although this approach is intuitive and potentially effective, its complexity and training time grows significantly in proportion with the number of available classes. This would render such a method very difficult to implement in terms of resources, but also highly impractical in a potential scenario in which re-training is necessary due to the emergence of new conditions [RDAS+20].

Instead, in this case the signal class information is incorporated into the neural network as an additional input. This was done in the form of one-hot encoding, as commonly done for the conversion of categorical data into feature vectors [SG17]. In particular, a vector of length $M_s$ is appended, which consists of all zeros, save for a single 1 at the location corresponding to the index of the modulation class of the desired signal.

## 3.4 Simulations and Results for RadioML

Unless otherwise stated, all simulations conducted for the initial AMC experiment are run upon datasets with 600000 samples in the training set and 100000 in the validation and testing set. Each input frame consists of 128 I/Q pairs sampled at the side of the receiver. The distribution of modulation schemes is as described in Section 3.2, and both the setting in which all 10 modulation classes are present, as well as the restricted, simplified setting, are considered, in order to assess the impact of the number of classes on accuracy.

As in the cases presented in [KKMP17] and [OWVC17a], a CNN classifier is leveraged with the purpose of circumventing the traditional process of feature extraction (under the assumption that the process of appropriate pattern recognition is wholly undertaken by the neural net). The CNN architecture used for classification is presented in Fig. 3.3. It relies on the template of a standard architecture used for signal classification algorithms, consisting of two pairs of

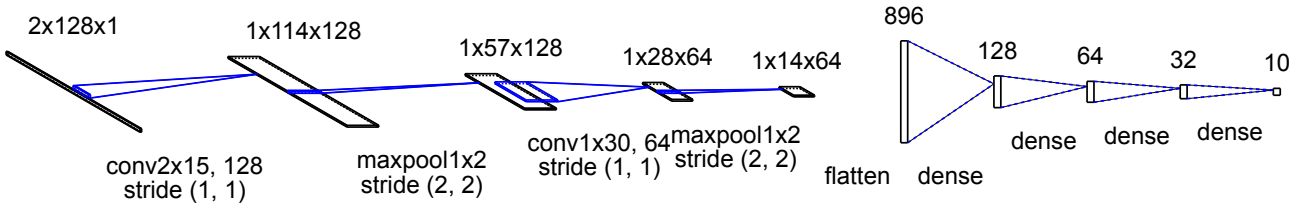



Figure 3.3: CNN classifier architecture for AMC with interference.

convolutional and max-pooling layers followed by three fully-connected layers, terminating in a softmax layer with either 10 or 5 outputs, depending on the setting. The number of fully-connected layers on which we settled was born out of the structure of effective pre-existing neural networks, as well as repeated experimentation, where it was noticed that decreasing their number would weaken performance, and increasing it would offer no tangible improvement.

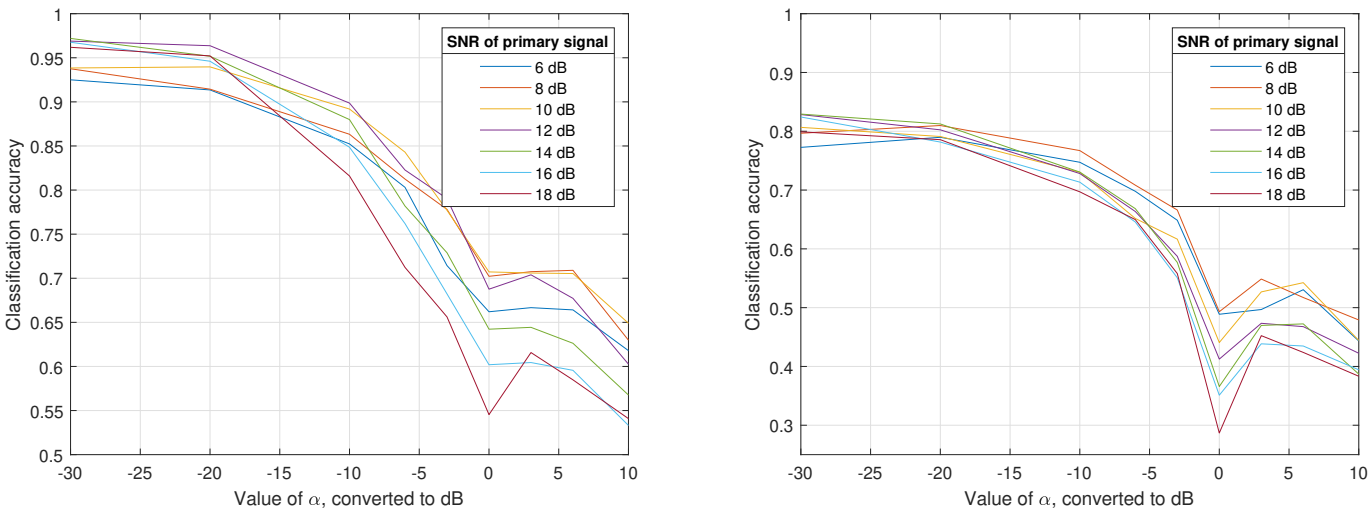
The fully-connected layers are initialised with the Xavier function [GB10], and the first two are also fitted with dropout mechanisms. The Xavier (Glorot) algorithm assigns initial values to the weights $w_i$ of a neural network layer by drawing them from a distribution with the following metrics:

$$\mathbb{E}[w_i] = 0$$
$$Var(w_i) = \mathbb{E}[w_i^2] = \frac{1}{N_{in}}$$

where $N_{in}$ is the number of the input neurons for the layer in question.

The final layer features a softmax function and has a number of outputs equal to the total number of classes in the dataset, and thus can be considered a confidence vector quantifying the probability that an input signal might belong to any one of the classes in $C_b$.

With the exception of the final layer, all others utilise a leaky rectified linear unit (ReLU) activation function [MHN13], the formula for which is given in Eq. 3.5. The traditional ReLU function has been distinguished as a superior alternative to other traditionally used activations such as tanh and sigmoid, in great part due to its smaller computational cost, its resilience to backpropagation errors, and its useful ability of overcoming the problem of vanishing gradients; its so-called "leaky" variant is often used to deal with the *dying ReLU* problem, which might




(a) 5 classes (b) 10 classes

Figure 3.4: Desired signal modulation unknown - I/Q input.

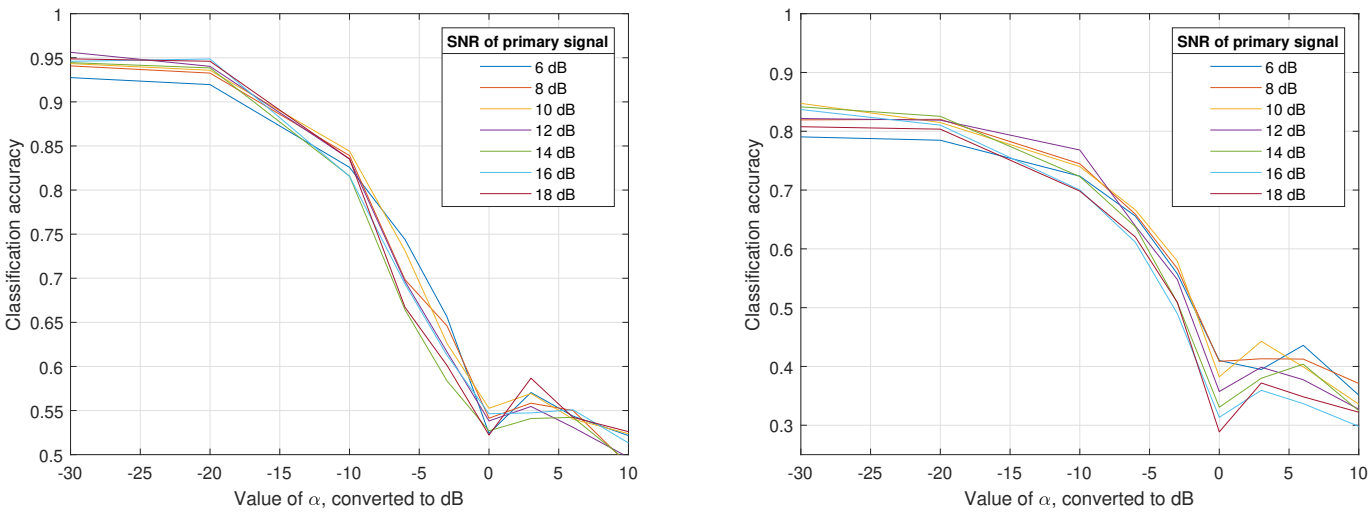
emerge e.g. if the available data are not properly normalised.

$$f(x) = \begin{cases} x & for \;\; x < 0 \\ \alpha x & for \;\; x \geq 0 \quad (\alpha > 0) \end{cases} \tag{3.5}$$

All experiments were carried out on a desktop computer running a Windows 10 64-bit OS on an Intel®Xeon®Silver 4110 CPU @2.1 GHz, equipped with a 32GB main memory and an 8GB graphics processing unit. It is noted that only one CNN was trained per experiment, i.e., each training phase included all datapoints, without regard to SNR or SIR. Simulation results for the simpler (5 classes) and more difficult settings (10 classes) will be presented next to each other for ease of comparison.

In Figures 3.4 and 3.5, we demonstrate the classifier's performance when there is no knowledge of the modulation scheme of either signal. Note that $\alpha = -30$ dB corresponds to the case in which the transmission is uninhibited point-to-point, which is equivalent to the scenario detailed in [OH17a].

It becomes apparent that, as the signal-to-interference ratio, quantified by $\alpha$, increases, the desired signal gradually becomes more dominant and the expected deterioration of performance




(a) Amplitude/phase format - 5 classes

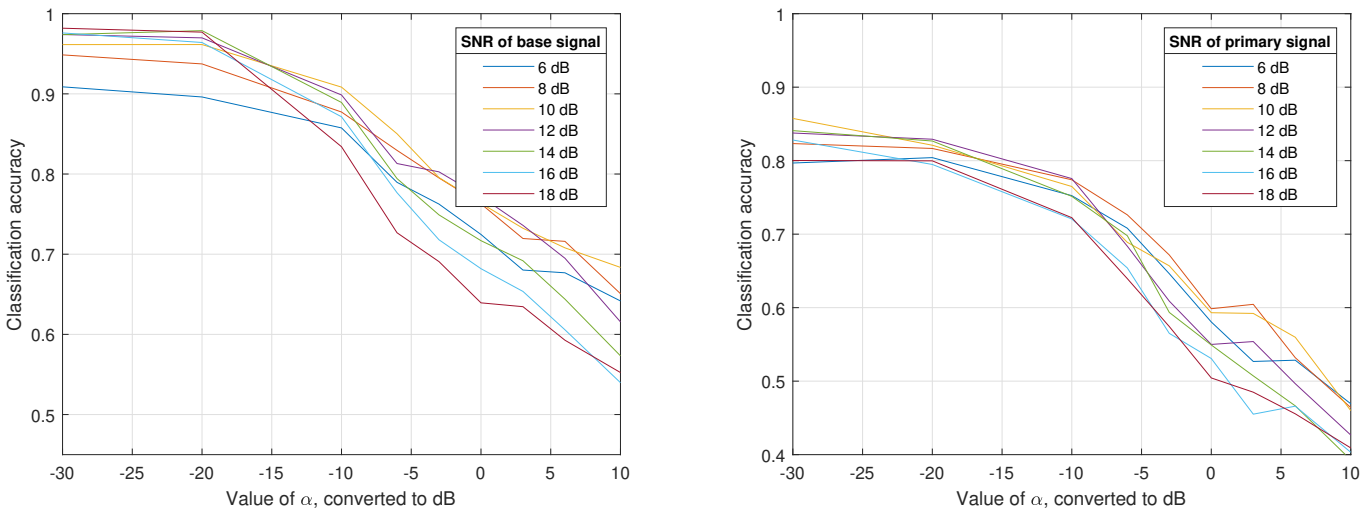
(b) Amplitude/phase format - 10 classes

Figure 3.5: Effects of amplitude-phase transform.

is noticed. Worth noting is that the higher SNR values for the interference signal only provide better results when the desired signal is weak, whereas the high-SNR curves are outperformed by those derived by noisier inputs for higher $\alpha$, sometimes even by 10%. Also notable is a steep decline of most curves when the $\alpha$ factor is equal to 0 dB, which is accompanied by a stronger rise afterwards in the 10-class scenario. This behaviour was mainly attributed to the confusion of the classifier over two independent components superposed at equal power.

Fig. 3.5 further demonstrates that the conversion of I/Q values into pairs of amplitude and phase does not improve the performance of the classifier, even though it seems to be reducing the overall variance between different curves.

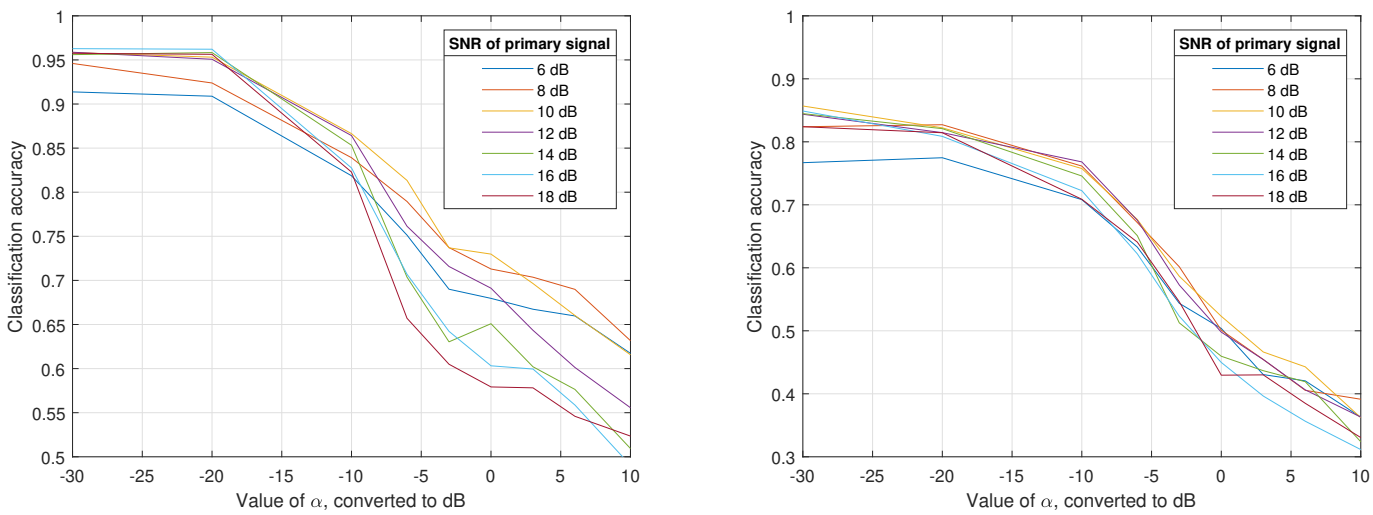
In Fig. 3.6 and 3.7, the results are presented for the case in which it is assumed that the desired signal's modulation class is known. In general, the one-hot encoding method, despite its simple nature, succeeds in non-negligible performance improvements. It is observed that detection accuracy has improved significantly compared to Fig. 3.4 and 3.5, though with a slightly higher variance between the accuracy of different curves (corresponding to different SNR values). The performance still degrades with further the increase of $\alpha$; this is because, although the desired signal modulation class is unknown, a complete removal of its effect is not plausible for this current algorithm.




(a) One-hot method - 5 classes

(b) One-hot method - 10 classes

Figure 3.6: Desired signal modulation known - I/Q input with one-hot encoding.




(a) One-hot plus $A/\phi$ - 5 classes

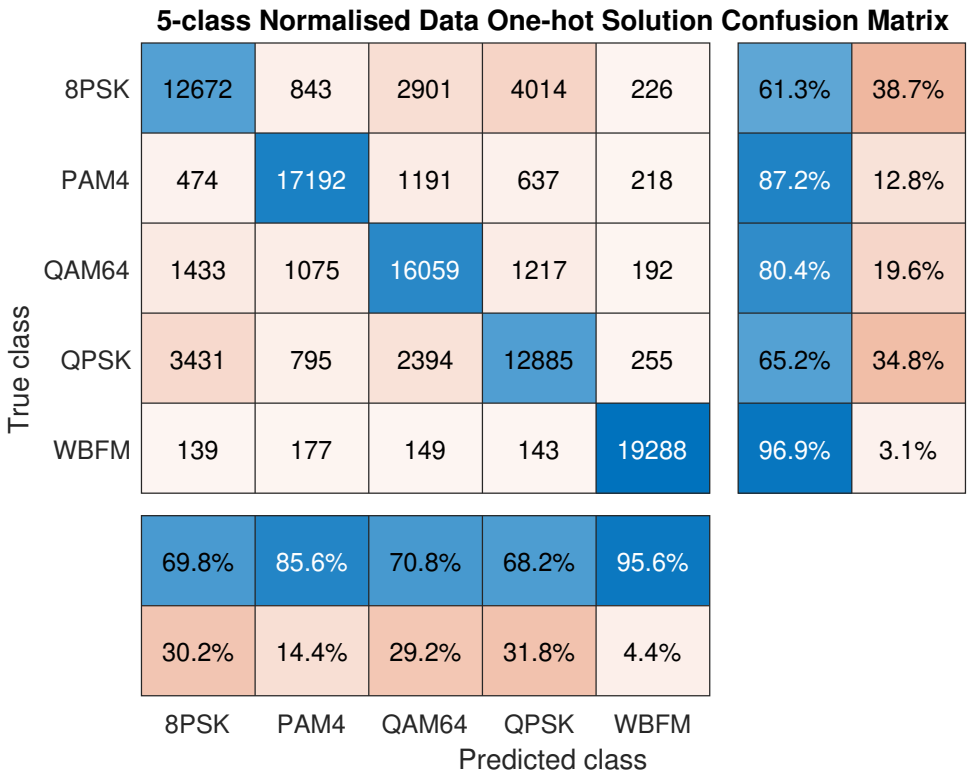
(b) One-hot plus $A/\phi$ - 5 classes

Figure 3.7: Effects of combined $A/\phi$ and one-hot techniques

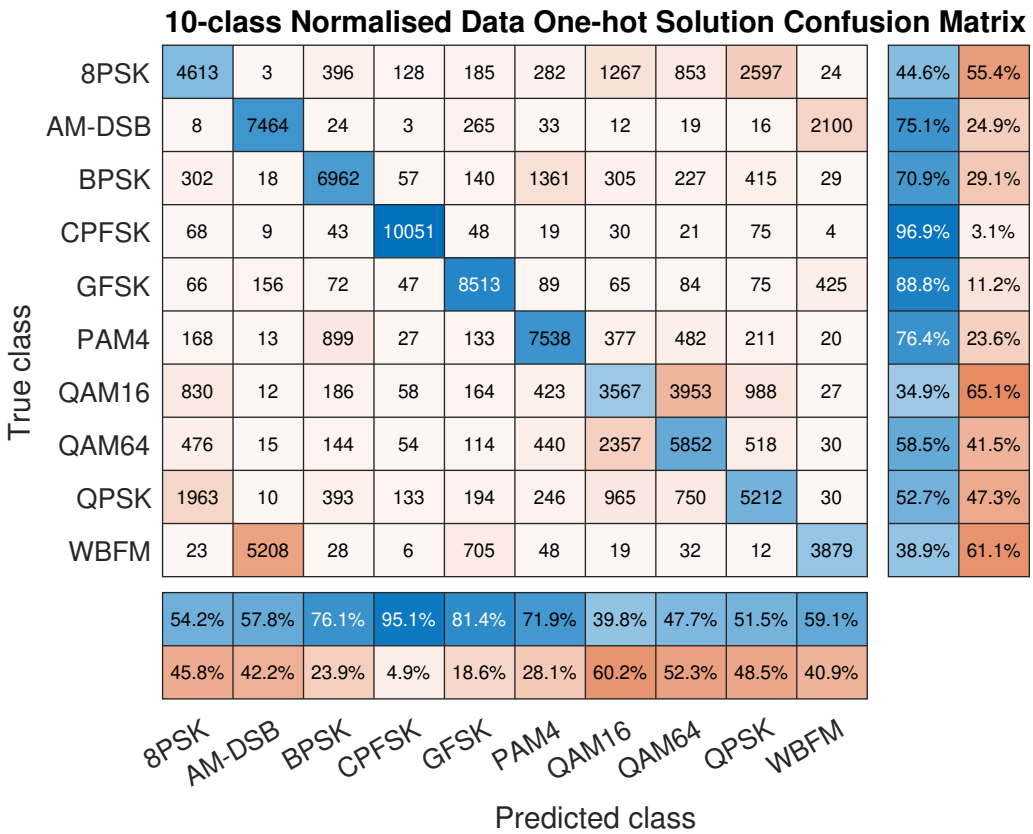
In the 10-class case, the application of one-hot encoding is especially noteworthy for its effect upon the 0dB "canyon", which disappears almost entirely.

Figures 3.5 and 3.7 additionally demonstrate that, although the transformation of I/Q values into pairs of amplitude and phase was proven beneficial in its original context [KKMP17], in this superposition problem it does not seem to solve many of the problems present; on the contrary, it seems to introduce additional confusion, which is more apparent for higher values

of $\alpha$.




(a) One-hot solution, 5 classes




(b) One-hot solution, 10 classes

Figure 3.8: Confusion matrices for experiments with normalised signals

Singling out the cases described in Fig. 3.6 as the best which were achieved through the implementation of the algorithms analysed above (reaching 78.2% accuracy for 5 classes, and 55.7% for 10, over all sub-categories of SIR and SNR), confusion matrices are computed for the classification results on the validation set after the end of training so as to render clearer the effectiveness of the classifier in cases of different signal classes; the results for both the simple and complex settings are presented in Fig. 3.8.

In the less challenging setting of the experiment, it is reasonably expected that WBFM should be the most effective to recognize, as it is the only analogue modulation present in the reduced dataset, while the two different PSK modulations are easier to confuse with each other; those expectations are confirmed through the experiment.

In the setting where all 10 signal classes participate, the most telling feature of the final confusion matrix is the low performance rate for QAM16, which is most easily misclassified as the only other available QAM modulation scheme (QAM64); a similar tendency towards confusion is observed with 8PSK and QPSK, though to a less pronounced degree. Likewise, WBFM loses its formerly observed edge, as many of its instances are, instead, classified as AM-DSB signals (an effect which, again, can be explained by the fact that those are the two available analogue modulations). The two frequency-shifting modulations (CPFSK, GFSK) are proven the most robust, and do not mutually deteriorate their performance as it happens, e.g., with PSK or analogue schemes.

## 3.5 Experiments on New MATLAB Dataset

For purposes of enhanced control over data and propagation parameters, which is not yet possible to achieve with a "black-box" library such as RadioML, additional datasets for modulation recognition were created with the aid of specialised algorithms in MATLAB R2019a (an example of their usage is given in [MATb]), still focused on the problem of point-to-point modulation recognition and its complication in the presence of interference.

Dataset creation follows the steps described below, for each individual frame:

1. Data are drawn from an initial distribution (random integers for digital modulation, a .wav audio file including speech and music for analogue) so as to create $N$ symbols.

2. The aforementioned symbols are modulated according to the selected method.

3. The resulting waveform is transmitted through a Rician fading single-input-single-output (SISO) channel model (the process includes conversion to and from bandpass).

4. Finally, the received signal is further degraded at the receiver side by the addition of additive white Gaussian noise (AWGN), at a signal-to-noise-ratio (SNR) value selected randomly from a finite, discrete uniform distribution. (This operation might be either independent or embedded into the channel model.)

The above procedure is repeated for all classes ($C = 11$; an additional analogue modulation is available in the MATLAB codes), and so that each class comprises a given number of frames. The potential classes of a signal represent one of three analogue (BFM, AM-SSB, AM-DSB) or eight digital (BPSK, QPSK, 8PSK, 16QAM, 64QAM, PAM4, CPFSK, GFSK) modulation schemes available. Where different setups necessitated this, new data groups were created from earlier samples through isolation of particular characteristics, repetition, shifting, amplification/attenuation, and combination, according to case-specific needs.

The Rician fading channels configured for the simulation feature one line-of-sight (LOS) and three non-LOS coefficients, the latter with delays of $[1.1,\ 3,\ 5.5] \times T_s$, where $T_s = \frac{1}{f_s}$ is the sampling period, and corresponding gains $[-2, -3, -8]dB$. AWGN is added following a distribution of SNR values ranging from $0dB$ to $18dB$, in increments of $2dB$.

The sampling frequency is set to $f_s = 200kHz$, for a constant rate of 8 samples per symbol. The carrier frequences are $f_{ca} = 100MHz$ for analogue and $f_{cd} = 902MHz$ for digital modulations.

The combination of signals on the side of the receiver follows the same template presented in Eq. 3.2, and with the same values for $\alpha$ assigned earlier.

The MATLAB data were used all throughout this new iteration of the AMC experiment. For the primary signal ($b[j]$), all different SNR values from the initial candidate set are assessed.

For the secondary signal ($s[j]$), an additional auxiliary dataset was created, following the same procedure described above, but a constant and very low noise level was aimed for (SNR = 60 dB), so as to avoid further deterioration due to the resulting amplification of the additive noise, especially for $\alpha > 0$ (see Eq. 3.2).

### 3.5.1 Simulation Setup

After testing the case where no knowledge regarding the secondary signal is presupposed, we seek again to determine whether the modulation of that signal, when known *a priori*, can be incorporated into the classifier structure as extra information so as to improve performance. Whereas the idea which was implemented earlier (i.e. under RadioML) was to append the one-hot encoded representation [SG17] of the class of the secondary signal to the in-phase vector, while the quadrature vector was padded with zeros, an alternative method was introduced, inspired by a technique for combination of heterogeneous data as ANN inputs in [LPKQ16], consisting in appending the one-hot vector instead to the flattened one-dimensional output of the convolutional parts. The latter method was ultimately preferred due to slight improvements in performance.

In this later conception of the setup, a form of ANN called fully convolutional neural network (FCNN) (the applications of which include e.g. semantic segmentation [WYO16][LYSX18]) is selected as optimal over other candidate configurations (simple multilayer perceptron, conventional CNN, long short-term memory (LSTM), convolutional LSTM deep net (CLDNN)) for this specific problem (a general outline is given in Fig. 3.9). FCNNs were first proposed in [LSD14] for tasks of semantic segmentation of image data, and their success was attributed in large part to their ability of outputting heatmaps instead of vectors.

The characteristic element of FCNNs is the absence of fully-connected layers in any section of the network (except e.g. in the case of classification, where the output needs to be a confidence vector and the presence of a single dense layer – typically with softmax activation – is necessary so that the network outputs a vector whose number of elements is equal to the number of

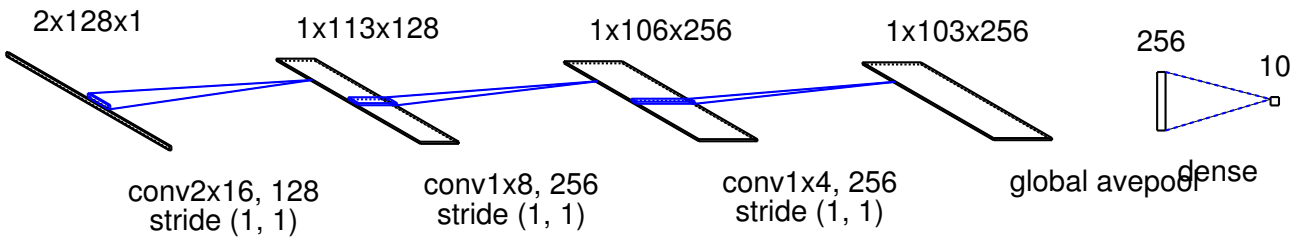



Figure 3.9: General FCNN classifier architecture

classes); by contrast, most ANN implementations in relevant literature make use of one or more additional fully-connected layers [RJY+19][ZDL+20].

FCNNs have been preferred in our work henceforth, as during the course of our experimentation they were evidenced to offer superior results and better operation times than classic CNNs and other rival configurations. Furthermore, since it has been pointed out [DDB20] that one of the most crucial factors in the reduction of the size of ANNs is the reduction of the number of filters in fully-connected layers, the FCNNs which we have used additionally benefit in this respect through the complete absence of dense layers (except, if counted, for the softmax at the end).

A signal is presented as input to the ANN in the form of a tensor with dimensions $2 \times N$, thus representing the in-phase and quadrature components of the same. The value of $N$ differs according to the needs of each individual experiment, as do the dimensions of the kernels (in general, an increase in input dimensions would prompt the usage of larger kernels). Each of the three convolutional blocks present in the architecture performs three operations in the following order:

$$\hat{\mathbf{x}} = \mathbf{W} \circledast \mathbf{x} + \mathbf{b}$$

$$\hat{\mathbf{x}}_{\mathbf{n}} = Batch\_Norm(\hat{\mathbf{x}})$$

$$\mathbf{h} = ReLU(\hat{\mathbf{x}}_{\mathbf{n}})$$

The first of these three equations represents a convolutional layer, which convolves input data $\mathbf{x}$ with kernels $\mathbf{W}$ and adds weights $\mathbf{h}$ ($\circledast$ represents the convolution operator) so as to perform a form of dimensionality reduction and seek local correlations in the input.

ReLU stands for the Rectified Linear Unit operator, which is defined as $ReLU : f(x) =$

$max(0, x)$ and is implemented as a way of introducing a nonlinearity (without which the entire structure of any deep ANN would become redundant, as it would be reduced to a chain of linear modules). The leaky version of the ReLU function, mentioned earlier, was eschewed in this case due to its apparent inability to provide better performance results, as evidenced through repeated experimentation.

Batch normalisation was employed due to its properties of:

- Encouraging greater independence between layers in learning,
- Reducing covariate shift, thus enchancing generalisation abilites and suppressing overfitting, and
- Accelerating both training and classification operations.

Finally, the last layer in the structure maps the output of the previous one to a final vector from which the classifier decision is derived, as it is equipped with a softmax activation function and its output is of dimension equal to the number of classes $C$ (see Eq. 2.5 for definition of softmax function). The decision is made by the algorithm based upon the maximum element, thus it constitutes a vector of confidence scores for each class. Intermediate pooling operations were omitted in this case, to the end of increasing the generalisation capabilities of the network.

The final (in the current case, third) block of layers is followed by a 2-dimensional global average pooling layer, which is preferred to the conventional flattening function and subsequent filtering through fully-connected layers, so as to reduce the total number of weights, but also because the elimination of fully-connected structures forces the feature maps to be more closely correlated to the various classes, i.e. to act as "class confidence maps" [LCY14].

Both training and testing are carried out with the corresponding datasets divided in mini-batches of size $B = 256$. The computation of gradients and subsequent update of network parameters benefits significantly from the parallelisation capabilities of Python and TensorFlow, so that training with batches with a size greater than 1 improves performance and reduces

operation time compared to the case when one element from the dataset is selected on each iteration of gradient descent.

The loss function which was chosen was the sparse softmax cross-entropy loss available on TensorFlow, a general description of which is the following:

$$\mathcal{L} = -\frac{1}{S}\sum_{i=1}^{S}(\mathbf{y_i} \odot \log(\hat{\mathbf{y}}_\mathbf{i}) + (1 - \mathbf{y_i}) \odot \log(1 - \hat{\mathbf{y}}_\mathbf{i})) \tag{3.6}$$

Where $S$ is the total number of frames in a given dataset or subset thereof, $y_i$ is the one-hot expression of the true class, $\hat{y}_i$ is the corresponding softmax output of the CNN (i.e. the class confidence vector), and $\odot$ stands for the inner product of vectors. Optimisation follows through the implementation of stochastic gradient descent, with the CNN parameters $\theta$ upgraded from step $t$ to $t+1$ as such:

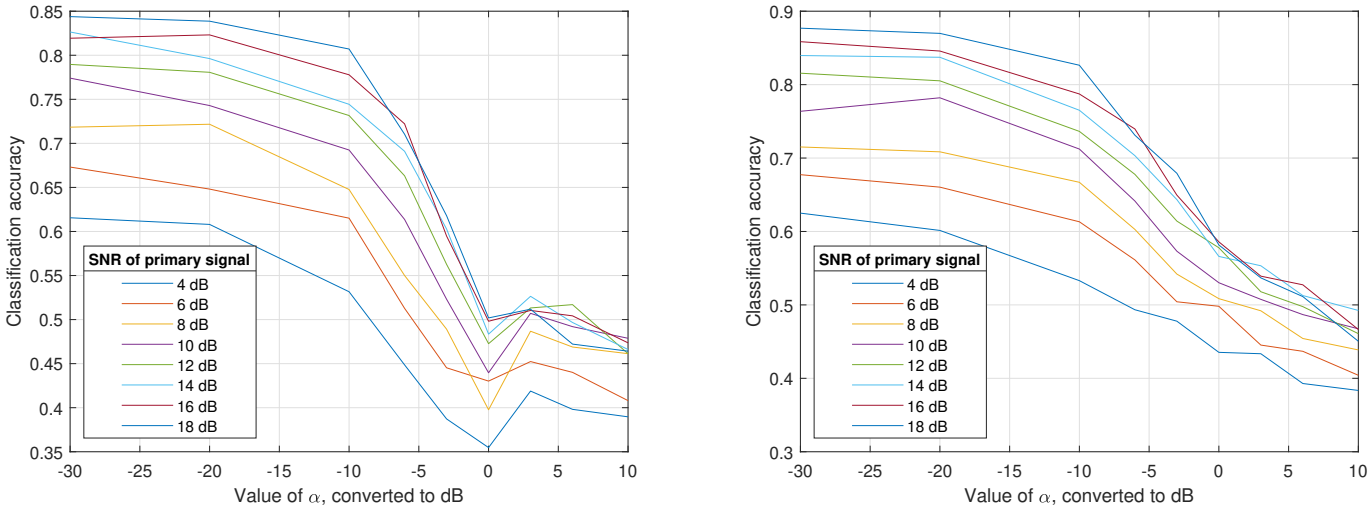
$$\theta_{t+1} = \theta_t - \eta\nabla\mathcal{L}(\theta_t) \tag{3.7}$$

Where $\eta$ is the selected base learning rate. The exact gradient descent model which was applied is the adaptive moment estimation method (Adam) [KB14]. Adam relies on the dynamic adaptation of learning rates based on lower-order moments of the gradient, aided by the calculation of exponential moving averages of the same and the squared gradient.

### 3.5.2 Results

Various groupings of modulation classes were assessed during tests; here, the most challenging setting is presented, where all 10 available classes were selected for both signals. The results for performance on a test set of $10^5$ samples (with $6 \times 10^5$ in training) are demonstrated in the curves of Figure 3.10, separated by primary signal SNR value and power ratio.

The effect of the SNR upon classification accuracy is clearly observed, accounting for significant differences between different curves in most regions of the graph (even close to 30%, indicatively, for $\alpha = -10dB$); this divergence is noted to be much more pronounced than in the case of the RadioML data (it was noted during the creation of the dataset that plots of signals featuring the same SNR would look more heavily distorted on the MATLAB dataset).




(a) Without one-hot modification

(b) With one-hot modification

Figure 3.10: Effect of leveraging desired signal modulation - 10 classes.

An overall increase in accuracy is observed when the proposed modification of the one-hot incorporation step is applied to the problem, in spite of the simplicity of the method; the application thereof results in accuracy improvements ranging from 15% to 25% for different SNR values.

Although it has not been possible to derive an exact function of the classifier for this problem, at the time of the experiments described here the conclusion was deduced, with reference to [PJW+19], that a CNN-based modulation detector interprets input signals as constellations on a 2-dimensional plane. In the current case, likewise, the input is viewed as the superposition of two constellations for every single frame, occupying areas of different radii according to the value of $\alpha$. In light of that, the behaviour of the proposed classifier may be in part explained, particularly with regard to the 0dB-area performance drop, since it is exactly there that the power ratio will result in the superposed constellations occupying largely the same area of the

2D plane.

## 3.6 Effects of Low-Resolution Signal Quantisation on AMC

Massive multiple-input-multiple-output (MIMO) communications is a family of telecommunication techniques which rely on serving a large number of end users with comparably large numbers of antennas at the base station [ZDL+18][MKK+19]. Among the results of the popularisation of massive MIMO systems, one can note the increased interest in high-speed, low-resolution analogue-to-digital converters (ADCs) [Ld17]. More specifically, the presence of multiple antennas, while highly promising in its offering, e.g., improved spectral efficiency and suppressed inter-user interference, also comes at an increased computational cost, since it creates the need for processing much greater volumes of data. Additionally, another practical constraint is introduced in the form of energy consumption issues, since the operating power increases not only with the number of required ADCs but also with the bit resolution level.

Given this issue, various forms of signal quantisation have been implemented as methods of reducing data sizes, and it is worth assessing the impact of the different levels of signal quantisation upon a modulation recognition algorithm. Although implementations where an antenna is fitted with one 1-bit quantiser each for the in-phase and the quadrature components of the baseband signal are already influential [CMH16], settings where the quantisation level is more granular are also assessed.

### 3.6.1 Point-to-point modulation recognition

A simple assessment of the effects of quantisation upon the proposed AMC algorithm is demonstrated in Fig. 3.11, with results from the RadioML dataset which was used in earlier attempts, as explained in previous sections. The resolution of the initial 32-bit floating point values is reduced to either 1, 2 or 4 bits for each of the in-phase and quadrature values.

Python does not feature native support for data resolutions beneath 16-bit for floating point

numbers. As such, we derived a custom-made method for the quantisation of a real-valued signal $x_q$ to $b_q$ bits (ergo $2^{b_q}$ potential values) per sample per channel according to the following procedure:

- If $b_q = 1$, then the binarised version of the signal is derived with a simple sign function:

$$x_q = sign(x)$$

- If $b_q > 1$, then each signal is first normalised through a division by its maximum absolute value. Consequently, given the normalised signal $x_{norm}$, the quantised version is acquired through the following steps:

$$\begin{aligned} \Lambda &= 2^{b_q} \\ \chi &= \mathcal{H}(\chi + 1 - \frac{2(\Lambda - 1)}{\Lambda}) * \frac{2}{\Lambda - 1} \\ FOR \quad i &= 0 : (\Lambda - 2) \quad DO \\ \chi &\leftarrow \chi + \mathcal{H}(\chi + 1 - \frac{2(i+1)}{\Lambda}) * \frac{2}{\Lambda - 1} \\ END \quad FOR \\ x_q &= \chi - 1 \end{aligned}$$

  Where $\mathcal{H}$ denotes the Heaviside function, and both that function and the subtraction at the end apply element-wise to the input vectors.

It is demonstrated that the threshold of 50% accuracy is easily met for all quantisation levels in the point-to-point modulation recognition experiment (at -6 dB SNR for 4 bits per channel, -3 dB for 1 bit), and that, when the main signal prevails over the noise, even the most aggressive quantisation scheme (1-bit) returns more than 70% correct predictions, whereas the 4-bit scheme exhibits performance almost identical to the case where full-resolution data are used.

With the aforementioned results at hand, we proceeded to assess all subsequent problems also under the restrictive influence of quantised data.

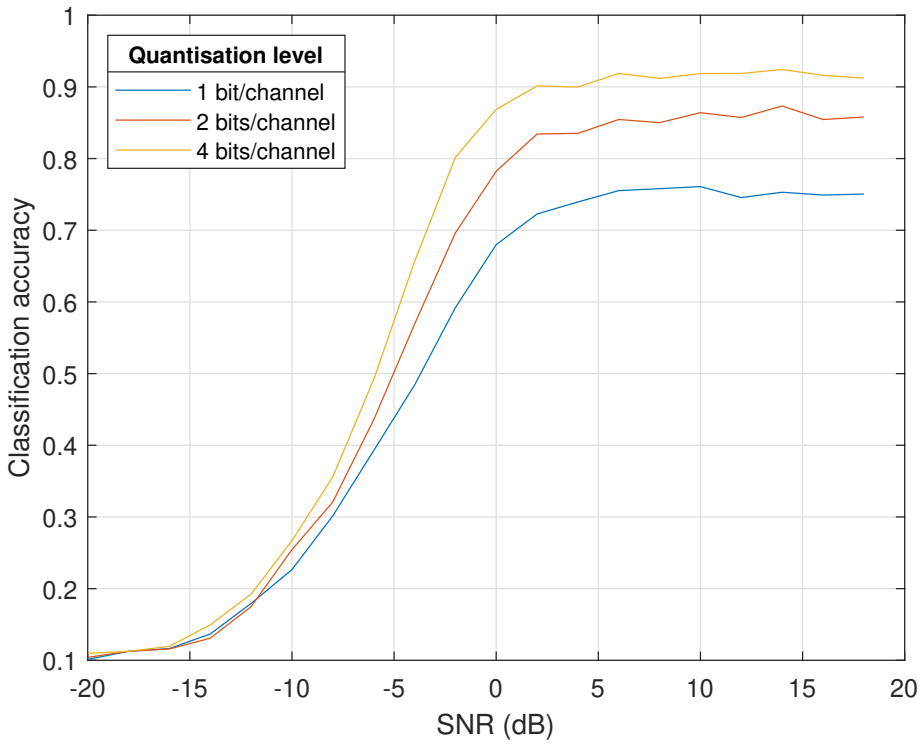


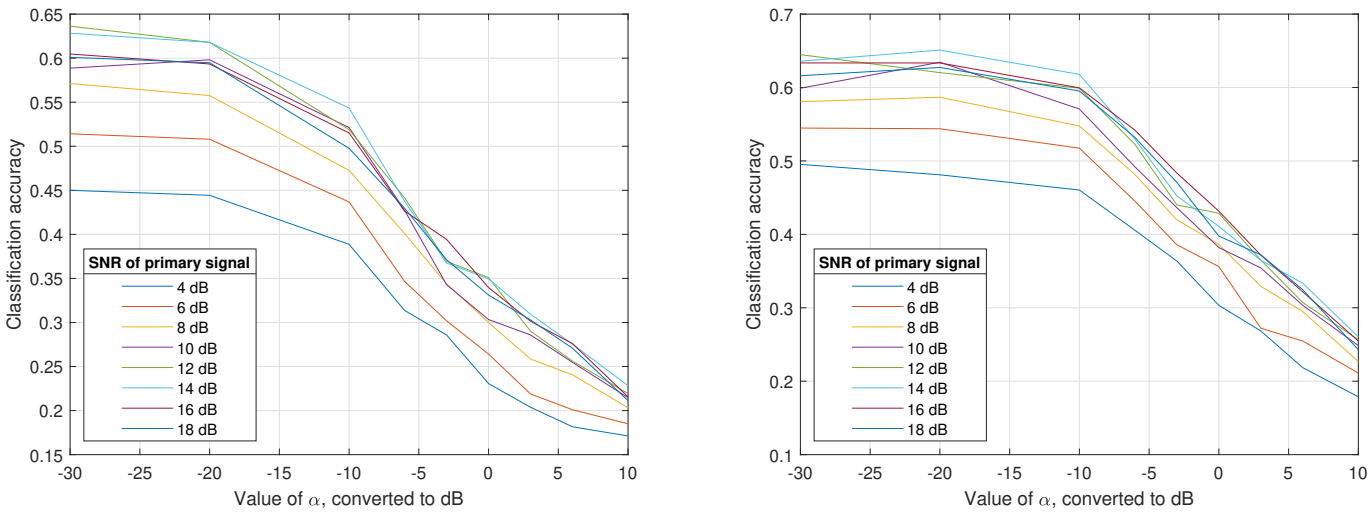

Figure 3.11: Effects of signal quantisation on modulation recognition.

### 3.6.2 Recognition with Interference

In order to render the overall quantisation process possible over many different signals, the input frames were first normalised. As before, all available classes participate in both the interference and the base signal. Let it also be noted that the quantisation is applied after the superposition of the two signals.

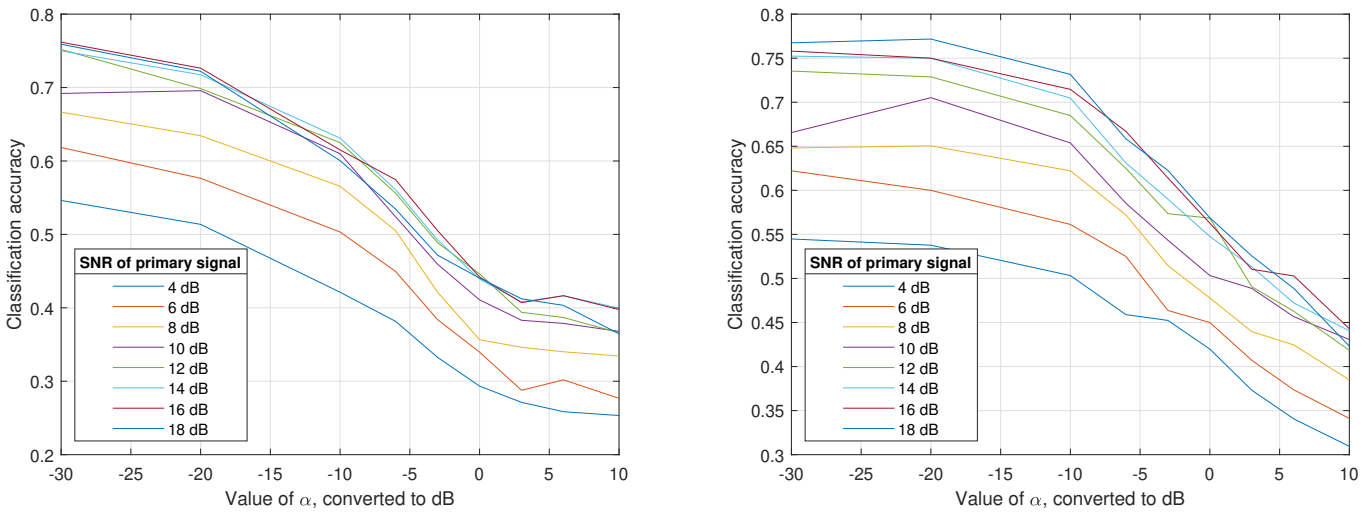
Results demonstrating a representative case of the difference between the lower- and the higher-resolution settings are presented in Figures 3.12 and 3.13, respectively. The expected significant reduction of the overall accuracy is easily noticed in all relevant cases; nevertheless, as was also observed in the point-to-point case, retention of efficiency is still high in some areas of the graphs.

The confusion matrix for the 4-bit experiment is seen in Fig. 3.14, and concerns performance across the whole spectrum of values for $\alpha$. It is evident that, even despite the quantisation and presence of interference, high probability of correct detection is still observed in some modulation schemes (esp. frequency shift keying (FSK) and amplitude modulation (AM)).




(a) Without modification (b) With modification

Figure 3.12: Effect of signal quantisation on AMC with interference - 1 bit per channel.




(a) Without modification (b) With modification

Figure 3.13: Effect of signal quantisation on AMC with interference - 4 bits per channel.

An observation of a more general nature to be made is that, despite the considerable loss of information resulting from the quantisation process, especially at the single-bit-per-channel level, the classifier still exhibits resilience to this change, and and the degradation of performance is disproportionately low compared to the degree of loss of granularity in the data. Therefore, the conclusion can be drawn that the CNN does not necessarily interpret an input signal solely as a diagram of 2D constellations, but also relies in the location of other representations and dependencies (potentially e.g. assessment of the frequency of zero-crossings, detection of

| True Class \ Predicted Class | 16QAM | 64QAM | 8PSK | B-FM | BPSK | CPFSK | DSB-AM | GFSK | PAM4 | QPSK | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 16QAM | 2380 | 3805 | 1636 | 843 | 1166 | 1152 | 845 | 786 | 1679 | 1708 | 14.9% | 85.1% |
| 64QAM | 2268 | 4419 | 1363 | 866 | 1035 | 1136 | 889 | 806 | 1799 | 1419 | 27.6% | 72.4% |
| 8PSK | 1387 | 1945 | 3448 | 855 | 1212 | 1142 | 930 | 835 | 1189 | 3057 | 21.6% | 78.4% |
| B-FM | 108 | 289 | 209 | 12527 | 329 | 954 | 298 | 668 | 392 | 226 | 78.3% | 21.7% |
| BPSK | 817 | 1226 | 1014 | 836 | 6578 | 919 | 799 | 815 | 2059 | 937 | 41.1% | 58.9% |
| CPFSK | 277 | 551 | 375 | 977 | 304 | 11556 | 293 | 848 | 466 | 353 | 72.2% | 27.8% |
| DSB-AM | 413 | 467 | 646 | 460 | 758 | 466 | 10907 | 435 | 725 | 723 | 68.2% | 31.8% |
| GFSK | 47 | 70 | 117 | 496 | 134 | 593 | 221 | 14101 | 127 | 94 | 88.1% | 11.9% |
| PAM4 | 723 | 1689 | 636 | 810 | 1370 | 925 | 709 | 799 | 7616 | 723 | 47.6% | 52.4% |
| QPSK | 1325 | 1978 | 2712 | 906 | 1220 | 1116 | 1025 | 806 | 1175 | 3737 | 23.4% | 76.6% |
| | 24.4% | 26.9% | 28.4% | 64.0% | 46.6% | 57.9% | 64.5% | 67.5% | 44.2% | 28.8% | | |
| | 75.6% | 73.1% | 71.6% | 36.0% | 53.4% | 42.1% | 35.5% | 32.5% | 55.8% | 71.2% | | |

Figure 3.14: AMC with interference: Performance by class - 4-bit-per-channel quantisation.

temporal patterns by the convolutional layers). This leads us to conclude that neural networks trained for signal recognition tasks such as AMC successfully combine the strengths of different lower-level methods, without being explicitly directed to resemble or incorporate any of them.

## 3.7 Summary

In this initial part of our research work, the simple point-to-point setup for modulation recognition which dominates most AMC literature is subjected to complication via the introduction of an interferer. This secondary device introduces an additional source of undesired influence to an extant CNN classifier, in addition to AWGN and channel impairments, and stronger signal-to-interference power ratios effect a severe impact upon detection accuracy. The incorporation of the modulation scheme utilised by the interferer as an additional input for the neural network is demonstrated to have a beneficial effect upon performance, especially in cases where the superposed signals from the two transmitters are received at power ratios approaching 1 –

the latter observation granting a first insight into the inner functions of the CNN.

The effects of signal quantisation, a technique which would be used extensively in our later research, is also first examined in this part of our work. Furthermore, here we establish our usage of a modified fully convolutional neural architecture, which eschews the traditional dense layers following the convolutional ones, in the interest of reducing the number of parameters; this architecture still remained favoured throughout the rest of our research, as it exhibited higher capabilities compared to other candidate architectures which were assessed.

# Chapter 4

# AMC with Multiple Antennas

The rapid expansion of massive MIMO communication systems, an issue which was already briefly mentioned in Section 3.6, has already commanded the attention of higher-order ML, particularly in the area of MIMO detection [JCZ19]. At the same time, some assessments have been carried out for AMC and SEI in environments with multiple receiving devices, in various modes of cooperation between the latter [HWLW19][MD12].

In this chapter, the multiple-output concept is considered in the context of AMC as a continuation of the experiments and methods laid out in Chapter 3. Our novel approach was inspired by experiments such as the ones mentioned above, and additionally concerns a variety of settings with regard to the presence of an interfering device, as well as data compression. Techniques, ideas and methodologies were retained from previous areas of study, but were accordingly updated to fit the new parameters of the problem.

## 4.1 Multiple-Antenna Setup

The setup which was conceived for this new mode of AMC does not differ significantly from the one depicted in Figure 3.1, in that it includes a transmitter and receiver pair, possibly an interfering device, and is characterised by models which include environmental degradation

and noise. In this case, however, the receiving device is chosen to feature multiple receiving antennas, and as such, to receive multiple versions of the same signal at each iteration.

The combination of information from multiple receivers for signal recognition problems is a step which can be implemented at any one of the different stages of a DL-based AMC system. In cases where several NN inputs run in parallel to each other, this step can be positioned at an intermediate level inside the network (feature fusion), after the output of the softmax layers (confidence fusion), or even after the classification output (voting-based classification) [HWLW19][ZQCY19] – these considerations were concerned in later parts of our work.

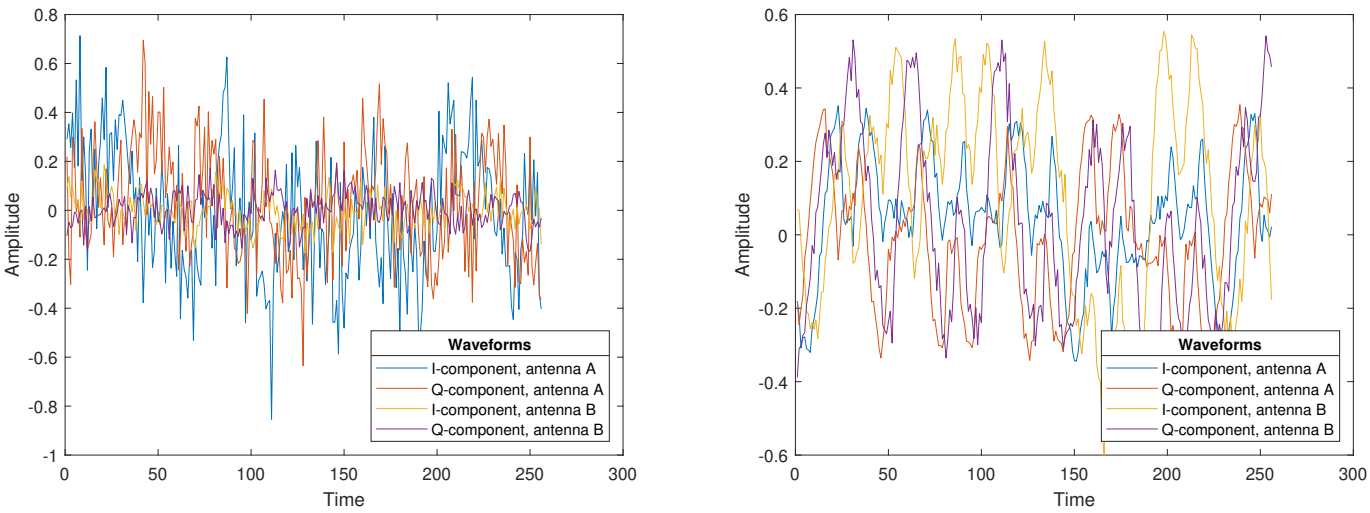
Alternatively, a single NN classifier can be trained instead if the raw signals from the receivers are combined to form a single input, and the latter is the approach which is followed in our experiments (given that the setup implies a single receiving device with multiple antennas, and not any degree of significant geographic separation between different receiving ends).

Using the built-in MIMO channel models of the MATLAB communications toolbox, we model our receiver as the output of a single-input multiple-output (SIMO) setup with the following parameters:

- $N_t = 1$ transmit antennas
- $N_r = 10$ receive antennas
- 1024 transmitted symbols per frame (transients not removed at data collection stage)
- Sampling rate $f_s = \frac{1}{T_s} = 200kHz$
- 8 digital modulators (BPSK, QPSK, 8PSK, PAM4, 16QAM, 64QAM, CPFSK, GFSK) and 3 analogue (SSB-AM, DSB-AM, B-FM)
- Rician fading distribution with path delays $[0,\ 1.1,\ 3,\ 5.5] \times T_s$ and corresponding gains $[0, -2, -3, -8]dB$, as in the SISO case

Data creation for the new category of experiments follows the same process as the one described in Section 3.2. According to the data limitations imposed by each separate sub-problem, different dataset sizes are considered; as it is made clear in the following sections of the current report, smaller volumes of data did not necessarily translate into degradation of the results.

The experiments described in this chapter were carried out utilising a relatively small dataset, consisting of 80960 frames for training and 20240 for testing, equally distributed between the 11 participating signal classes.




(a) 0 dB SNR (b) 18 dB SNR

Figure 4.1: Representative signals from SIMO AMC dataset.

### Noisy channel behaviour

Fig. 4.1 presents some sample plotted signals from the generated SIMO dataset, so as to demonstrate the effect of added noise. The plots depict frames selected from the waveforms of two different receivers, for the extremes of the SNR value spectrum (0 and 18 dB, respectively), both from the CPFSK modulation class. It becomes obvious that the presence of AWGN severely degrades signal quality, as the 0dB samples are hard for the naked eye to distinguish from noise, while for 18dB the randomisation is clearly reduced, and parts of the periods of the carrier signal are easily distinguishable, even though certain edges and spikes are still visible.

The case of point-to-point communication in the conditions described above is examined under

the influence of posterior incorporation of additive white Gaussian noise (AWGN) through the MATLAB noise model, as this attribute is not incorporated in the MATLAB MIMO channel functions. SNR levels for this were set to range from 0dB to 18dB, bearing in mind that most applications shall operate within that range (it is noted that an SNR $\geq 6dB$ is considered typical in the majority of wireless communication setups [TBLS19]). As with the case of SISO modulation detection described earlier, SNR values were distributed uniformly among the discrete values of the aforementioned spectrum, namely from $0dB$ up to $18dB$ in increments of $2dB$.

### 4.1.1 Simulation Setup and Results

In each experiment, the execution of the algorithms follows the isolation of a subset of the original data, in that the number of receptions from which data is utilised for classification is limited, as is the number of time samples. Due to GPU capacity limitations, the length of each frame was limited to $N = 256$ samples for almost all different iterations, while the participating antenna count ranged from the minimum to the maximum values available (1 to 10); the only exception with regard to input vector length was the case of 10 antennas, where only 200 time samples were isolated initially due to memory limitations, and further ahead it was observed that longer vectors only served to increase training time, but not accuracy in the specific case.

The fully convolutional architecture (as visualised in Fig. 3.9) is still preferred, as it resulted in the best classification performance among all candidate classifiers. The kernel dimensions in the FCNN architecture were accordingly increased to accommodate data from more channels. Training is always carried out on the whole subset, i.e. not separately for each SNR value.

A summary of the results from the experiments described above is presented in Fig. 4.2, with reference to the number of receive antennas used, and to the values of SNR. Though the expected gradual improvement of performance over the increasing SNR is present, a point of practical importance is that the participation of a greater number of antennas in the classification process can lead to significant improvements in the final quality of the algorithm, albeit with a tradeoff in time always applying (as shall be explained further ahead). As an example, we note that

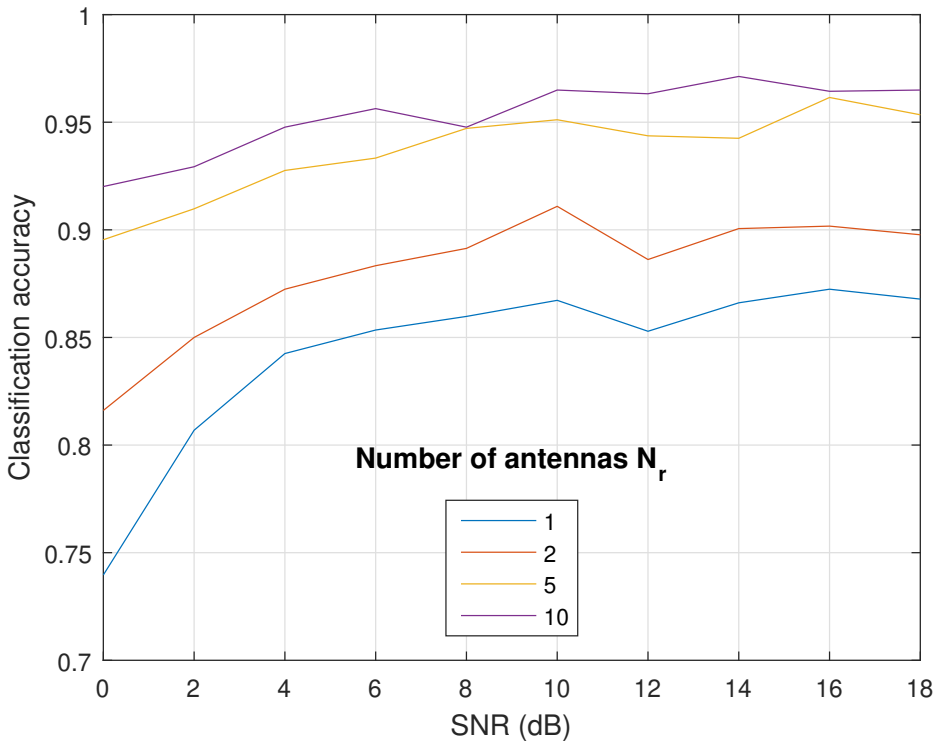



Figure 4.2: Modulation classification in multiple-antenna setup.

an upgrade from one antenna to five results in acccuracy improvements ranging from 10% to 15%, depending on the SNR region in question. Algorithms where data are drawn from large numbers of antennas reach saturation status with greater ease. Classification times increase with the number of participating antennas, varying between 64 $\mu s$ per frame for 1 antenna and 824 $\mu s$ for 10.

Fig. 4.3 is the confusion matrix concerning the best-case scenario in the experiment, namely when all 10 antennas are being used for classification. It should be pointed out that most of the 11 modulation classes exhibit excellent performace throughout the spectrum of SNR values.

Similarly to previous experiments, there persists the problem (mostly in regions of higher noise power) of the mutual confusion of the two existing QAM classes, with 64QAM being much less robust. As an attempt at rectifying the problem, a separate neural network (though with the same general structure and hyperparameters) was trained on the QAM data alone, on data from all antennas, yet that solution was proven insufficient, as it only results in a detection accuracy of 59.5% for 16QAM and 63.6% for 64QAM, vastly inferior to the previously noted

performance.

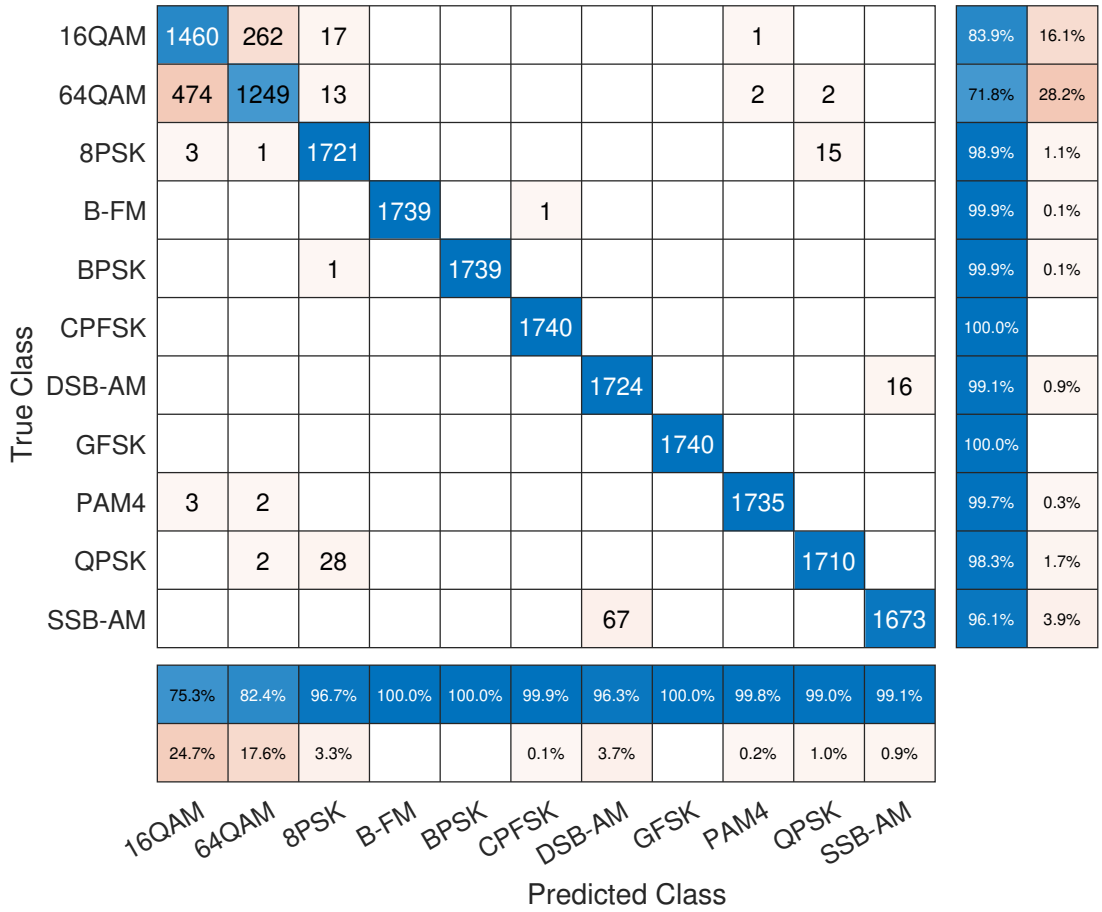



Figure 4.3: SIMO AMC performance by class - 10 receive antennas.

Nevertheless, other areas of possible confusion, such as between different members of the AM or PSK groups, have had their classification difficulty mostly rectified through the adoption of multiple signals as classification inputs.

## 4.2 Implementations of Data Compression

### 4.2.1 Low-Resolution Quantisation

With regard to the normalisation of data and overall quantisation method, the procedure is identical to the one described in Section 3.6. All other parameters are retained as described in the SIMO experiments described so far.

Fig. 4.4 summarises the results of the SIMO AMC experiment with the input data subjected to quantisation of 1 bit per channel per time sample. Although the loss of information resulting from the quantisation process manifests in severe degradation of quality, the leveraging of data from additional antennas acts as a balancing factor, always resulting in significant gains. 5- and 10-antenna setups exhibit very early saturation points still, and it is also worth noting that the performance in these two cases is superior to the single-antenna setup even without data compression. This observation serves to illustrate the benefits of reception diversity in detection accuracy; receiving single-bit representations of information from multiple antennas proves itself more useful than full-resolution data from a single antenna.

It should also be noted that classification times, while remaining largely unaltered for the SISO case, are now reduced to 683 and 772 $\mu s$ per frame for 5 and 10 antennas respectively, still offering competitive results while operating on heavily compressed data.

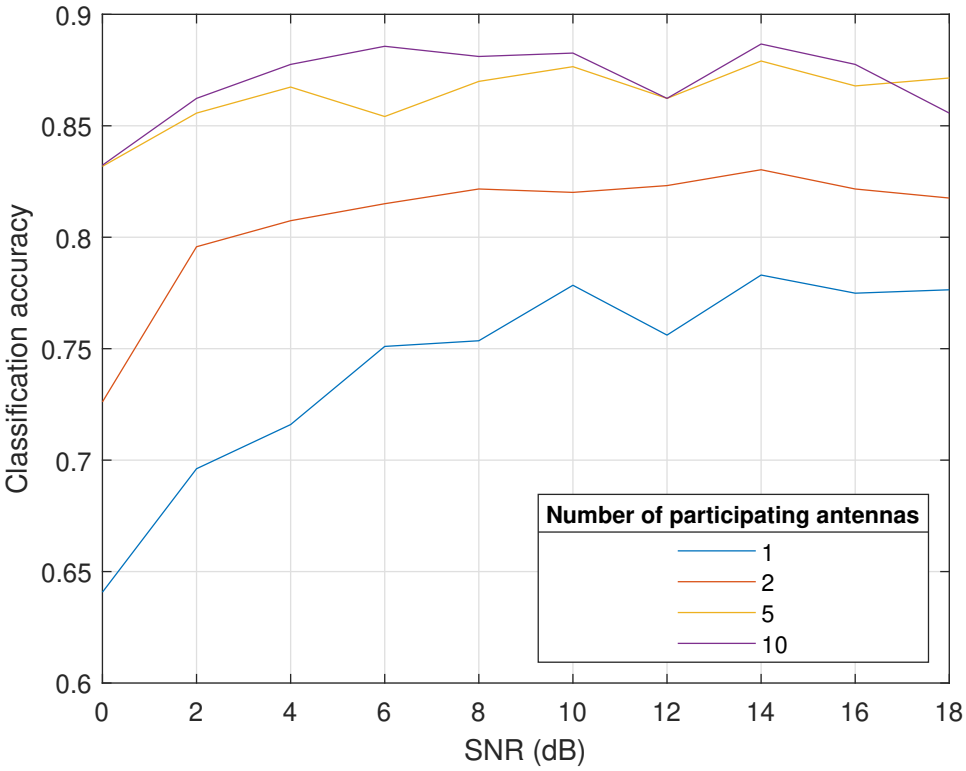



Figure 4.4: Multiple-antenna AMC - 1 bit per channel quantised data.

Experimentation continued with cases of 2-bit-per-channel quantisation of the input signals, yet no significant improvement was noted in the new results, thus the accuracy curves for those

cases are not included. Furthermore, certain small inconsistencies noted in the results (note e.g. that in some of the curves in Figure 4.4 are not monotonically increasing, as we should expect the curves to be when using a larger and more diverse dataset, and exhibit undulations), are expected to the degree that they manifest, due to the small dataset size, but were further amplified when the signals were represented in their 2-bit quantised versions.

### 4.2.2 Uniform Subsampling

The issue of the training duration of ANNs has posed a consistent challenge for signal identification algorithms based on machine learning over the years, in that training times are often proven prohibitively long for real-life deployments, especially if any degree of re-training is deemed necessary [RDAS+20]. This can easily pose significant problems, especially in settings predicted to become widespread in 5G networks [RJY+19], where classifier modules might have to be re-trained at frequent intervals due to volatile environmental variables.

A data compression technique which has been proposed in relevant literature for reducing training times and data sizes in the context of time-series classification algorithms is uniform subsampling. In contrast to binarisation and quantisation in general, the idea behind this method is the reduction of the dimensionality of input data, rather than an implementation of lower resolutions. Despite appearing a rather simplistic method at face value, it has been observed to yield results comparable to (or in some cases even better than) principal component analysis (PCA), at only a fraction of the computational cost [RJY+19].

By definition, uniform subsampling of a time-series dataset of dimensions $M \times N \times T$, with $T$ corresponding to time samples, means that the range $1, \ldots T$ is sampled at constant intervals of length $k$, thus making the classifier operate upon a smaller set of $M \times N \times \frac{T}{k}$.

For the experiments detailed henceforth, subsampling factors which have been tried are $k = 2, 3, 5$.

**Results**

The summary of results for uniform subsampling over all different numbers of receive antennas is presented in Table 4.1.

Table 4.1: Classification accuracy and time intervals ($\mu s$ per frame)

| **Number of antennas** | **1** | **2** | **5** | **10** |
|---|---|---|---|---|
| No subsampling | **84.3%**(64) | **88.2%**(132) | **93.6%**(762) | **95.2%**(824) |
| Subsampling $\frac{1}{2}$ | **76.4%**(41) | **81.7%**(57) | **88%**(306) | **91.9%**(430) |
| Subsampling $\frac{1}{3}$ | **75.8%**(25) | **81.2%**(39) | **86.9%**(199) | **89.3%**(210) |
| Subsampling $\frac{1}{5}$ | **68.1%**(15) | **73.4%**(21) | **80.2%**(112) | **85.2%**(163) |

The improvement with regard to classification time per frame is apparent in all cases, and while the acceleration of the process is sub-linear for a single antenna, and largely linear for 10, the two intermediate cases (2 and 5 antennas) benefit even more and exhibit a supralinear relation between input dimensions and operation time.

Additionally, it is demonstrated that increasing the number of antennas seems to partly rectify the effect of dimensionality reduction upon detection accuracy; this is made clear through a comparison of most of the individual cases presented in the table, and at the extremes we notice that an accuracy difference of 16.2% between complete dataset usage and $\frac{1}{5}$ subsampling in the single-antenna case is gradually reduced through the leveraging of multiple antennas, reaching 10% when the full dataset is utilised.

In general, through both methods of dimensionality reduction discussed above, various degrees of trade-off are made available for potential usage in a variety of cases, e.g., according to whether detection reliability is desirable at the cost of higher running times.

## 4.3 Multiple-Antenna AMC with Interference

Consecutively, the equivalent setup to the one presented in Section 3.1, i.e. with interfering devices creating confusion and degrading the signal quality, was also tested for an environment

where multiple antennas are available at the reception stage. Overall, the general process and parameters were retained, as described in Section 4.1, and 256 consecutive timestamps were selected again for each instance of the dataset.

A setup with data sizes comparable to the ones previously considered (Section 4.1) was tested initially, but it resulted in unexpectedly erratic results, with very high variances, and performance exhibiting certain unprecedented behaviours as a function of SNR and signal-to-interference ratio (SIR). This was attributed to the fact that, for experiments dealing with interference as well, each point in the graph represents a subset based on an intersection of both SNR and SIR, and is thus reliant on very small sets (ca. 280 per plot point for validation).

As such, dataset creation scripts were run anew on MATLAB for the interference classification data, yielding $3 \times 10^5$ frames in the training set and $7.5 \times 10^4$ in the validation set. A drawback resulting from this is that the training process based on these datasets requires several hours to be completed.

### 4.3.1 Recognition of single unknown signal

Figure 4.5 demonstrates the results of this experiment as a function of the number of antennas from which data are drawn for classification. For the purpose of simplifying the diagram, which would otherwise have to include no less than 30 curves, the values were averaged over all potential SNR values for each of the three curves which are presented here. The accuracy metrics are 57.6%, 65%, and 71.4% (corresponding to the blue, orange, and yellow curves, respectively, in Figure 4.5). The improvement noted as a consequence of receiver diversity is far from negligible, even under the presence of just one additional antenna, whereas for 5 antennas optimal performance is reached, for a minimum accuracy of 57% in the lowest SIR region ($\alpha = 10$), which is approximately 6.3 times above random classifier performance.

In Fig. 4.6, we detail the results in the case of using 5 antennas, with regard to the SNR. It is observed through comparison to Fig. 3.10a that a significant gain is achieved not only in the

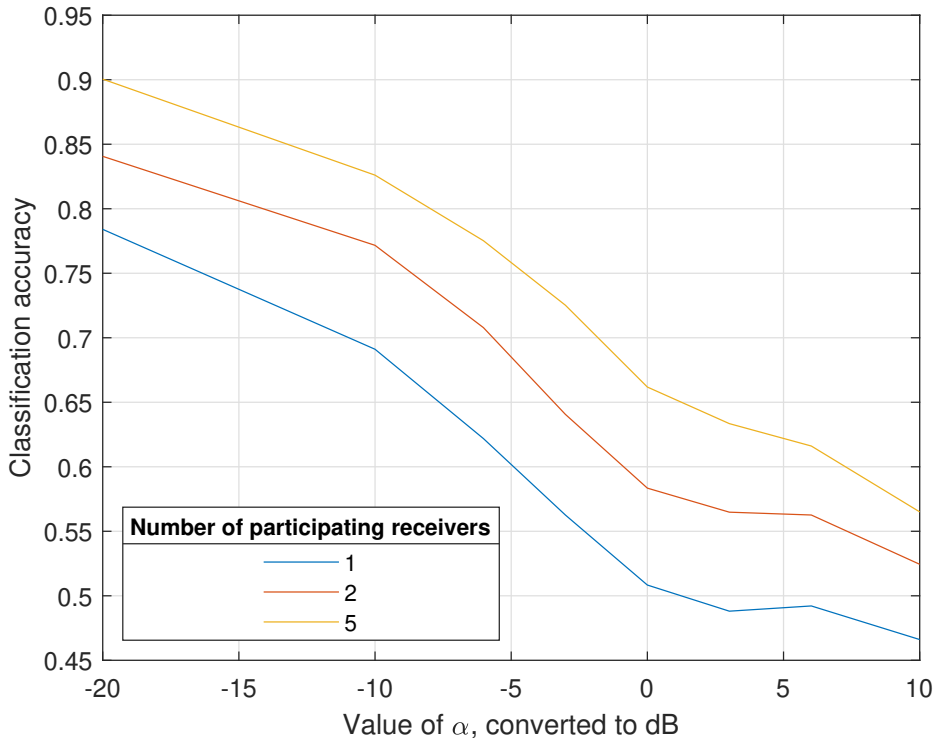



Figure 4.5: AMC with interference - Multiple antennas.

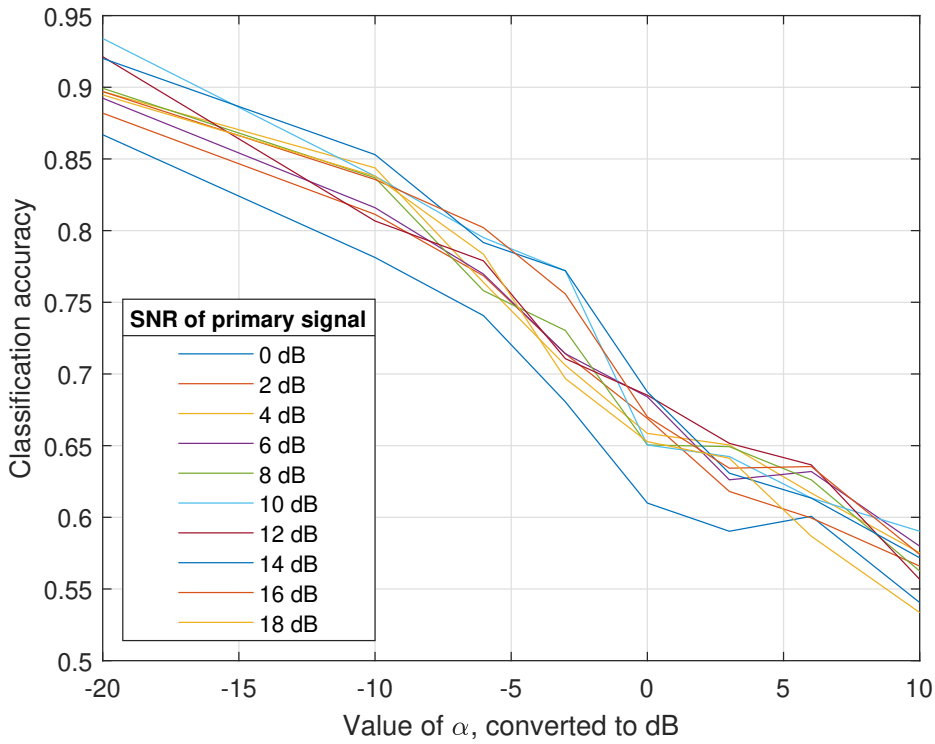



Figure 4.6: AMC with interference, 5 antennas - Effect of SNR.

improvement of general performance, but also in the reduction of the distance between curves describing different SNR conditions (i.e., lower-SNR data does not perform significantly worse here); whereas in our earlier tests, extreme values of differences between curves would reach even 30%, the curves in the multiple-output environment are located much closer and their divergence never exceeds 10%.

The practice, as established earlier, of the simple appendage of a one-hot encoding of the secondary signal's modulator to late-stage layer outputs of the network seemed to lose its edge, however, for this new multiple-output setup (the gains observed were less than 1% overall). An explanation for this phenomenon might be found in the theory that most of the improvements are from information already present due to the diversity of "views" of a signal from different antennas, which limits the amount of information that the aforementioned one-hot information vector method can offer, and thus its contribution is of little value.

In an attempt to rectify this drawback, and rely less on a brute-force sort of simplicity with regard to the incorporation of the additional modulation information, we implemented an additional convolutional layer with a leaky ReLU activation function accepting the one-hot vector as input and outputting a vector of 11 elements, which is then concatenated with the output of the global average pooling layer.

The outcome of this attempt is visualised in Fig. 4.7. Although general gains were not as noticeable as e.g. the effect of leveraging data from more antennas, a performance improvement still manifests as a result of the proposed modification, primarily in regions around 0dB SIR, and slightly more pronounced in cases of low SNR. With regard to the distance between different curves, no significant difference in variance is observed. Additionally, in the case of the multiple-output environment, the abrupt performance degradation noted around the 0dB SIR area, as observed in previous experiments, is far less pronounced.

It also bears mentioning that the process of incorporating the one-hot vector into the classifier appears to have no significant bearing upon classification time, which stays constant at $761\mu s$ per frame under the usage of 5 antennas.

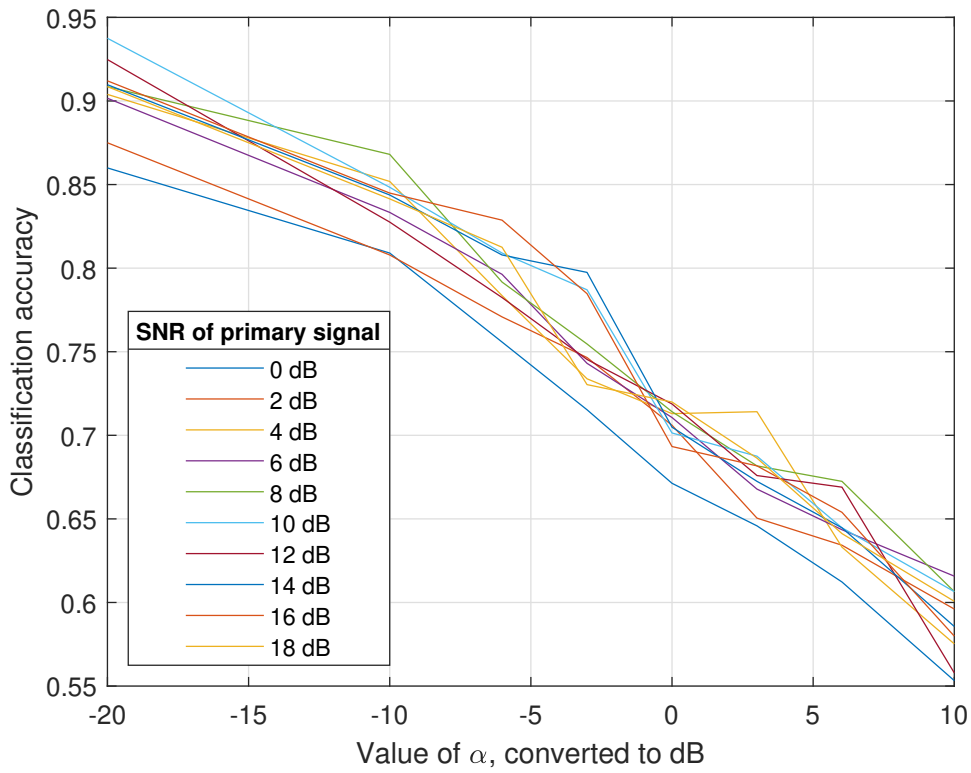



Figure 4.7: AMC with interference, 5 antennas, modified net - Effect of SNR.

In Fig. 4.8, the influence of different classes upon classification performance is demonstrated, and the effect of the network modification is also particularised with reference to each separate modulation class. It is evident that, apart from the already acknowledged problem of QAM, which is rendered even more pronounced under the influence of interference, the PSK modulation family also begins to struggle, as does the pulse amplitude modulation (PAM) scheme, yet they seem to benefit most from the incorporation of the external information vector, while even the classes which either way perform well also exhibit incremental improvements. The robustness of FSK and analogue schemes is retained to a high degree, in spite of the concurrent influence of interference and noise.

### Uniform subsampling

Similarly to the procedure described in Section 4.2.2, the data for 5-antenna input were subjected to a uniform subsampling process retaining every second time sample from the initial I/Q vector, and the altered data were classified both with and without knowledge of the class

| True Class \ Predicted Class | 16QAM | 64QAM | 8PSK | B-FM | BPSK | CPFSK | DSB-AM | GFSK | PAM4 | QPSK | SSB-AM | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 16QAM | 2492 | 1814 | 682 | 41 | 194 | 47 | 14 | 22 | 414 | 765 | 15 | 38.3% | 61.7% |
| 64QAM | 1946 | 2333 | 685 | 42 | 228 | 38 | 7 | 11 | 459 | 743 | 8 | 35.9% | 64.1% |
| 8PSK | 708 | 650 | 3214 | 51 | 286 | 41 | 18 | 14 | 321 | 1190 | 7 | 49.4% | 50.6% |
| B-FM | 30 | 11 | 19 | 6060 | 28 | 190 | 13 | 67 | 31 | 30 | 21 | 93.2% | 6.8% |
| BPSK | 215 | 173 | 229 | 41 | 4540 | 43 | 27 | 26 | 848 | 348 | 10 | 69.8% | 30.2% |
| CPFSK | 14 | 7 | 13 | 139 | 9 | 6264 | 2 | 24 | 11 | 14 | 3 | 96.4% | 3.6% |
| DSB-AM | 10 | 5 | 10 | 10 | 15 | 2 | 5917 | 4 | 15 | 17 | 495 | 91.0% | 9.0% |
| GFSK | 5 | 8 | 4 | 50 | 22 | 15 | | 6376 | 10 | 6 | 4 | 98.1% | 1.9% |
| PAM4 | 303 | 314 | 175 | 58 | 829 | 33 | 11 | 17 | 4420 | 334 | 6 | 68.0% | 32.0% |
| QPSK | 632 | 502 | 982 | 49 | 332 | 56 | 17 | 17 | 312 | 3591 | 10 | 55.2% | 44.8% |
| SSB-AM | 8 | 11 | 9 | 10 | 10 | 5 | 583 | 10 | 11 | 5 | 5838 | 89.8% | 10.2% |
| | 39.2% | 40.0% | 53.4% | 92.5% | 69.9% | 93.0% | 89.5% | 96.8% | 64.5% | 51.0% | 91.0% | | |
| | 60.8% | 60.0% | 46.6% | 7.5% | 30.1% | 7.0% | 10.5% | 3.2% | 35.5% | 49.0% | 9.0% | | |

(a) Without modification

| True Class \ Predicted Class | 16QAM | 64QAM | 8PSK | B-FM | BPSK | CPFSK | DSB-AM | GFSK | PAM4 | QPSK | SSB-AM | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 16QAM | 2461 | 1900 | 869 | 27 | 194 | 39 | 5 | 9 | 353 | 643 | | 37.9% | 62.1% |
| 64QAM | 1756 | 2643 | 847 | 16 | 224 | 38 | 5 | 4 | 383 | 584 | | 40.7% | 59.3% |
| 8PSK | 629 | 683 | 3593 | 13 | 208 | 33 | 3 | 5 | 257 | 1075 | 1 | 55.3% | 44.7% |
| B-FM | 15 | 16 | 16 | 6237 | 21 | 135 | 2 | 22 | 17 | 15 | 4 | 96.0% | 4.0% |
| BPSK | 131 | 150 | 231 | 19 | 4800 | 15 | 6 | 14 | 822 | 308 | 4 | 73.8% | 26.2% |
| CPFSK | 17 | 7 | 11 | 68 | 9 | 6351 | | 5 | 13 | 18 | 1 | 97.7% | 2.3% |
| DSB-AM | 2 | 8 | 8 | 1 | 5 | 1 | 6066 | | 13 | 8 | 388 | 93.3% | 6.7% |
| GFSK | 2 | 2 | 3 | 10 | 9 | 2 | 1 | 6465 | 2 | 4 | | 99.5% | 0.5% |
| PAM4 | 255 | 283 | 177 | 16 | 770 | 25 | 8 | 8 | 4739 | 217 | 2 | 72.9% | 27.1% |
| QPSK | 551 | 515 | 1134 | 21 | 280 | 40 | 9 | 9 | 285 | 3654 | 2 | 56.2% | 43.8% |
| SSB-AM | 1 | 2 | 1 | 2 | 1 | | 668 | 1 | 4 | | 5820 | 89.5% | 10.5% |
| | 42.3% | 42.6% | 52.1% | 97.0% | 73.6% | 95.1% | 89.6% | 98.8% | 68.8% | 56.0% | 93.5% | | |
| | 57.7% | 57.4% | 47.9% | 3.0% | 26.4% | 4.9% | 10.4% | 1.2% | 31.2% | 44.0% | 6.5% | | |

(b) With modification

Figure 4.8: IC performance by class - 5 antennas

of the secondary signal. Fig. 4.9 demonstrates the related results.

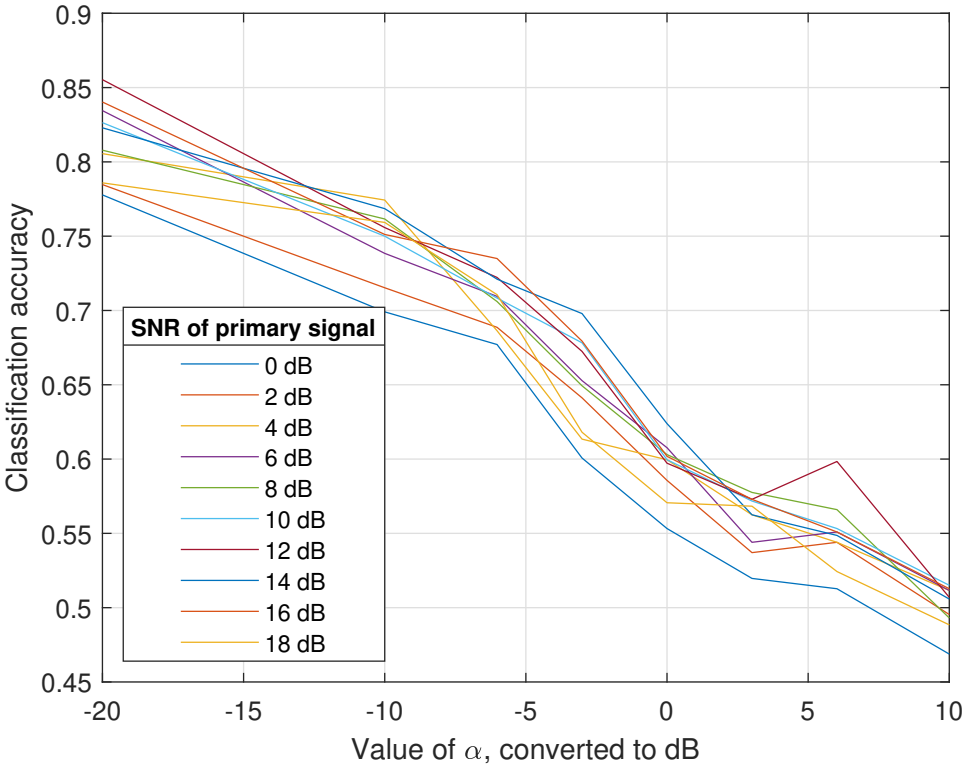



Figure 4.9: Effect of uniform subsampling on IC

As with the previously examined case, one may clearly notice the overall performance degradation, which is nevertheless not severe (about 6-7% general difference), whereas classification time is reduced to 303 $\mu s$ per frame, less than half the original time.

## 4.4 Summary

In the experiments detailed in this chapter, a new setting which was considered in addition to the established AMC paradigm, in which the receiving device features multiple antennas and thus has access to data obtained for each signal via many realisations of the communication channel. Both the traditional setup (one device at the transmitting and one at the receiving end) and one with an additional interferer are tested, and the cooperative approach for modulation recognition demonstrates a clear potential, resulting in significant gains in classifier accuracy. Issues of neural network operation times are also taken into account, and uniform subsampling

introduced as a method of speeding up the overall classification operation.

# Chapter 5

# Distributed AMC

The results presented in Chapter 4 highlighted the benefits of information diversity in the accuracy of AMC algorithms. In practice, nevertheless, multi-antenna setups might not always be available for detection. Additionally, the signals received by different antennas of the same receiver might be correlated, thus limiting the benefits which can result from such an approach. On the other hand, it has been demonstrated that multiple single-antenna receivers can cooperate in order to achieve at least part of the benefits of a multi-antenna receiver [SEA03].

Inspired by the above, in this chapter we proceed to study distributed AMC, where multiple single-antenna receiver units collaborate over rate-limited backhaul links to detect the modulation type of a wireless communication signal.

## 5.1 Conception of a geographically distributed AMC system

We consider a system in which a single transmitting entity is emitting data streams modulated according to the $C = 11$ modulation schemes mentioned in Section 3.5. A set of $R = 4$ receiving devices are positioned at the vertices of a square area with a side of length $L$, and the originating device is presumed to transmit from any point within the square, with the exception

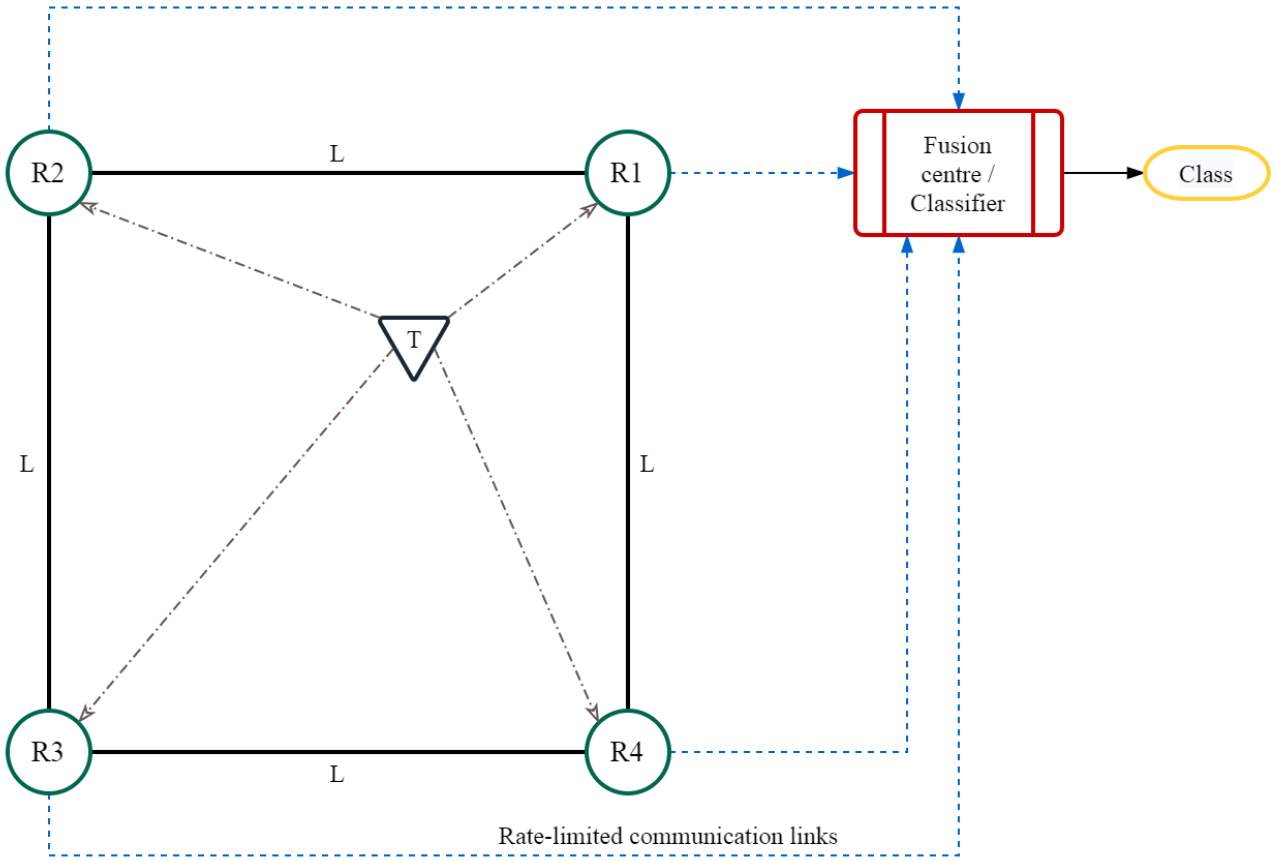



Figure 5.1: Geographically distributed AMC system.

of four quadrants near the vertices, defined such that a minimum transmitter-receiver distance of $d_{min}$ is kept at all times. For the sake of certain settings of the experiment, the edge devices may be equipped with hardware allowing for the training of one signal classifier each, which is trained off-line. Fig. 5.1 is a schematic representation of the system considered in the currently described experiments.

Instead of the usual method of modelling transmission paths with Rician or Rayleigh fading channels, the multipath fading paths in the experiment described here were chosen to be modelled in the context of a single given geographical area, with the simulation thereof including a common set of scatterers, reflectors, and other environmental factors influencing the integrity of the signal (in addition to the presence e.g. of thermal noise). For the setup considered in the following experiments, and without loss of generality, the transmission procedure may be described as such:

$$\mathbf{r}_i(t) = Ae \sum_{p=0}^{N_{pa}} \mathbf{h}_p^T \sum_{k=0}^{K-1} \mathbf{x}_k^m p(t-(k+c)T_{sym}) + \mathbf{A}_{n,i} w(t) \qquad i = 1, \ldots, R \tag{5.1}$$

where $\mathbf{r}_i$ is the vector of continuous I/Q values reaching receiver $i$, $R$ is the number of receivers, $A$ is an amplitude value modelling the transmission power, $N_{pa}$ is the number of different paths consisting of the line-of-sight (LOS), if applicable, and the non-LOS paths (e.g. due to reflectors), with each path resulting in a fading channel with coefficients $\mathbf{h_p}$ (we can safely consider that the manifestation of delays associated with each path is implicit in the channel coefficients), $x_k^m$ is the k-th symbol out of K which are to be transmitted during a given time slot, drawn from the constellations of modulation scheme $m$, $p(t)$ is the pulse-shaping function, $c$ is the timing offset, $T_{sym}$ is the period of a single pulse, and $w(t)$ is additive white Gaussian noise, with $\mathbf{A}_n$ being an amplitude vector modelling noise power at different receivers; in practical terms, we have decided to model this as Johnson-Nyquist noise, which means that its equivalent power is calculated as:

$$P_{noise} = k * T_0 * BW, \tag{5.2}$$

where $k = 1.38 * 10^{-23} \frac{J}{^\circ K}$ is Boltzmann's constant, $T_0$ is the ambient temperature, and $BW$ is the channel bandwidth.

For purposes of local data storage in digital format, and for the further usage of the data, the waveforms are sampled at equal intervals:

$$\mathbf{r}_i[n] = \mathbf{r}(nT_s), \quad n = 1, \ldots, N, \tag{5.3}$$

where $N$ is the number of symbols retained for each frame, and $f_s = \frac{1}{T_s}$ is the sampling frequency.

An additional novel feature of the system, taking relevant state-of-the-art literature into account, is that the classifying entity is not identical to any one of the receivers, but is instead considered independent. That is, the neural network which is available for classification is situated, either in part or in whole, at a remote fusion centre which does not coincide with any receiver, but is instead connected to each of them via a backhaul link which may potentially be subject to bandwidth restrictions. These conditions are considered well within the possible parameters of a real-life telecommunication system.

As such, the received frames $\mathbf{r}_i[n]$ are presented to the fusion centre in alternative (typically compressed) forms:

$$\mathbf{r_{fusion}} = \{\mathcal{G}_1(\mathbf{r_1}), \ldots, \mathcal{G}_R(\mathbf{r_R})\}, \tag{5.4}$$

where the functions $\mathcal{G}_i$ represent the format conversion and size modification which is imposed by the different edge users upon the data before transferring them to the fusion centre. As such, each receiver sends over its respective link a finite-resolution quantity (thus a bitstream) $b_i$ in the following manner:

$$\begin{aligned} \mathbf{b}_i &= \mathcal{G}_i(\mathbf{r_i}[1], \ldots, r_i[N]) \\ \mathcal{G}_i &: \mathbb{R}^N \mapsto \{0, \ldots, 2^\rho - 1\} \end{aligned} \tag{5.5}$$

These functions can assume many different forms, including, but not limited to:

- The identity function (in which case the central node has full access to all data, ergo when $\rho$ is not limited).

- A quantisation (resolution reduction) function.
- A compression function, possibly deriving a latent representation of the signal.
- Combinations of the above

Consequently, the fusion centre, having received the bitstreams $m_i$, is tasked to map their combination to one of the $C$ classes via a classifier $F$ parameterised by $\theta$, arriving at an estimation of the identity of the received signal:

$$\begin{gathered} \hat{m} = F(\mathbf{b}_1, \ldots, \mathbf{b}_R; \theta), \\ F : \{0, \ldots, 2^\rho - 1\} \mapsto \{1, \ldots, C\} \end{gathered} \tag{5.6}$$

With the above in mind, a $(C, R, N, r)$ code in this case consists of $R$ encoding functions $\mathcal{G}_1, \ldots, \mathcal{G}_R$ deployed at the sensors, and a decoding function $F$ at the fusion centre, which attempt to identify, in a collaborative manner, the correct modulation scheme of a signal from $N$ I/Q samples at each receiver, using information communicated to the fusion centre using $r$ bits per receiver per instance. Optimisation is achieved through minimising the misclassification probability:

$$\underset{\mathcal{G}_1, \ldots, \mathcal{G}_R, F}{\arg\min} \, Pr\{\hat{m} \neq m\} \tag{5.7}$$

The randomness element affecting this error probability arises in great part from the fact that, in the chosen modelling of the system, the location of the transmitter changes at each trial, in addition to the presence of noise and channel fading.

### 5.1.1 Data reduction strategies and their assessments

As was pointed out previously, the fusion centre can be trained to take advantage of different kinds of data obtained from the receiving devices in order to reach a decision with regard to the identity of a detected signal; by way of example, it may have access to full-resolution data, reduced-resolution versions of the same, intermediate representations, etc. Because the situation regarded in the present work is one where the volume of data transmitted over the information propagation channels is an important factor, we are concerned not merely with maximising the classification accuracy and accordingly optimising the detection algorithms, but rather also with the trade-off between the overall performance and the volume of data communicated from each sensor to the classifying node. As such, in the context of cooperative modulation recognition as described above, we aim to explore potential methods of data reduction and how they might impact the performance of different detection algorithms.

Consequently, alternative schemes which are considered for studying the issue of the restriction of rate $\rho$ are summarised in the following list:

1. Full resolution scheme (constituting the initial benchmark): infinite capacity on backhaul. In this scenario, the fusion centre has access to the entirety of the signals recorded at the receivers, and thus is able, in theory, to reach the best-possible result, since it relies on full-resolution data from all sensors to reach a decision (this is essentially identical to the setting detailed in Section 4.1). In such a case, the data transferred per receiver shall total $\rho_{opt} = 2 \times N \times \rho_{res}$ bits, where $N$ is the length of the time sequence, and $\rho_{res}$ is the highest resolution level available.

2. Scalar quantisation scheme. As a simple means of reducing data volume, the signals are presented in reduced-resolution versions via uniform quantisers. Each sample (that is, the value of the signal for a specific timestamp and channel) is quantised to resolution $\rho_{low}$, and as such, can assume only $2^{\rho_{low}}$ different values. The transferred data load becomes $\rho_q = 2 \times N \times \rho_{low}$ bits.

3. Voting based on local decisions. In this approach, local classifiers are trained at each

receiver, and the fusion centre reaches its final decision based on the local decisions, *i.e.* votes, from all sensors. This can be achieved with a method as simple as majority voting, or even in a more sophisticated manner, *e.g.*, with a simple perceptron operating upon the individual votes received from the sensors, potentially allowing also for the implementation of weighting decisions from different receivers. Regardless of the particular method employed by the centre, this setting would correspond to communication links with capacity $\rho_{vote} = \lceil log_2 C \rceil$ bits, where $C$ is the number of modulation classes.

4. Feature-based distributed detection. Here, the central node receives intermediate $\rho_{lat}$-bit representations of the signal resulting from the local classifiers, which shall essentially function as feature vectors. The decision is reached with the help of a shallowed-out classifier, and the required backhaul link capacity becomes $\rho_{lat} = L_{vec} \times \rho_{res}$ bits when the length of the feature vector is $L_{vec}$.

5. Compressed feature representation and detection. The feature representation based detection in the above scheme can be considered a lossy compression approach, where the received noisy samples at each sensor are projected onto a lower-dimensional space while enabling optimal detection accuracy at the fusion centre. However, it can be safely assumed that the latent representations at the sensors may have additional redundancy, and can be further compressed using a lossless compression algorithm with the aim of reducing the communication volume to $\rho_{comp}$ bits, and the central node decides based on the reconstructions of the same. The transmitted data per receiver will be $\rho_{comp} = \mathcal{B}(\mathcal{CF}(\mathbf{v}, \zeta))$ bits, namely a function of the content of the feature vector $\mathbf{v}$ and the compression algorithm $\mathcal{CF}$ parameterised by $\zeta$.

An approach to cooperative classification partly resembling the one detailed here, though without regard for modulation recognition, was pursued by Zhu et al. in [ZAF+19]. The objective of the proposed method there is the minimisation of sensor activation frequency through gating functions in a low-rate Internet-of-Things (IoT) network, rather than the reduction of transferred information. This focus results from the fact that the main concern in [ZAF+19] is energy reduction, and it is mentioned that, in low-power contexts, such as a typical IoT set-

ting, transmission energy consumption depends more on activation than on output amplitude or data dimensionality. The method employed for optimisation is Hessian-based and bears more resemblance to partly-analytical FB approaches for AMC and employs select elements of ML (such as non-linear activations and stochastic gradient descent) rather than relying e.g. on a neural architecture.

## 5.2 Experimental setup

### 5.2.1 Simulation parameters and dataset creation

For purposes of enhanced control over data parameters, the datasets for modulation recognition were created with the aid of specialised algorithms in MATLAB R2019a (an example of their usage is given in [MATb]). As per Figure 5.1 simulation setup consists of one transmitter and $R = 4$ receivers located at the vertices of a square with a side of length $L = 200m$, and the minimum transmitter-sensor distance is selected at $d_{min} = 5m$.

Dataset creation follows the steps described below, for each individual frame:

1. Sample data are drawn from an initial distribution (random integers are used for digital modulation, a .wav audio file including speech and music for analogue) so as to create N sequential symbols.

2. These symbols are modulated according to the selected method.

3. The resulting waveform is transmitted through a different realistic multipath fading SISO channel model for each of the four receivers (the process includes conversion to and from bandpass).

4. Finally, the received signal is degraded by the addition of noise. Eschewing the traditional method where additive white Gaussian noise (AWGN) is added at discrete values, in this case the noise level depends on the operating bandwidth, and remains constant

throughout. What accounts for different degrees of signal degradation is the variance in the received signal power at each of the end nodes (which, among others, depends on the location of the transmitter).

The above procedure is repeated for all classes ($C = 11$), and in a way that each class comprises the same given number of frames.

The potential classes of a signal represent one of three analogue (BFM, AM-SSB, AM-DSB) or eight digital (BPSK, QPSK, 8PSK, 16QAM, 64QAM, PAM4, CPFSK, GFSK) modulation schemes available. For each class, during the simulation, $M = 9200$ frames are created, which are subsequently broken down into training and testing sets by a ratio of 4 to 1 (i.e. 80% of the samples are retained for training, and another 20% for testing).

The baseband sampling frequency is set to $f_s = 200kHz$. This results in a rate of 8 samples per symbol, and each transmitted frame consists of 1024 samples, with every sample representing the in-phase and the quadrature component of the signal, respectively. The RF carrier frequency for transmission is set at $f_{ca} = 2.45GHz$ for analogue and digital modulations alike.

Before being passed to the fading channel, each frame is normalised to attain a transmission power of $P_T = 3dBm$, a value which is well within the capabilities of 4G LTE user equipment (UE) (the maximum value allowed for is $P_{T,max} = 23dBm$) [JCT+17].

To the best of our knowledge, the most appropriate models available for the simulation of realistic channel fading effects in geographically distributed modes to which we had access at the time of the creation of the datasets to be used in the relevant experiments were the 802.11 models of the MATLAB 2019a Wireless Local Area Network (WLAN) toolbox [MATa]. The environment is considered dynamic, so at every iteration of the algorithm (*i.e.*, for each individual frame) new geographical conditions and channel models are created.

Further important parameters of the channels are laid out below:

- Delay profile 1: Profile E, characterised by:

  - Breakpoint distance $r_{break} = 20m$
  - RMS delay spread $t_{RMS} = 100ns$
  - Maximum delay $t_{max} = 730ns$
  - Rician K-factor $K_F = 6dB$
  - Number of taps $N_{tap} = 18$
  - Number of clusters $N_{cluster} = 4$ (representing the independently modelled propagation paths)

- Channel bandwidth $BW = 20MHz$ (the smallest bandwidth available for this channel model)
- Scatterer speed $\upsilon_{sct} = 10\frac{km}{h}$
- Large-scale fading both through pathloss and shadowing effects
- Doppler effects from fluorescent lighting active

With regard to the noise power levels (eq. (5.2)), the ambient temperature is selected to be $T_0 = 290°K$, and $BW = 20MHz$.

### 5.2.2 CNN architecture

In the experiments which concern our experimentation with geographically distributed AMC, the fully convolutional neural network (as described e.g. in Figure 3.9 and used from Section 3.5 onwards) is still chosen as optimal over various contender architectures, including LSTMs. In the geographically-distributed context described in the current chapter, FCNNs were still proven to provide better and faster convergence, and seemed to suffer from minimal overfitting, thus rendering additional regularisation processes largely redundant.

At the same time, it bears mentioning that, as in previous experiments, the dimensionality of the network was altered according to the needs of each individual experiment, and various modifications also took place as explained in the following sections.

## 5.3 Results and discussion

Table 5.1: Results of experiments for modulation detection (Delay profile E)

| Setting identifier | Classification accuracy | Classification time ($\mu s$ / frame) | Data transfer $\rho$ (bits/sensor) | Effective compression rate |
|---|---|---|---|---|
| BL | 74.7%-79% | 45 | N/A | N/A |
| FC1 | 95.5% | 593 | 16384 | 1 |
| FC2 | 77.2% | 181 | 16384 | 1 |
| FC3 | 87.8% | 424 | 16384 | 1 |
| QFC1 | 86.5% | 478 | 512 | 32 |
| QFC2 | 93.3% | 1018 | 1024 | 16 |
| QFC3 | 95% | 1000 | 2048 | 8 |
| QFC4 | 95% | 522 | 4096 | 4 |
| LRC1 | 89.1% | 0.07 | 4 | 4096 |
| LRC2 | 89.3% | 5 | 4 | 4096 |
| LRC3 | 90.93% | 3 | 36 | 455.11 |
| LRC4 | 91% | 9 | 1024 | 16 |
| LRC4a | 79.25% | 4 | 32 | 512 |
| LRC4b | 82.38% | 10 | 64 | 256 |
| LRC4c | 89.86% | 10 | 128 | 128 |
| LRC4d | 91% | 10 | 256 | 64 |
| LRC5 | 91% | 3 | 352 | 46.54 |
| ALRC1 | 92.8% | 0.09 | 4 | 4096 |
| ALRC2 | 93.2% | 2 | 4 | 4096 |
| ETE1 | 83.5% | 163 | 256 | 64 |
| ETE2 | 87.78% | 166 | 1024 | 16 |
| QETE1 | 86.18% | 164 | (32) | (512) |
| QETE2 | 71.47% | 330 | (32) | 512 |
| QETE3 | 9% | 164 | (32) | (512) |
| QETE4 | 46.45% | 173 | 32 | 512 |
| EEPT1 | 89.7% | 135 | 256 | 64 |
| EEPT2 | 97% | 246 | 1024 | 16 |
| ETEC | 87.2% | 170 | 179.9 | 91.07 |
| EEPTC | 96.4% | 167 | 196.37 | 83.43 |

Here we proceed to present the results of our experiments, and endeavour to comment upon noteworthy observations arising therefrom. A complete summary of the results is presented in Table 5.1, where all quantities relevant to our analysis are laid out for each different case which is tested. These quantities are, in detail:

- Classification accuracy obtained from the test set.
- Time required for the classification of a single frame. (NB Attempting to train the

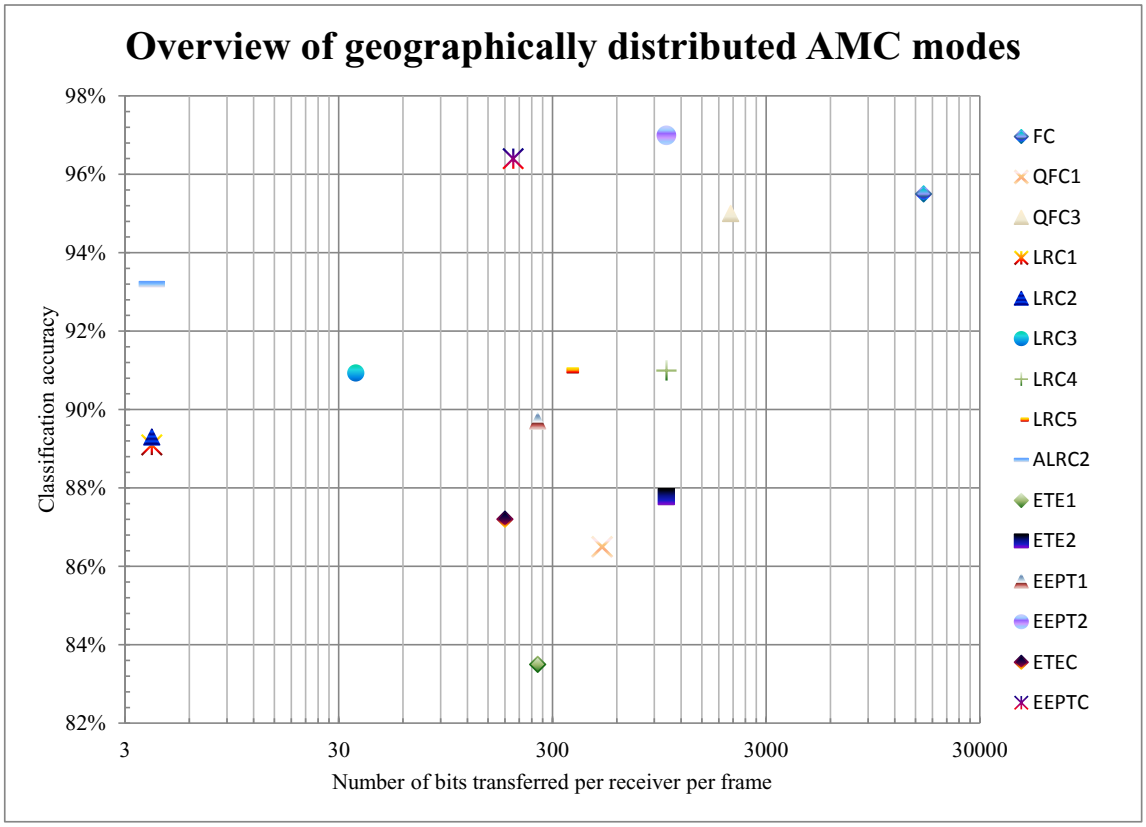



Figure 5.2: Overview of results

classifier multiple times from scratch on the same problem will not result in noteworthy deviation in terms of classification times).

- Number of filters in the last layer of the FCNN. (Wherever this classifier structure is used, this is the most commonly altered hyperparameter.)
- Volume of data required for transfer from each receiver to the central node.
- Effective compression rate. (This is a quantity defined in reference to experiment FC1, as it shall be explained further on, and expresses the reduction of transferred data compared to FC1, which is considered an ideal case.)

The presentation of the different methods shall follow the list laid out in Section 5.1.1. This presentation shall be laid out largely in a logical order of increasing complexity: after starting

from baselines and theoretical best-case results, first we present the simpler data reduction techniques (the first being quantisation), and consequently more complicated ones. In conjunction with Table 5.1, different trade-offs setting the methods apart are described, and we demonstrate how much the overall AMC scheme gains in terms of detection accuracy, decision time, and data transfer load in return for increased complexity. The identifiers for different settings are explained in the following subsections.

A visualisation of the overview of results can be found in Fig. 5.2.

### 5.3.1 Centralised training

#### Baseline (BL)

For reasons of general comparison, a theoretical setting is examined in which each receiving node operates completely independently on its local level, and is fitted with the CNN needed for classification based only on its own data. In other words, the problem here is treated as a simple point-to-point setup for reasons of comparison. This is intended to serve as a baseline against which all other experimental results will be measured.

Let it be noted that both in this setting and in others, unless otherwise stated, the signals presented to the input of the classifier are of length $N = 256$. The format is 32-bit floating point.

The performance which is achieved in this simple case demonstrates how the data used for our experiments can pose a significant challenge to a neural classifier, as the accuracies fall well below the state-of-the-art performance, and is marginally more challenging than e.g. the commonly used RadioML datasets in similar contexts [HGKL20][TTCG19]. This observation also justifies the proposition that the environmental factors modelled during the creation of these data play a crucial role in detection accuracy, as many realistic imperfections are manifest in the received signals.

**Fully cooperative AMC (FC)**

The setup which is at first considered a theoretically best-case scenario is FC1, in which the fusion centre has access to all the recorded data at the same time, and receives input from all four receivers simultaneously. As such, the backhaul links are considered to have infinite capacity. The approach followed for this setting can be considered similar to the one laid out in Section 4.1.1, where signals from ten different antennas of a SIMO system are processed by the same CNN for classification.

The datasets are concatenated in such a way that each input sample assumes a dimensionality of $8 \times 256$, and the widths of the convolutional kernels of the neural network are expanded in order to accommodate the new input structure. For the evaluation of an incoming signal, the classifying node receives a 256-long I/Q waveform from each of the four receivers.

The advantages which a cooperative approach to signal identification can offer are made evident here, as the combination of information from all receivers, which corresponds to different versions of the same signal, offers a detection confidence much higher than any of the theoretical individual classifiers could achieve. A potential explanation for this phenomenon is given if one should conceive this approach as similar to the practice of multiple-view image datasets, where, as the name implies, each element consists of view of the same object from multiple angles. As such a practice offers a more comprehensive "description" of an object for an image classifier, we can safely theorise that the cooperative AMC approach based on multiple versions of a signal, received by geographically distanced nodes, functions in a similar manner.

At the same time, the trade-off which has been mentioned also makes its appearance: similarly to the observations made in 4.1.1, the time required for classification has increased significantly, and we also see that a transfer of data of which the volume is far from negligible is necessitated for the regular function of the classifier. As such, the need for compression and other forms of dimensionality reduction is made evident.

Settings FC2 and FC3 concern a modified version of this experiment, with data drawn from a reduced number of receivers (2 and 1, respectively) but with longer input sequences in inverse

proportion ($N = 512$ and $N = 1024$, respectively). The results demonstrate that performance improvement depends much more heavily on a cooperative approach than on a simple increase of the dimensionality.

**Cooperative classification from quantised data (QFC)**

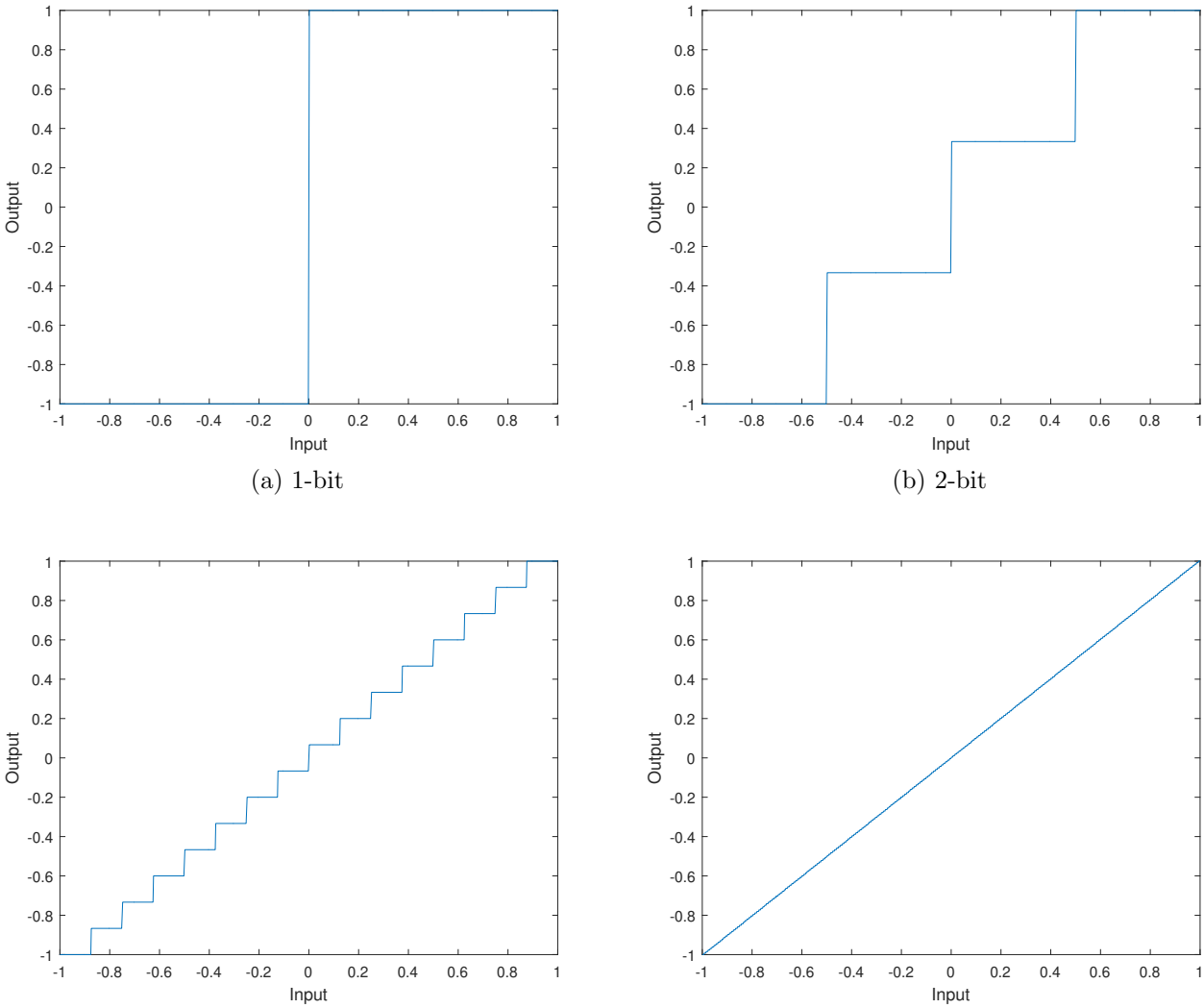
A rather aggressive method of data volume reduction consists in the quantisation of datasets, by which we refer to the practice of reducing the resolution of the individual values to lower levels than their original ones. In the current case, the whole dataset is quantised from 32 to 1 bit per time sample per channel. The procedure followed for this purpose is the same as described in Section 3.6.1.

The cases presented with numbers QFC1 to QFC4 concern a quantisation of the input signals for quantisation to $b_q = 1, 2, 4, 8$ bits.

What we observe in the related results for $b_q = 1$ is that such a reduction of resolution leads to a serious degradation of performance compared to the ideal scenario, yet at the same time, the cooperative aspect of the method remains a rectifying factor. One may note, by way of example, that even the best classifier trained on a single receiver at full resolution does not outperform the quantised cooperative algorithm.

For $b_q = 2$, however, there is already a noticeable improvement, and for $b_q = 4$ we notice that the performance is practically identical to that of the ideal case FC1. The reason for this can be demonstrated in Figure 5.3, where a vector covering the range of values spanning from $-1$ to $-1$ in increments of 0.002 is quantised with the aforementioned technique to 1-bit, 2-bit, 4-bit and 8-bit resolution.

As it becomes clear, even an increase to 4 bits makes a much higher level of resolution available, while an 8-bit quantiser is almost identical to the identity function for all intents and purposes. This confirms that the CNN-based modulation detection structure succeeds at detecting crucial patterns in the input even at lower resolution; as such, the depth of precision offered by the 32-bit floating point format is not necessary for reliable results. Given this observation, repeating




(a) 1-bit (b) 2-bit

(c) 4-bit (d) 8-bit

Figure 5.3: Quantisation of a linear space between -1 and 1

the experiment for $b_q = 16$ was deemed redundant.

A problem which is still present in the quantisation cases discussed here is that of execution time, which is somewhat lower than in FC1, but still relatively high. This presumably persists because the CNN still accepts the data as 32-bit floating point number, even if fewer than $2^{32}$ possible values can be assumed.

**Cooperation based on intermediate representations (LRC)**

For the experiments denoted as LRC1 to LRC5, the backbone is still the same baseline localised algorithm the results of which are reported in BL, i.e. with each edge node being equipped with an individual, trainable classifier structure and operating upon its own received data alone. However, the code is altered in such a way that, during the execution, the following quantities resulting from each individual input signal are retained:

- The final decision/vote with regard to the identity of a received signal (requires 4 bits due to the presence of 11 classes)
- The output of the global average pooling (GAP) operation (32-long vector of floats)
- The confidence vector extracted from the end of the neural network (11-long vector of floats)

Consequently, it is assumed that a cooperative system is planned to be trained at the central node based on these compressed quantities, which can be considered latent representations of the signal.

In experiment LRC1, classification is based on a very simple brute-force voting mechanism, where a decision is made by simply extracting the mode of the four decisions originating from the local classifiers. We see in this case that even such a simple form of inter-node cooperation can achieve great improvements, whereby two or three receivers can effectively compensate for the faults of others. In terms of operation time and data transfer requirements, this proves itself by far the most competitive method.

In LRC2, the local votes are used as the input to a single-layer dense neural network at the central node, and a minimal improvement is noted. The structure of the network in question is very simple, consisting only of a single layer (barring the final with output 11), fully-connected with 1024 output neurons, activated by a leaky ReLU function.

However, the improvement becomes more pronounced in experiment LRC3, where the the transmitter's position is implemented as additional information for the first time. More specifically, if the transmitter occupies the position $\mathbf{p_t} = [\chi, \psi]$, then with:

$$d_j = \|\mathbf{p_t} - \mathbf{p_j}\|, \quad j = 1, \ldots 4$$

As its distances from each of the four receivers, then with $d_{max}$ and $d_{min}$ as the maximum and minimum values, respectively, which $d_j$ can assume throughout the entire dataset, we define a set of weights as follows:

$$\omega_j = \frac{d_j - d_{max}}{d_{min} - d_{max}} \tag{5.8}$$

Under the presupposition that the receiver which is closer to the transmitter is likely to have the least attenuated and degraded version of the signal, and thus is more likely to result in an accurate prediction.

The vector of distance-based weights $\omega = [\omega_1, \omega_2, \omega_3, \omega_4]$ is concatenated with the vector of votes and subsequently fed to a neural network with a single convolutional layer for classification. The incorporation of distances as additional information has led to greater improvements in comparison to LRC2.

When it comes to exploiting the GAP layer outputs, under consideration of the notion that it can be considered an intermediate latent version of an input, it was theorised that this compressed version alone can be used to train a local mini-neural-network consisting of just the softmax layer which exists at the end of the usual FCNN, as in the latter case, it is the only element succeeding the GAP function. For this purpose, and because dense layers only accept one-dimensional inputs, the four latent representations of the signal are concatenated and

averaged. A high performance metric is reached based upon the above assumption, presented in row LRC4; yet still it falls short of the ideal performance. This can be possibly ascribed to the fact that in this case, the training is broken down in many different parts (four CNNs at the local receivers, another one at the central node) which do not communicate with each other at any phase, and thus the backpropagation of information from which the fully-cooperative case benefits is not present.

A possibility for leveraging the local GAP outputs while reducing the required channel capacities between the local and central nodes was sought in the quantisation of the GAP vectors instead of the signals (inspired by the experiments where the signal was quantised) to 1-bit, 2-bit, 4-bit, and 8-bit resolution, according to the procedure described inSection 5.3.1. The results for these settings are presented in rows LRC4a to LRC4d of the table, and the conclusions regarding quantisation reached earlier on are justified again.

An alternative latent representation to be considered is the 11-long output of the local CNN, which can be considered a confidence vector regarding the probabilities that a given input belongs to different classes, since the final decision depends on the distribution of this vector, and is mainly based upon locating its maximum element. These are classified in the central node by a single-layer convolutional network and result in the performance reported in the row with identifier LRC5, which offers competitive performance compared to the GAP approach, especially if the improved compression rate is taken into consideration.

#### Amplified local training for intermediate representations (ALRC)

The settings labelled ALRC1 and ALRC constitute repetitions of LRC1 and LRC2, respectively, but in this case, instead of training four classifiers on local data, we train only one on the conglomerate of the local datasets and subsequently deploy it on a local level for evaluation. The same latent versions of signals are retained, namely votes, GAP layer outputs, and final confidence vectors. As is made clear, the gains observed as a result of this effective augmentation are notable, albeit not dramatic, doubtlessly owing to the greater centralisation of the training phase and the benefit which is offered by the presence of a larger, conglomerate dataset. This

served, as we shall see, as an additional incentive for one of the methods which were employed later on.

### 5.3.2 End-to-end training

The experiments which have been detailed so far, i.e. from BL to ALRC, are conjoined by the common underlying factor that all of them rely on centralised or localised training, or a combination of the two, with the interaction between the local and central nodes restricted to one-way data transfer from the former to the latter. In direct contrast to these stands the end-to-end approach.

When referring to settings as trained "end-to-end" in the current context, what is meant is that the setup is considered as one great neural network beginning at each of the individual receiving nodes and ending at the central node. As such, it accepts four parallel inputs (corresponding to the locally received signals) and has several convolutional layers running parallel to each other, the outputs of which have to be combined at some point before reaching the final layer. In practical terms, this means that off-line, after training is completed, each of the local nodes will host part of the trained network, and the rest will reside in the central classifying device.

Experimentation showed that the optimal way conceived so far for the streams of information from the parallel inputs to be combined is with a concatenation of the outputs of the GAP layers (e.g. if each has length 32, then we concatenate into a vector of length 128) at the end of each convolutional branch. A generic visualisation of the architectures used for end-to-end settings is presented in Fig. 5.4.

#### End-to-end training of parallel CNNs branches (ETE)

Settings ETE1 and ETE2 as recorded here cover two different executions of the end-to-end algorithm. The parameter which is altered between them is the number of kernels employed by the last convolutional layer, which accordingly influences the output of the GAP operation and the amount of data transferred to the central node.

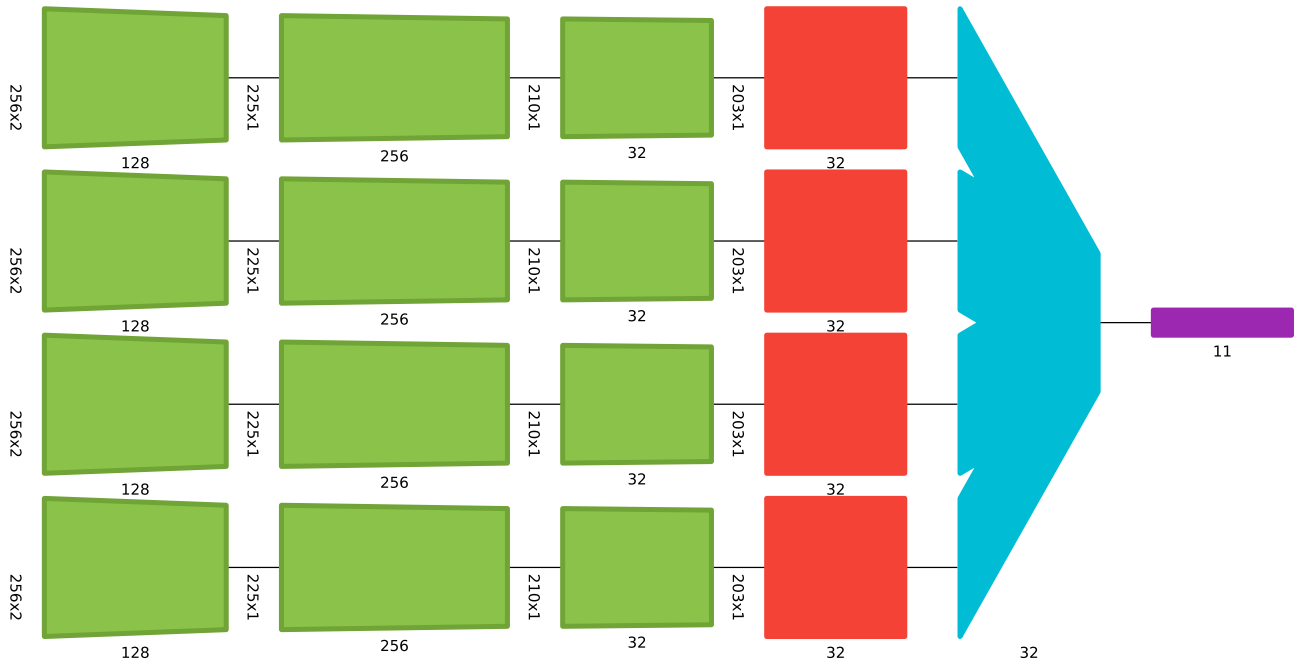



Figure 5.4: Generalised example of an FCNN used in end-to-end settings. The colour-coded families of layers are Convolutional (green), Global Average Pooling (red), Averaging (light blue) and Dense/Softmax (purple)

One of the most poignant differences which is present when the system is treated as an end-to-end network is that there is a loss in classifier accuracy, though not dramatically significant, compared to earlier, more centralised experiments. This can probably be ascribed to the fact that certain earlier approaches such as experiment FC1 benefit from having all of the inputs present at the start of the neural network and filtered by the same layer, thus facilitating the extraction of correlations between the different receivers via the function of the convolutional kernels, as well as information propagation between nearby branches during training. By contrast, the end-to-end architecture, where there are parallel branches running, and not interacting directly with each other, might render the propagation of information more difficult in certain instances.

### Approximation of quantised data in end-to-end structures (QETE)

Since the training of the neural classifier relies on the computation of partial gradients throughout the network, the implementation of a quantised output from the local branches to be transferred to the fusion centre is deemed impossible, due to the fact that the step function (on

which the quantisation process has relied so far) is not differentiable at zero.

A readily-available differentiable approximation of a quantisation procedure, however, can be given by the sigmoid function, which is presented in Equation (5.9) parameterised by a scaling factor $a$:

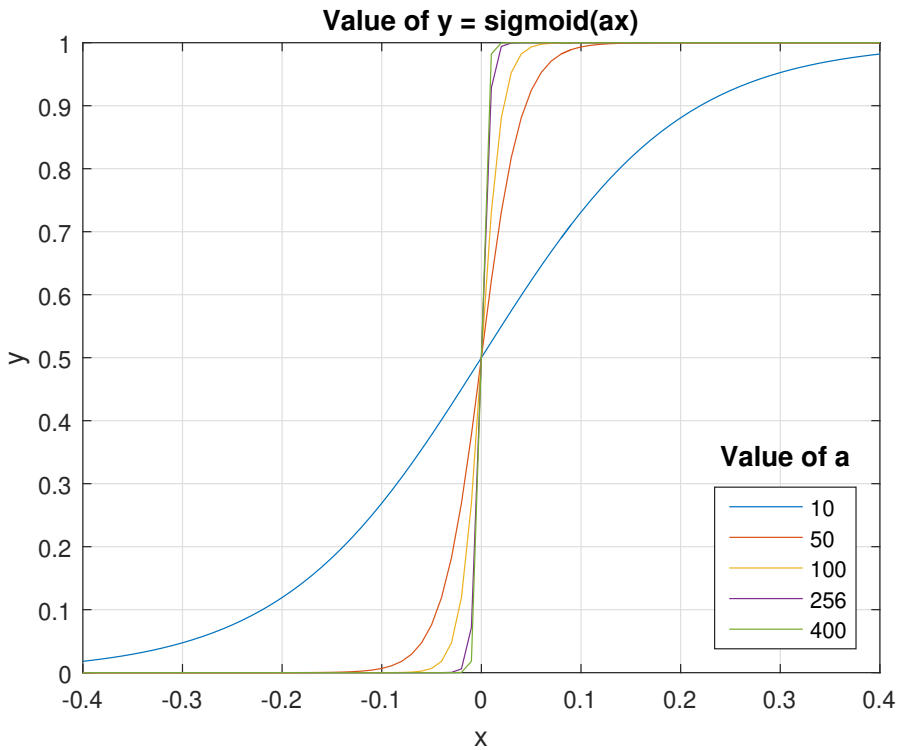



Figure 5.5: Sigmoid functions scaled by different factors

$$S(x; a) = \frac{1}{1 + e^{-ax}} \tag{5.9}$$

A visualisation of the sigmoid function for different values of $\alpha$ is given in Figure 5.5, and helps demonstrate the fact that, at least during the training phase, a sigmoid function can act as an approximator of a 2-bit quantisation process for adequately large values of the input variable $x$. By way of example, for $a = 50$ the only values whose output will significantly vary between 0 and 1 are $\|x\| \leq 0.1$; for $a = 100$ the corresponding range is halved approximately to $\|x\| \leq 0.05$, whereas for $a = 256$ and $a = 400$ the behaviour of the sigmoid is practically identical to that of a step function activated at $x = 0$.

With the above in mind, the proposed modification consisted of applying sigmoid activations after the GAP outputs in the following manner:

$$GAP_{out}[n] = 2S(GAP_{in}[n] - 1), \quad n = 1, \ldots, L_{GAP} \tag{5.10}$$

These activation layers are expected to map most of the values of their input to a binary output of either 1 or $-1$, as was the case in all previous implementations of 2-bit quantisation. Consequently, during deployment, these structures can be replaced by ordinary step functions, since no computation of gradients is necessary any longer, and result in a scheme where each element of the GAP outputs is reduced to 1-bit resolution and transferred thus to the fusion centre.

The results for this modification of the end-to-end structure are presented in the rows with numbers QETE1, QETE2, and QETE3. The sole difference between them concerns the value of $a$ selected for the sigmoids; 6a reports the results for $a = 100$, 6b for $a = 256$, and 6c for $a = 400$. As is made obvious, although a relatively high accuracy metric is retained initially, a degradation becomes evident as the value of $a$ increases, to the point where for large enough values the efficiency of the end-to-end CNN is almost identical to a random classifier. This behaviour is most likely attributable to the phenomenon of exploding and vanishing gradients.

Within the traditional context of deep neural network training, exploding and vanishing gradients have been a pressing issue [Han18][PMB13]. Simply expressed, given a general form of gradient descent for paremeter optimisation (Equation (3.7)), an exploding or vanishing problem is present during training when the norm of one or more derivatives assumes values which are either very large or very close to zero, respectively; this, in turn, causes the weights to suffer from excessive variation or to not be updated at all, and results in instability in the network. The exploding and vanishing gradient problem is part of the reason why, among others, the sigmoid and hyperbolic tangent functions, which used to be standard fare as neural activations, have been, to a large degree, replaced by ReLU and similar functions, since the former two can

exhibit large and very small gradients, especially if cascaded throughout multiple layers.

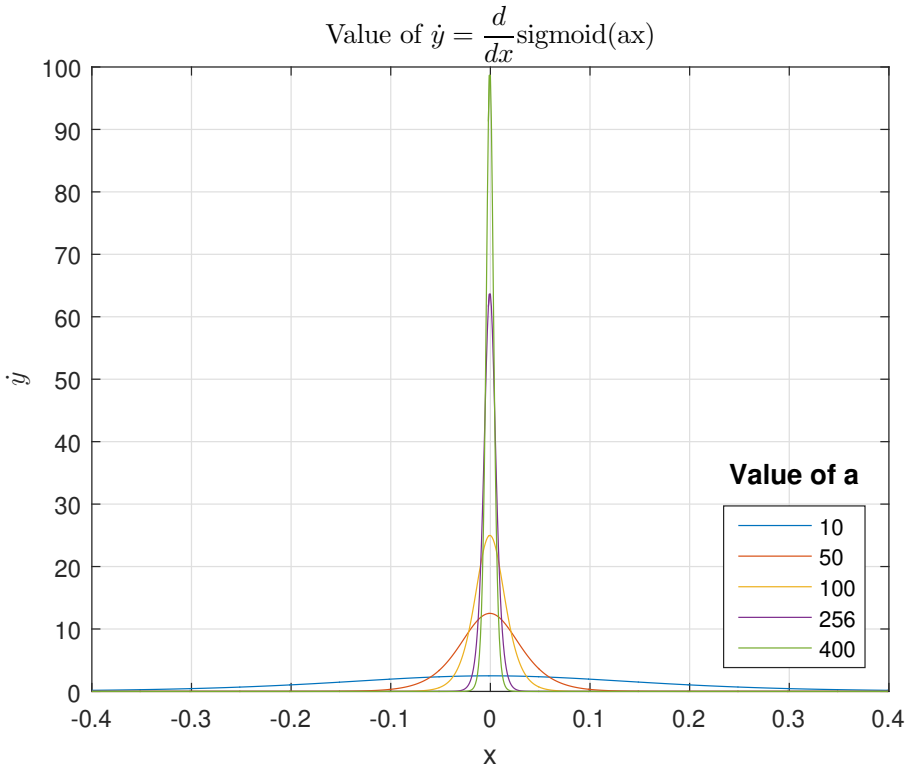



Figure 5.6: Derivatives of sigmoid functions

Should one assess the derivative of the paremetric sigmoid function:

$$S'(x;a) = \frac{\partial}{\partial x} S(x;a) = \frac{ae^{-ax}}{(1+e^{-ax})^2} \tag{5.11}$$

It is easy to understand why it introduces strong possibilities of gradient explosions: a part-linear dependence on the scaling factor $a$ is present, and particularly for $x = 0$ the value of the function becomes equal to $S'(0;a) = \frac{a}{4}$ (the influence of the parameter $a$ upon the derivative is presented in Figure 5.6).

Additionally, another quantity worth assessing, as a way of examining how the effective range of the function is impacted by $a$, is the $x$ for which the derivative assumes an (arbitrary) small value, for instance: $x_0$ such that $S'(x_0;a) = 0.05$. For values of $a \geq 10$, this is approximately equal to:

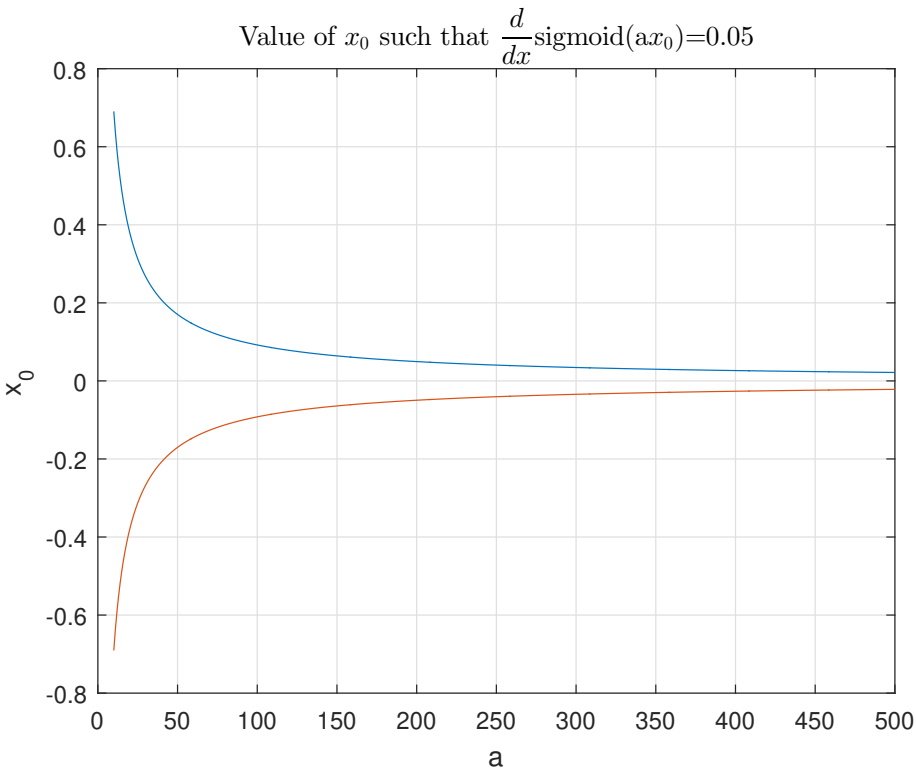



Figure 5.7: Roots of $S'(x;a) - 0.05$ for different $a$

$$x_0 \approx \frac{ln(50a \pm \sqrt{a(25a-1)} - 1)}{a}$$

This quantity is plotted in Figure 5.7, and thereby it is confirmed that, at least for $a \geq 10$, $\|x_0\|$ strictly descending as a function of $a$, and thus for larger $a$ there is a greater range of input values the output of which is close to zero.

As a result of the above, it is expected that increasing $a$ shall lead to a steeper slope near $x = 0$ for the quantisation-approximating sigmoid activations, thus resulting in an increasing destabilisation of the network due to the backpropagation of increasingly large gradients; at the same time, a vanishing gradient also becomes more likely, as it is theorised that the range of input values for which the derivative is adequately larger than zero shrinks. It is evident that these expectations are justified by relevant results.

QETE4 is a setting relying on pre-trained weights, and as such will be detailed in the following

segment of the report.

### End-to-end with pre-trained local branches (EEPT)

The idea for this variant of the experiment was motivated by an earlier observation regarding the case of latent-representation-based detection (see 5.3.1), namely that the local classifiers exhibit better performance if they are initially centrally trained on all available data and then used for the extraction of the intermediate features. Additionally, it was inspired by the general concept of transfer learning, whereby a ML structure is trained in a particular context and then used as a pre-trained classifier for different cases, typically involving different data distributions [SZL15][LFY19].

For cases EEPT1 and EEPT2, a complete FCNN (exactly like the one used in setting BL) is trained with the same configuration and parameters as in 5.3.1, then the weights of the convolutional and batch normalisation layers are frozen and transferred to the end-to-end structure represented in Fig. 5.4, covering the part reaching up to the red-coded layers. Consequently, the only part which is yet to be trained in the end-to-end phase is the final dense layer which reshapes the averaged GAP outputs into the confidence vector.

The performance of this approach has been so far the most successful evidence of the potential borne by cooperative methods for signal identification. Drawing upon the strength of all the available data, it produces the best results obtained so far, even better than the theoretically ideal experiment FC1 in the case of EEPT2, at only a fraction of the per-frame computational cost and data load upon the communication channels.

In addition to the three aforementioned schemes, QETE4 concerns an end-to-end setting where the sigmoids described above have been replaced by the sign operation available in TensorFlow. Though, as already mentioned, this is not a differentiable function, nevertheless the code seems to be able to train the end-to-end network with these operations in place (it is not certain whether this is done e.g. by treating them as equal to the identity function during training). The results of this method are, unfortunately, not worthwhile.

**Implementation of compression (ETEC)**

In their 2017 work [BLS17], Ballé et al. devised a novel technique for image compression based on an end-to-end trained autoencoder structure. The algorithm proposed there consists of three discrete stages: an analysis transform, a quantiser function, and finally a synthesis transform, all of which were implemented as combinations of convolution (or de-convolution) layers and non-linear activation functions.

The main foundation of this method, as well as image compression research in general, lies in the admission that compression usually does not follow from a quantisation of the pixel values of images directly, but rather of a latent representation (let $\nu$) of the same in an alternative vector space, via the imposition of an intermediate transformation. Additionally, $\nu$ is typically quantised (as $\hat{\nu}$) to a set of discrete values, and as a result, can be subject to lossy or lossless compression and thus have its redundancies minimised. [RL81]

The process of extracting latent versions of inputs is not dissimilar from our own approaches, as explained in several of the previously detailed experiments; the one-dimensional output of the GAP operation can be considered to be one such quantity, and it is even derived in a similar manner to the Ballé method, namely through dimension reduction via a CNN structure. Furthermore, the transmitted signals (at least in their versions which are not degraded by noise and environmental factors) can be considered to follow certain non-random, predetermined distributions depending on the different modulation types, this property can be expected to be manifest to some degree in the latent representations as well; consequently, their entropy is non-maximum, and thus they can benefit from arithmetic coding after quantisation.

With the above considerations in mind, it was theorised that our end-to-end approach could benefit from a reformulation rendering it similar to the Ballé architecture. The analysis transform is already available in the form of the local series of convolutional and GAP layers; in contrast to image compression problems, however, a synthesis transform is not needed, as we are concerned only with correctly identifying a signal and not with reconstructing it (it is for the same reason that lossy nature of the compression does not constitute a pressing issue). As

such, the only modification applied to the pre-existing architecture (as per 5.3.2) is the addition of an entropy coding layer after each of the parallel GAP operations; this layer, which is based on the estimation of entropy via a flexible non-parametric density model [BMS+18], imposes an information bottleneck upon the input data, extracts a bit-stream representation of the same, and then decodes.

The results seem to justify the aforementioned assumptions, as both a robust classification accuracy and a relatively large compression ratio are achieved.

#### Combination of compression and pre-trained branches (EEPTC)

Attempting a "best of both worlds" approach, according to what we have deduced to be the most appropriate approaches to the problem so far, we decided to merge the pre-trained local networks approach with the entropy-coding-based compression methods. As was the case in 5.3.2, a single-branch CNN is trained on data from all receivers, only this time it also includes an entropy bottleneck function following the GAP function. The weights which are frozen during transferring include (in addition to convolutional and batch normalisation layers) the entropy bottleneck, which by itself constitutes a complex element in its TensorFlow implementation, as it is modelled by 14 weight matrices (by contrast, a convolutional layer has only 2 matrices).

The combination of the two techniques appears once again to be justified in practice: not only does it reach one of the highest accuracy rates of all assessed experiments (better again than FC1), but a strong compression ratio is still prominently present.

### 5.3.3 Further discussion of results

#### Explaining CNN behaviour

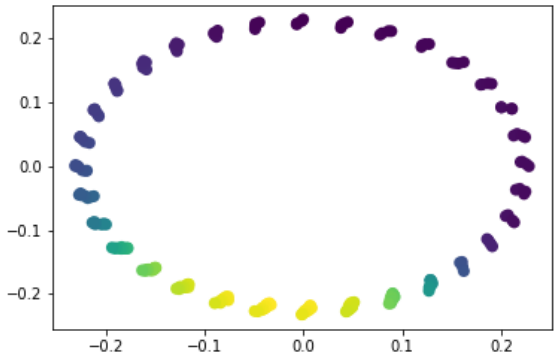
Although it has not been possible so far to derive an exact function of the classifier for the modulation classification problems which have concerned our research so far, and thus a mathematically tractable explanation for behaviours such as the ones mentioned above is rather

infeasible, still a rudimentary general explanation of the performance of the algorithms falling under the category of CNN-based AMC can be given by our theory, partly based on the observations of Peng et al. among others [PJW+19][SBSB], that one of the ways in which the CNN classifier views the modulated time series at its input is as a 2D representation of its constellations, not unlike a kind of image file; as such, the practice of augmenting the AMC dataset with measurements from different receiving nodes can be compared to the practice of image datasets providing multiple views of the same object. This theory would justify the significantly enhanced performance of the AMC structure in these new experiments.

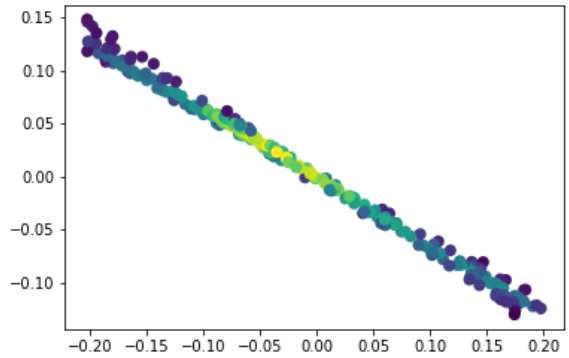
A theory mentioned in previous chapters dealing with AMC with interference, based upon this logic, was that a particular deterioration of performance noted when the superposition factor $\alpha$ was equal to 1 could be ascribed to the same tendency of the neural network to interpret input signals as constellations on a 2D space, and the deterioration was due to the two superposed signals occupying roughly the same expanse in this space. In order to assess this, we present in¬Figure 5.8 the scatter plots for two sample signals from the CPFSK and PAM4 classes, respectively, as well as their addition with varying values of $\alpha$. While these figures are a potent explanation for the overall behaviour of the classifier under interference (e.g., at $\alpha = 10$, which was the greatest value considered, the CPFSK constellation is unrecognisable), no particularly strong insight is provided with regard to the sudden performance drop at $\alpha = 1$. Further research will be required to adequately address the reasons for this effect, which is currently beyond the scope of this work.

An additional assessment which can be carried out, given the results of all the aforementioned experiments, may focus on the behaviour of different classes of signals in conjunction with the utilised algorithms. For this purpose, confusion matrices computed during the execution of four different architectures are presented in Fig. 5.9. The four experiments were selected in such a manner as to showcase a diversity of methods, which is why the theoretical best-case benchmark (FC1) is juxtaposed to a case of brute-force voting (LRC1), a low-performance end-to-end case (ETE1), and the optimised case where compression and pre-training are combined (EEPTC).

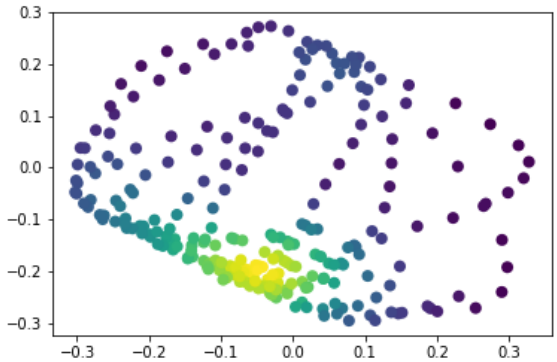
Firstly, it becomes apparent here is that most of the modulation classes exhibit excellent per-



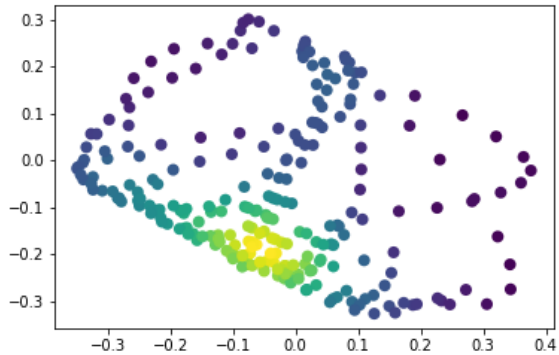

(a) CPFSK signal sample



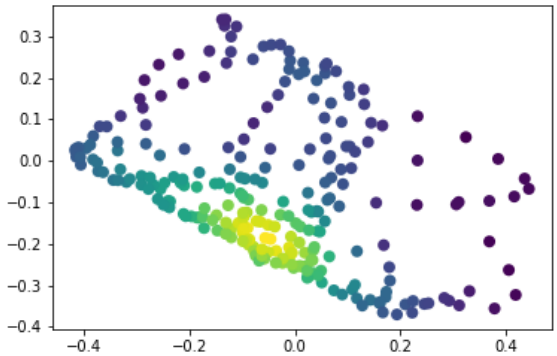

(b) PAM4 signal sample



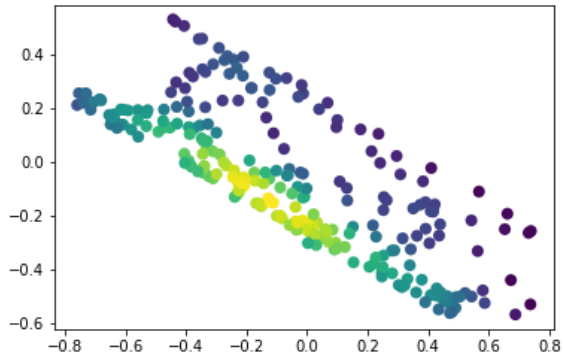

(c) Superposition at $\alpha = \frac{\sqrt{2}}{2}$




(d) Superposition at $\alpha = 1$



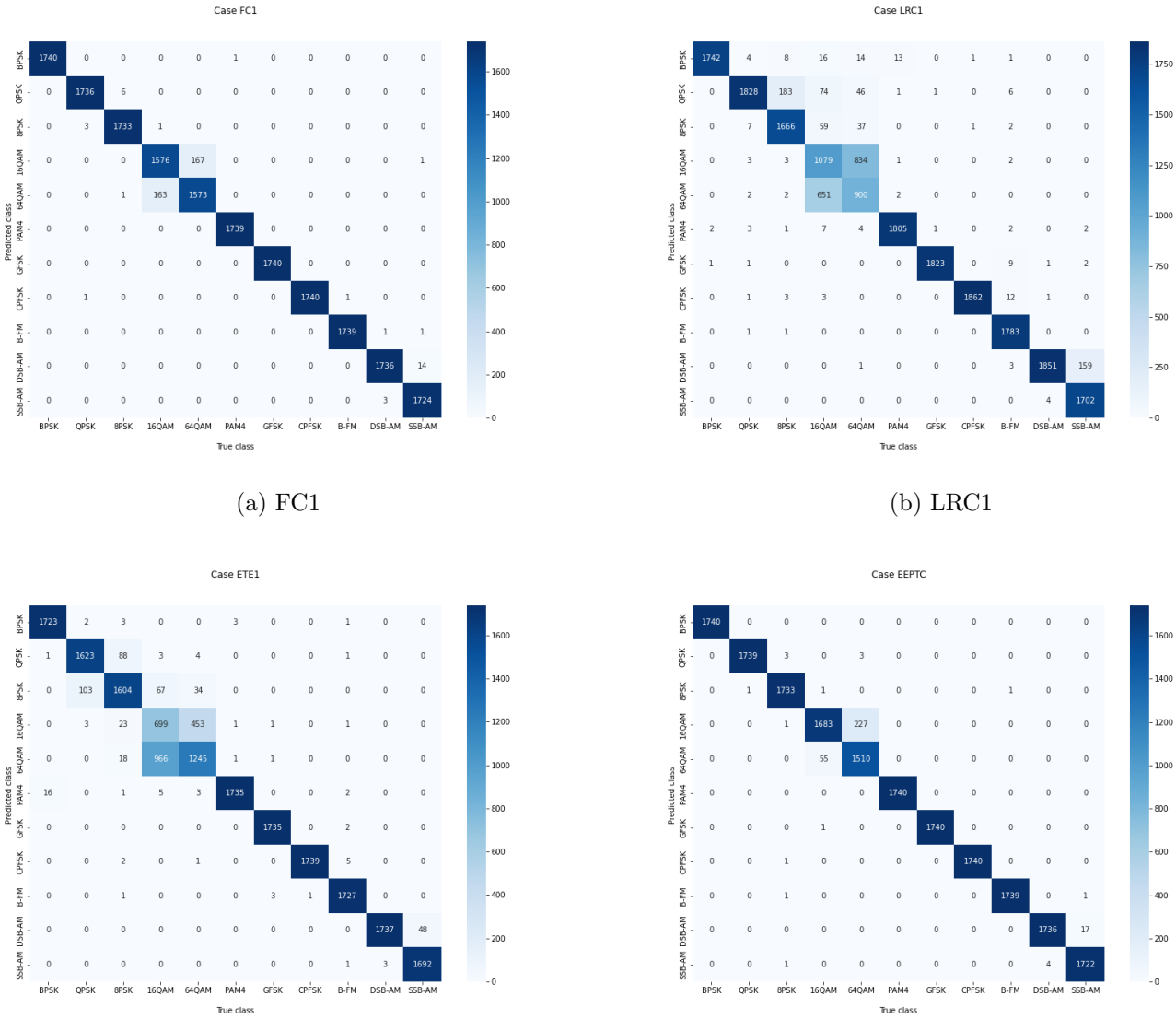

(e) Superposition at $\alpha = \sqrt{2}$




(f) Superposition at $\alpha = \sqrt{10}$

Figure 5.8: Scatter plots of two sample signals and their superposition.

formance throughout the entire dataset, irrespective of the degradation of the signals resulting from environmental influences. Additionally, it is observed that most of the misclassifications occur within the same family of modulations (i.e. typically QAM, and to a lesser degree PSK and AM).




(a) FC1

(b) LRC1

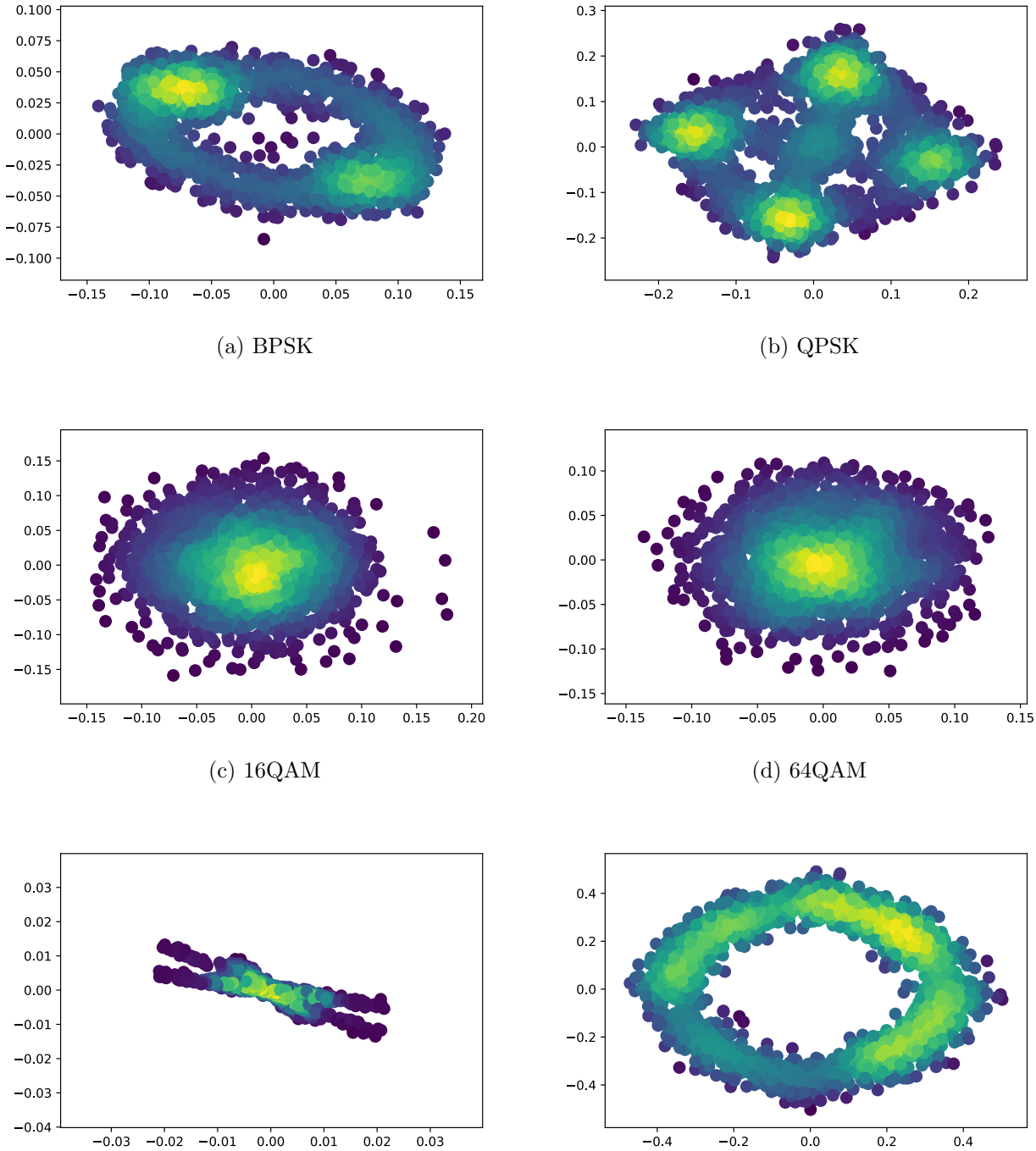
(c) ETE1

(d) EEPTC

Figure 5.9: Sample confusion matrices from different architectures

In a similar vein to previous works [TTCG19], and is still pointed out in most state-of-the-art literature on AMC, there persists the problem (mostly in regions of higher noise power) of a high probability of mutual confusion of the two existing QAM classes, with 64QAM being much less robust. This problem (which was also reported in [PJW+19]) may also be attributable, to a large degree, to the aforementioned tendency of neural networks to view input time series as a scatter plot view of constellations.

We proceed to further demonstrate this using data from an older MATLAB-generated dataset, where a set of signals modulated by the same $C = 11$ modulators are transmitted over a




(a) BPSK (b) QPSK

(c) 16QAM (d) 64QAM

(e) AM-DSB (f) CPFSK

Figure 5.10: Scatter plots of signals from different modulation classes

$1 \times 10$ single-input multiple-output (SIMO) channel with Rician fading, and further influenced by AWGN. Fig. 5.10 presents scatter plots of selected signals from six different modulation classes (only samples with SNR $= 18dB$ were selected, for clarity). It becomes easily noticeable that each scatter diagram exhibits a certain degree of similarity to the original constellation on

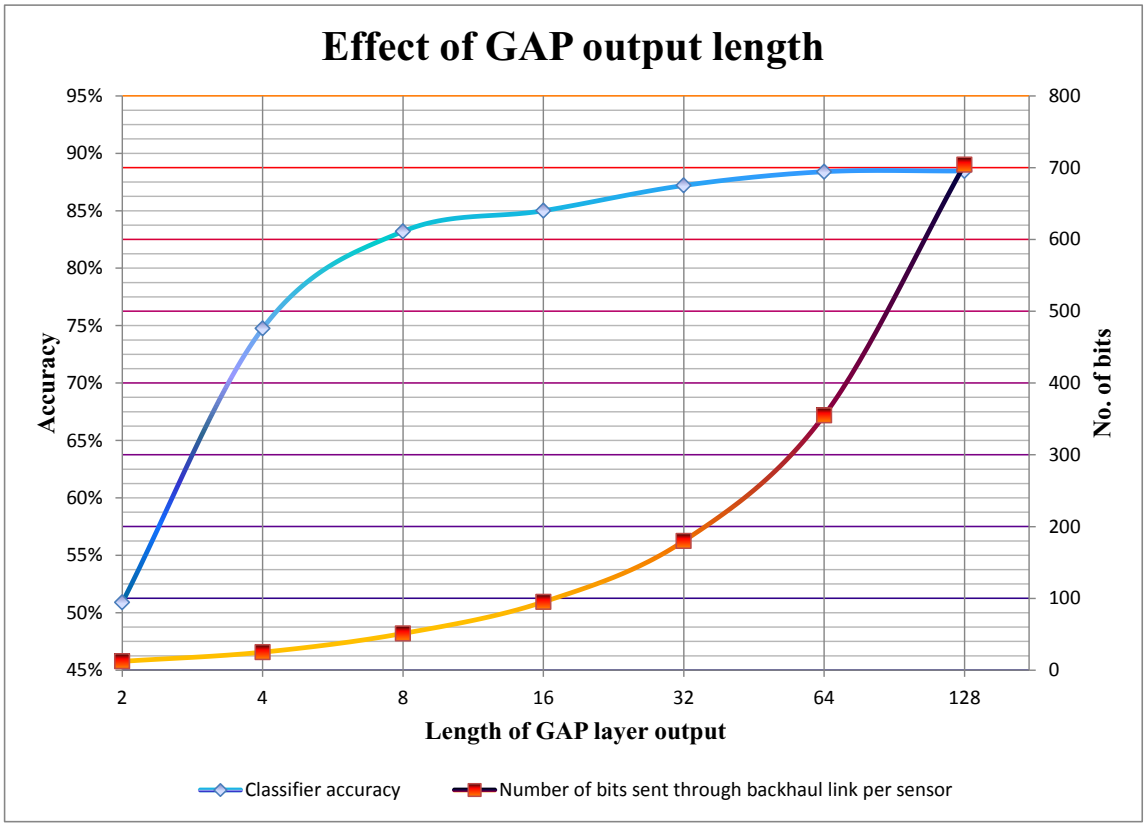



Figure 5.11: Assessment of the effect of the GAP output length upon classification accuracy

the I/Q plane, with the sample density expected to exhibit a number of local maxima largely corresponding to the modulation order; by way of example, BPSK and QPSK plots tend to feature 2 and 4 peaks, respectively.

It is likewise evident that the images resulting from the visualisation of QAM constellations are easily distinguishable from other modulation schemes, but exhibit great similarity to each other regardless of the order of modulation; consequently, this analysis serves to demonstrate that mutual confusion of M-QAM classes is all but expected to be a pervasive problem in AMC. On a data level, this could be attributed to impairments incorporated during signal creation and propagation, which are largely non-linear and multiplicative [SBSB]; the effect of these impairments is already clearly distinguishable in several of the samples which have been selected to be presented in Figure 5.10, as we see that a great number of scatter points

reside in the space between poles of high concentration. In conjunction with the fact that QAM modulations of an order of 16 or greater feature highly dense constellations and lead to heavily packed grids, it stands to reason that the respective classes in an AMC dataset shall be also highly dense and similar one to another, thus creating confusion.

**Data load vs. accuracy trade-off in schemes with compression**

One final remark may be made at this point, inspired by the observation of the different ETE settings of the experiment, pertaining to the relation between the chosen length of the local network branch outputs and the final classification accuracy of the end-to-end scheme. It should be expected that a trade-off should be found to exist involving the volume of information provided over the backhaul links, and the performance of the neural network.

For visualisation purposes, we present this relation for seven different parameter settings in Figure 5.11. As predicted, the length of the GAP output vector affects both the backhaul data load and the overall performance. The volume of the compressed versions of the latent representations of signals follows a largely linear relation to the uncompressed volume, and thus the effective compression rate remains constant. Classification accuracy, on the other hand, demonstrates a dramatic improvement in the lower areas (2-8 layers in the last layer), then the slope becomes less steep, and it can be safely assumed that saturation is reached for areas beyond a GAP output length of 64 (this confirming the general observation that widening a neural network offers tangible results with regard to performance improvement, but eventually reaches a point where an increasingly widening net does not result in any further optimisation).

### 5.3.4 Propagation in different channel conditions

Although the dataset corresponding to the spatial delay profile E was the one most extensively used throughout related experimentation for geographically distributed AMC modes, a second dataset, with the same total number of samples, was created in parallel to the one previously discussed, for reasons of comparison between different geographical conditions and models. In

the case of this second dataset, the environmental conditions match a different profile which is named model C, and is characterised by the following attributes:

- Breakpoint distance $r_{break} = 5m$
- RMS delay spread $t_{RMS} = 30ns$
- Maximum delay $t_{max} = 200ns$
- Rician K-factor $K_F = 0dB$
- Number of taps $N_{tap} = 14$
- Number of clusters $N_{cluster} = 2$

The rest of the parameters concerning the creation of the dataset are exactly the same as the ones described in Section 5.2.1.

**Results**

To the end of assessing the effectiveness of the proposed algorithms in a different setting than the one considered so far, as well as assessing practically manifested differences between the propagation profiles themselves, a representative sample of the previously attempted AMC algorithms were run on this secondary dataset.

The results of this endeavour are presented in Table 5.2. Note that the numbering in the leftmost column follows the same vein as the one in Table 5.1, which means that the same setting corresponds to the same experiment code in both tables.

The degradation of the performance of our AMC algorithms for this different environmental profile becomes apparent in all of the experiments which have been selected for comparison. By way of example, in the cases of the local classifiers acting alone (BL setting) and the infinite-capacity backhaul cooperative scheme (FC1), the comparative accuracy losses range from 10% to 20%, which marks a significant deterioration.

Table 5.2: Results of experiments for modulation detection (Delay profile C)

| Scheme # | Classification accuracy | Classification time ($\mu s$ / frame) | Data transfer (bits/receiver) | Effective compression rate |
|---|---|---|---|---|
| BL | 50.3%-57% | 40-45 | N/A | N/A |
| FC1 | 87.8% | 662 | 16384 | 1 |
| QFC1 | 72.5% | 657 | 512 | 32 |
| LRC1 | 67.3% | 0.07 | 4 | 4096 |
| ETE1 | 81.19% | 164 | 256 | 64 |
| ETE2 | 82.79% | 212-214 | 1024 | 16 |
| ETEC | 80.86% | 90 | 179.9 | 91.07 |
| EEPTC | 89.88% | 90 | 196.37 | 83.43 |

What is most likely the foremost source to which this behaviour can be attributed is the Rician K-factor, which has been reduced by 6 dB from the value which it assumes in profile E. As is known from the modelling of fading channels with a Rician distribution [TAG03], a complex received signal $\mathbf{r}(t)$ resulting from the superposition of LOS and non-LOS RF waveforms can be expressed, without loss of generality, as such:

$$\mathbf{r}(t) = \sqrt{\frac{K\Omega}{K+1}} e^{j(2\pi f_D cos\theta_o t + \phi_o)} + \sqrt{\frac{\Omega}{K+1}} h(t) \tag{5.12}$$

With $f_D$ being the maximum Doppler frequency, namely the ratio of the receiver velocity to the wavelength, $\theta_o$ the angle of arrival, and $\phi_o$ the phase of the LOS. With the K-factor equal to 0 dB, the direct wave and the sum of the reflected ones contribute to the received signal at equal power levels, whereas with profile E the LOS wave is much more prominent.

The above observation serves to explain the severely deteriorated state of the CNN modulation classification deployment with these new, alternative data. However, once again, it is noted that the compression methods followed in earlier experiments still exhibit largely the same trade-offs. As before, especially setting EEPTC (the combination of the Ballé compression architecture with the pre-training of local branches) presents an all-around very high-performance scenario, with a potent compression rate being achieved alongside near-optimal accuracy.

## 5.4 Summary

In the final part of the present work, the effect of cooperative approaches to AMC was examined in the context of a somewhat more challenging setting, namely in a distributed scenario where four receiving devices (for a single transmitter) are located in a geographically open area with obstacles and scatterers, and the CNN classifier resides (in the general case) in a fusion centre independent from the receivers. In addition to previously already present concerns, namely classifier reliability and evaluation time, an additional aspect present in this iteration of the problem is the data load placed upon the communication channels relaying information from the edge nodes to the fusion centre; as such, different methods of organising data flows and distributing the CNN over the whole structure are assessed not only with regard to accuracy but also concerning the aforementioned data loads, and trade-offs between the two.

# Chapter 6

# Conclusion

## 6.1 Summary of Thesis Achievements

As was made clear via the literature review accompanying the present work, automatic modulation classification constitutes a problem still pertinent in contemporary telecommunications research. Through the course of this thesis, the problem was defined, previous approaches to the same considered, and solutions to newly-examined settings of AMC proposed and tested. In particular, the research presented above concerns AMC in the presence of interference, in a SIMO environment, and in a distributed setting.

What is consistently demonstrated throughout this research work is, first and foremost, that state-of-the-art neural network technology, which has already proven its vastly fortified capabilities in other disciplines, is able to offer competitive results with regard to problems of signal identification such as AMC. High-reliability ANN performances are certainly achievable even without the use of highly specialised or complicated architectures, and sometimes even a single model can suffice for a variety of different settings of a problem, given at least some rudimentary alterations.

Additionally, we may confidently deduce that the cooperative approach to supervised identification tasks has proven itself a highly promising for the optimisation of AMC and possibly

other signal recognition tasks. Even when datasets are used which are of a smaller size, and/or suffer degradation from a wide variety of environmental factors, the combination of information resulting from multiple different "views" of a data point can lead to dramatic performance improvements for the classifying algorithm.

In the first part of the work, the simple point-to-point setup for modulation recognition is subject to complication via the introduction of an interferer. This secondary device introduces an additional source of undesired influence of an extant CNN classifier, in addition to AWGN and channel impairments, and stronger signal-to-interference power ratios effect a severe impact upon detection accuracy. The incorporation of the modulation scheme utilised by the interferer as an additional input for the neural network is demonstrated to have a beneficial effect upon performance, especially in cases where the superposed signals from the two transmitters are received at power ratios approaching 1 – the latter observation granting a first insight into the inner functions of the CNN. Furthermore, this part is where we establish a successful usage of a modified fully convolutional neural architecture, in the interest of reducing the number of parameters and improving performance.

Consequently, a new setting was considered atop the usual AMC paradigm, in which the receiving device features multiple antennas and thus has access to data obtained for each signal via many realisations of the channel. This local cooperative approach for modulation recognition shows a clear potential for improvements, resulting in significant gains in classifier accuracy. Issues of neural network operation times are also taken into account, and uniform subsampling introduced as a method of speeding up the classification operation.

In the final part of the present work, the effect of cooperative approaches to AMC was examined in a more challenging context, namely in a distributed scenario where four receiving devices (for a single transmitter) are located in a geographically expansive area with obstacles and scatterers, and the CNN classifier resides in a fusion centre independent from the receivers. In addition to previously already present concerns, namely classifier reliability and evaluation time, an additional aspect present in this iteration of the problem is the data load placed upon the communication channels relaying information from the edge nodes to the fusion centre;

as such, different methods of organising data flows and distributing the CNN over the whole structure are assessed not only with regard to accuracy but also concerning the aforementioned data loads, and trade-offs between the two.

## 6.2 Future Work

Although the coverage of the research presented above aimed at a comprehensive exploration of AMC in a variety of settings, there are still some highly promising directions of future research which could constitute a continuation.

### Generalising the AMC Algorithm

An assessment of the generalisation capabilities of a DL-based classification algorithm typically has been part of this work from the beginning, in the computation of accuracy scores on a subset of the data (validation/testing set) which was not used for training. However, a direction more seldom pursued is the capability of an NN classifier to detect a new class of data and, instead of misclassifying the new instances into one of the existing categories, upgrade the structure so as to adapt to an increased number of classes.

Therefore, a simple expansion of the functions of any one of the proposed algorithms would be the implementation of a "signal belonging to newly-detected class" alert capability. Theoretically, this need not be modelled as an additional output class for the CNN, but rather, the presence of a new modulation scheme would be determined e.g. based on a low degree of confidence in the classifier's decision. [BMT20] The plausibility of such an idea is corroborated by the concept of unsupervised and semi-supervised learning [TC19], where a neural network's ability to be trained for the extraction of useful features from data without explicit instruction pointing to target classes is employed in the interest of training NNs with datasets which are at least partly unlabelled.

### Specific Emitter Identification

As already mentioned in our introduction, specific emitter identification constitutes a subcategory of signal identification research which is parallel to AMC. Its exact scope is the proposition of methods for the recognition of the nature of the particular device responsible for a received signal based on distortions which are manifest in the signal [ZHC16][HWLW19][QHM19] (in juxtaposition to AMC, where the identity of the transmitter under evaluation depends on a deliberate characteristic of the signal). Because SEI relies not only on the primary characteristics of a signal, but also takes into account the manifestations of hardware imperfections (impairments) which are a feature of the originating device, and it can be argued that those impairments are unique to each device [DZC12], it is theoretically plausible that SEI techniques may be utilised for any number of applications ranging from detecting different kinds of devices to differentiating between models of the same root device.

Although the usage of neural network algorithms has been making its appearance with a somewhat increasing frequency in recent SEI literature [JWXL20][PYP+19][DWWZ18], the algorithms employed for the identification tasks are in most cases still heavily reliant on the extraction of expert features [MWM19], and are typically concerned with small numbers of receivers (even as few as 4) and high SNR values. As such, a direction of research which could constitute a continuation of the present work would be a survey of the effectiveness in SEI tasks of the FCNN which we have extensively employed (or other architectures) without feature extraction, with greater numbers of candidate classes, and also with consideration of distributed settings.

### Adversarial robustness

An adversarial attack upon a neural network happens when a malicious agent corrupts input data with the purpose of confusing the network and leading to misclassifications. Typically, adversarial attacks are carried out through the introduction of perturbations aimed at maximising confusion in the confidence of the classifier's decision and increasing error probabilities, in conjunction with certain upper bounds which ensure that the compromised data samples

are not easily distinguishable from genuine ones by a human observer [HPG+17][RHF18]; this process is often framed as a search for the minimum perturbation which will lead to reduced accuracy [UAQ+19].

Since AMC, as mentioned in the introduction, is of particular usefulness in military applications, among others, it follows that an assessment of the robustness of related algorithms to adversarial attacks ought to be of high priority. As such, an additional extension of the present work could be pursued in that direction, and if possible, the crafting of appropriate defence mechanisms should also be considered.

### Optimisation of interfaces and speed

While the minimisation of inference time is an issue which has concerned a certain proportion of AMC literature in recent years, the overwhelming majority of this body of knowledge is almost exclusively focused upon the architecture and data side of the problem, i.e., what is researched is how different DNN models and data pre-processing or combination methods can affect the time required for operations and classification. While hardware specifications are sometimes mentioned, these typically constitute a rather inconsequential part of the works which they accompany. A recent challenge which has been presented at *AI for Good*, an online platform hosting year-round presentations and problems concerning all areas of artificial intelligence, attempts to compensate for the aforementioned lack of research output on the issues of hardware optimisation pertaining to AMC [Lig].

Beginning with a general assessment of the components of a modulation recognition system based on neural classifiers, the authors conclude that specialisation is necessary both on the neural network part (e.g., through quantisation or sparsity) and the supporting hardware part (e.g., through flexible arithmetics and increased internal bandwidth). It is deduced that the deciding factor determining inference throughput is the cost of multiply-accumulate (MAC) operations and weights, which is, in turn, influenced by parameters such as:

- Number of MACs.

- Weight storage required.
- Operation precision.

Field programmable gate arrays (FPGAs) are suggested as readily available equipment suitable for meeting the aforementioned goals, as they can provide capabilities of high specialisation for DNNs, not least because of features such as:

- Arbitrary resolution.
- Mixed-bitwidth environments.
- Fine-grained sparsity.
- Parallelisation of layer operations.

It can be conluded, therefore, that this could potentially constitute the strongest area for further research concerning AMC and other signal recognition problems.

# Bibliography


[AMM13] Anup Aprem, Chandra R. Murthy, and Neelesh B. Mehta. Transmit power control policies for energy harvesting sensors with retransmissions. *IEEE Journal of Selected Topics in Signal Processing*, 7(5):895–906, 2013.

[AYSP20] S. Ahn, D. Yoon, H. Shim, and S. Park. Performance analysis of modulation classification with a preprocessing. In *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1519–1521, 2020.

[BBGO08] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. Wireless device identification with radiometric signatures. pages 116–127, 01 2008.

[BCP+03] Ruud Bolle, J Connell, S Pankanti, N Ratha, and A Senior. Guide to Biometrics Springer Professional Computing, 2003.

[BDA20] Eren Balevi, Akash Doshi, and Jeffrey G. Andrews. Massive mimo channel estimation with an untrained deep neural network. *IEEE Transactions on Wireless Communications*, 19(3):2079–2090, 2020.

[BDBF+19] Jose Balsa, Tomás Domínguez-Bolaño, Óscar Fresnedo, José A. García-Naya, and Luis Castedo. Transmission of still images using low-complexity analog joint source-channel coding. *Sensors*, 19(13):2932, Jul 2019.

[BLS17] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end optimized image compression, 2017.

[BMS+18] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior, 2018.

[BMT20] Akshaya B and Kala M T. Convolutional neural network based image classification and new class detection. In *2020 International Conference on Power, Instrumentation, Control and Computing (PICC)*, pages 1–6, 2020.

[CCGZ17] B. Chen, J. Chen, Y. Gao, and J. Zhang. Coexistence of lte-laa and wi-fi on 5 ghz with corresponding deployment scenarios: A survey. *IEEE Communications Surveys Tutorials*, 19(1):7–32, 2017.

[CCS+19] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah. Artificial neural networks-based machine learning for wireless networks: A tutorial. *IEEE Communications Surveys Tutorials*, 21(4):3039–3071, 2019.

[CGT+10] W. E. Cobb, E. W. Garcia, M. A. Temple, R. O. Baldwin, and Y. C. Kim. Physical layer identification of embedded devices using rf-dna fingerprinting. In *2010 - MILCOM 2010 MILITARY COMMUNICATIONS CONFERENCE*, pages 2168–2173, Oct 2010.

[Cla99] Trevor Clarkson. Applications of neural networks in telecommunications. In *Proc. ERUDIT Workshop on Application of Computational Intelligence Techniques in Telecommunication*, 1999.

[CMH16] J. Choi, J. Mo, and R. W. Heath. Near maximum-likelihood detector and channel estimator for uplink multiuser massive mimo systems with one-bit adcs. *IEEE Transactions on Communications*, 64(5):2005–2018, May 2016.

[DABNS07] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su. Survey of automatic modulation classification techniques: classical approaches and new trends. *IET Communications*, 1(2):137–156, April 2007.

[DDB20] P Dileep, Dibyaiyoti Das, and Prabin Bora. Dense layer dropout based cnn architecture for automatic modulation classification. pages 1–5, 02 2020.

[DOP+14] Brad K. Donohoo, Chris Ohlsen, Sudeep Pasricha, Yi Xiang, and Charles Anderson. Context-aware energy enhancements for smart mobile devices. *IEEE Transactions on Mobile Computing*, 13(8):1720–1732, 2014.

[DWWZ18] Lida Ding, Shilian Wang, Fanggang Wang, and Wei Zhang. Specific emitter identification via convolutional neural networks. *IEEE Communications Letters*, 22(12):2591–2594, 2018.

[DZC12] Boris Danev, Davide Zanetti, and Srdjan Capkun. On Physical-layer Identification of Wireless Devices. *ACM Comput. Surv.*, 45(1):6:1–6:29, December 2012.

[GB10] Xavier Glorot and Y. Bengio. Understanding the difficulty of training deep feed-forward neural networks. *Journal of Machine Learning Research - Proceedings Track*, 9:249–256, 01 2010.

[GGR04] Matteo Gandetto, Marco Guainazzo, and Carlo S. Regazzoni. Use of time-frequency analysis and neural networks for mode identification in a wireless software-defined radio approach. *EURASIP Journal on Advances in Signal Processing*, 2004(12):863653, Sep 2004.

[Han18] Boris Hanin. Which neural net architectures give rise to exploding and vanishing gradients?, 2018.

[HBK04] Jeyanthi Hall, Michel Barbeau, and Evangelos Kranakis. Enhancing intrusion detection in wireless networks using radio frequency fingerprinting. *Communications, Internet, and Information Technology*, 01 2004.

[HDCP10] Fahed Hameed, Octavia Dobre, and Dimitrie C. Popescu. On the likelihood-based approach to modulation classification. *IEEE Trans. Wireless Comms.*, 8:5884 – 5892, 01 2010.

[HGKL20] A. P. Hermawan, R. R. Ginanjar, D. Kim, and J. Lee. Cnn-based automatic modulation classification for beyond 5g communications. *IEEE Communications Letters*, 24(5):1038–1041, 2020.

[HPG+17] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies, 2017.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[HWLW19] B. He, F. Wang, Y. Liu, and S. Wang. Specific emitter identification via multiple distorted receivers. In *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6, May 2019.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[HZST20] J. He, Y. Zhou, J. Shi, and Q. Tang. Modulation classification method based on clustering and gaussian model analysis for vlc system. *IEEE Photonics Technology Letters*, 32(11):651–654, 2020.

[JCT+17] P. Joshi, D. Colombi, B. Thors, L. Larsson, and C. Törnevik. Output power levels of 4g user equipment and implications on realistic rf emf exposure assessments. *IEEE Access*, 5:4545–4550, 2017.

[JCZ19] Z. Jia, W. Cheng, and H. Zhang. A partial learning-based detection scheme for massive mimo. *IEEE Wireless Communications Letters*, 8(4):1137–1140, Aug 2019.

[JPO+18] J. Jagannath, N. Polosky, D. O'Connor, L. N. Theagarajan, B. Sheaffer, S. Foulke, and P. K. Varshney. Artificial neural network based automatic modulation classification over a software defined radio testbed. In *IEEE Int'l Conf. on Comm. (ICC)*, pages 1–6, May 2018.

[JWXL20] Hao Ji, Tao Wan, Wanan Xiong, and Jingyi Liao. A method for specific emitter identification based on surrounding-line bispectrum and convolutional neural network. In *2020 IEEE 3rd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, pages 328–332, 2020.

[JZR+17] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. C. Chen, and L. Hanzo. Machine learning paradigms for next-generation wireless networks. *IEEE Wireless Communications*, 24(2):98–105, April 2017.

[KB14] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[KKC+16] B. Kim, J. Kim, H. Chae, D. Yoon, and J. W. Choi. Deep neural network-based automatic modulation classification technique. In *Int'l Conf. Inform. and Comm. Tech. Conv. (ICTC)*, pages 579–582, Oct 2016.

[KKMP17] Merima Kulin, Tarik Kazaz, Ingrid Moerman, and Eli De Poorter. End-to-end learning from spectrum data: A deep learning approach for wireless signal identification in spectrum monitoring applications. *CoRR*, abs/1712.03987, 2017.

[KSJY20] Y. Kumar, M. Sheoran, G. Jajoo, and S. K. Yadav. Automatic modulation classification based on constellation density using deep learning. *IEEE Communications Letters*, 24(6):1275–1278, 2020.

[LCY14] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[Ld17] L. T. N. Landau and R. C. de Lamare. Branch-and-bound precoding for multiuser mimo systems with 1-bit quantization. *IEEE Wireless Communications Letters*, 6(6):770–773, Dec 2017.

[LFY19] H. Liang, W. Fu, and F. Yi. A survey of recent advances in transfer learning. In *2019 IEEE 19th International Conference on Communication Technology (ICCT)*, pages 1516–1523, 2019.

[Lig] Lightning-fast modulation classification with hardware-efficient neural networks. `https://aiforgood.itu.int/event/lightning-fast-modulation-classification-with-hardware-efficient-neural-netw` Accessed: 2021-08-10.

[LPKQ16] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *CoRR*, abs/1603.02199, 2016.

[LSD14] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[LYG17] Xiaoyu Liu, Diyu Yang, and Aly El Gamal. Deep neural network architectures for modulation classification. *CoRR*, abs/1712.00443, 2017.

[LYSX18] A. Liu, Y. Yang, Q. Sun, and Q. Xu. A deep fully convolution neural network for semantic segmentation based on adaptive feature fusion. In *2018 5th International Conference on Information Science and Control Engineering (ICISCE)*, pages 16–20, July 2018.

[MATa] Filter signal through 802.11ac multipath fading channel. `https://uk.mathworks.com/help/wlan/ref/wlantgacchannel-system-object.html`. Accessed: 2021-Apr-3.

[MATb] Modulation classification with deep learning. `https://uk.mathworks.com/help/deeplearning/examples/modulation-classification-with-deep-learning.html`. Accessed: 2019-11-06.

[MCWW18] F. Meng, P. Chen, L. Wu, and X. Wang. Automatic modulation classification: A deep learning enabled approach. *IEEE Transactions on Vehicular Technology*, 67(11):10760–10772, 2018.

[MD12] G. B. Markovic and M. L. Dukic. Cooperative amc schemes using cumulants with hard and soft decision fusion. In *2012 20th Telecommunications Forum (TELFOR)*, pages 400–403, Nov 2012.

[MHN13] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[MKK+19] S. Moon, I. Kim, D. Kam, D. Jee, J. Choi, and Y. Lee. Massive mimo systems with low-resolution adcs: Baseband energy consumption vs. symbol detection performance. *IEEE Access*, 7:6650–6660, 2019.

[MS15] Setareh Maghsudi and Sławomir Stańczak. Channel selection for network-assisted d2d communication via no-regret bandit learning with calibrated forecasting. *IEEE Transactions on Wireless Communications*, 14(3):1309–1322, 2015.

[MW01] David Meyer and FH Technikum Wien. Support vector machines. *R News*, 1(3):23–26, 2001.

[MWM19] Jason M. McGinthy, Lauren J. Wong, and Alan J. Michaels. Groundwork for neural network-based specific emitter identification authentication for iot. *IEEE Internet of Things Journal*, 6(4):6429–6440, 2019.

[MYG19] Mahdi Boloursaz Mashhadi, Qianqian Yang, and Deniz Gündüz. Cnn-based analog CSI feedback in FDD MIMO-OFDM systems. *CoRR*, abs/1910.10428, 2019.

[OC16] Timothy J. O'Shea and Johnathan Corgan. Convolutional radio modulation recognition networks. *CoRR*, abs/1602.04105, 2016.

[OCM16] Timothy J. O'Shea, T. Charles Clancy, and Robert W. McGwier. Recurrent neural radio anomaly detection. *CoRR*, abs/1611.00301, 2016.

[OH17a] T. O'Shea and J. Hoydis. An introduction to deep learning for the physical layer. *IEEE Trans. Cogn. Comm. and Net.*, 3(4):563–575, Dec 2017.

[OH17b] T. O'Shea and J. Hoydis. An introduction to deep learning for the physical layer. *IEEE Transactions on Cognitive Communications and Networking*, 3(4):563–575, Dec 2017.

[OWVC17a] T. O'Shea, N. West, M. Vondal, and C. Clancy. Semi-supervised radio signal identification. In *Int'l Conf. Adv. Comm. Tech.*, pages 33–38, Feb 2017.

[OWVC17b] Timothy J O'Shea, Nathan West, Matthew Vondal, and T Charles Clancy. Semi-supervised radio signal identification. In *Advanced Communication Technology (ICACT), 2017 19th International Conference on*, pages 33–38. IEEE, 2017.

[Pet09] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[PG14] A. C. Polak and D. L. Goeckel. Wireless device identification based on rf oscillator imperfections. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2679–2683, May 2014.

[PJW+19] S. Peng, H. Jiang, H. Wang, H. Alwageed, Y. Zhou, M. M. Sebdani, and Y. Yao. Modulation classification based on signal constellation diagrams and deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(3):718–727, 2019.

[PMB13] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, page III–1310–III–1318. JMLR.org, 2013.

[PYP+19] Yiwei Pan, Sihan Yang, Hua Peng, Tianyun Li, and Wenya Wang. Specific emitter identification based on deep residual networks. *IEEE Access*, 7:54425–54434, 2019.

[QHM19] Yao Qun, Chai Heng, and Gao Moyun. Radar specific emitter identification using carrier frequency feature. In *2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP)*, pages 1–4, 2019.

[RC07] Kasper Rasmussen and Srdjan Capkun. Implications of radio fingerprinting on the security of sensor networks. pages 331 – 340, 10 2007.

[RDAS+20] Francesco Restuccia, Salvatore D'Oro, Amani Al-Shawabka, Bruno Costa Rendon, Stratis Ioannidis, and Tommaso Melodia. Deepfir: Addressing the wireless channel action in physical-layer deep learning, 2020.

[RGJ+05] K. A. Remley, C. A. Grosvenor, R. T. Johnk, D. R. Novotny, P. D. Hale, M. D. McKinley, A. Karygiannis, and E. Antonakakis. Electromagnetic signatures of wlan cards and network security. In *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology, 2005.*, pages 484–488, Dec 2005.

[RHF18] Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack, 2018.

[RJY+19] Sharan Ramjee, Shengtai Ju, Diyu Yang, Xiaoyu Liu, Aly El Gamal, and Yonina C. Eldar. Fast deep learning for automatic modulation classification, 2019.

[RJY+20] Sharan Ramjee, Shengtai Ju, Diyu Yang, Xiaoyu Liu, Aly El Gamal, and Yonina C. Eldar. Ensemble wrapper subsampling for deep modulation classification, 2020.

[RL81] J. Rissanen and G. Langdon. Universal modeling and coding. *IEEE Transactions on Information Theory*, 27(1):12–23, 1981.

[RMG+18] Sreeraj Rajendran, Wannes Meert, Domenico Giustiniano, Vincent Lenders, and Sofie Pollin. Deep learning models for wireless signal classification with distributed low-cost spectrum sensors. *IEEE Transactions on Cognitive Communications and Networking*, 4(3):433–445, 2018.

[SBSB] Venkatesh Sathyanarayanan, Jahya Burke, Rui Shang, and Richard Bell. Modulation classification using neural networks. `http://noiselab.ucsd.edu/ECE228_2019/Reports/Report42.pdf`. Accessed: 2022-Apr-21.

[SEA03] A. Sendonaris, E. Erkip, and B. Aazhang. User cooperation diversity. part i. system description. *IEEE Transactions on Communications*, 51(11):1927–1938, 2003.

[SG17] S. Sun and X. Gu. Support vector machine equipped with deep convolutional features for product reviews classification. In *Int'l Conf. Nat. Comp., Fuzzy Sys. and Knowledge Disc.*, pages 130–135, July 2017.

[SZL15] L. Shao, F. Zhu, and X. Li. Transfer learning for visual categorization: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 26(5):1019–1034, 2015.

[TAG03] C. Tepedelenlioglu, A. Abdi, and G.B. Giannakis. The ricean k factor: estimation and performance analysis. *IEEE Transactions on Wireless Communications*, 2(4):799–810, 2003.

[TBLS19] S. Tridgell, D. Boland, P. H. W. Leong, and S. Siddhartha. Real-time automatic modulation classification. In *2019 International Conference on Field-Programmable Technology (ICFPT)*, pages 299–302, Dec 2019.

[TC19] Yuan Tian and Marc Compere. A case study on visual-inertial odometry using supervised, semi-supervised and unsupervised learning methods. In *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pages 203–2034, 2019.

[TKD08] K. Tsagkaris, A. Katidiotis, and P. Demestichas. Neural network-based learning schemes for cognitive radio systems. *Computer Communications*, 31(14):3394 – 3404, 2008.

[TRPC09] Nils Ole Tippenhauer, Kasper Rasmussen, Christina Pöpper, and Srdjan Capkun. Attacks on public wlan-based positioning systems. pages 29–40, 01 2009.

[TTCG19] P. Triantaris, E. Tsimbalo, W. H. Chin, and D. Gündüz. Automatic modulation classification in the presence of interference. In *2019 European Conference on Networks and Communications (EuCNC)*, pages 549–553, 2019.

[TWN10] V. K. Tumuluru, P. Wang, and D. Niyato. A neural network based spectrum prediction scheme for cognitive radio. In *2010 IEEE International Conference on Communications*, pages 1–5, May 2010.

[UAQ+19] Muhammad Usama, Muhammad Asim, Junaid Qadir, Ala Al-Fuqaha, and Muhammad Ali Imran. Adversarial machine learning attack on modulation classification, 2019.

[WJV+15] T. Wimalajeewa, J. Jagannath, P. K. Varshney, A. Drozd, and W. Su. Distributed asynchronous modulation classification based on hybrid maximum likelihood approach. In *IEEE Military Comms. Conf.*, pages 1519–1523, Oct 2015.

[WLY12] Y. Wen, S. Loyka, and A. Yongacoglu. Asymptotic analysis of interference in cognitive radio networks. *IEEE Journal on Selected Areas in Communications*, 30(10):2040–2052, November 2012.

[WOA05] B. Wang, S. Omatu, and T. Abe. Identification of the defective transmission devices using the wavelet transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):919–928, 2005.

[WSK18] Chao-Yuan Wu, Nayan Singhal, and Philipp Krähenbühl. Video compression through image interpolation. *CoRR*, abs/1804.06919, 2018.

[WYO16] Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. *CoRR*, abs/1611.06455, 2016.

[XOIH12] Ming Xia, Yasunori Owada, Masugi Inoue, and Hiroaki Harai. Optical and wireless hybrid access networks: Design and optimization. *IEEE/OSA Journal of Optical Communications and Networking*, 4(10):749–759, 2012.

[YMG19] Qianqian Yang, Mahdi Boloursaz Mashhadi, and Deniz Gündüz. Deep convolutional compression for massive MIMO CSI feedback. *CoRR*, abs/1907.02942, 2019.

[ZAF+19] Pengkai Zhu, Durmus Alp Emre Acar, Nan Feng, Prateek Jain, and Venkatesh Saligrama. Cost aware inference for iot devices. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89

of *Proceedings of Machine Learning Research*, pages 2770–2779. PMLR, 16–18 Apr 2019.

[ZDL+18] J. Zhang, L. Dai, X. Li, Y. Liu, and L. Hanzo. On low-resolution adcs in practical 5g millimeter-wave massive mimo systems. *IEEE Communications Magazine*, 56(7):205–211, July 2018.

[ZDL+20] D. Zhang, W. Ding, C. Liu, H. Wang, and B. Zhang. Modulated autocorrelation convolution networks for automatic modulation classification based on small sample set. *IEEE Access*, 8:27097–27105, 2020.

[ZHC16] F. Zhuo, Y. Huang, and J. Chen. Specific emitter identification based on linear polynomial fitting of the energy envelope. In *2016 6th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pages 278–281, June 2016.

[ZLHZ10] Yonghong Zeng, Ying-Chang Liang, Anh Tuan Hoang, and Rui Zhang. A review on spectrum sensing for cognitive radio: Challenges and solutions. *EURASIP Journal on Adv. in Signal Proc.*, (1), Jan 2010.

[ZN15] Z. Zhu and A.K. Nandi. *Automatic Modulation Classification: Principles, Algorithms and Applications*. Wiley, 2015.

[ZQCY19] S. Zheng, P. Qi, S. Chen, and X. Yang. Fusion methods for cnn-based automatic modulation classification. *IEEE Access*, 7:66496–66504, 2019.

[ZYWZ21] Rui Zhang, Zhendong Yin, Zhilu Wu, and Siyang Zhou. A novel automatic modulation classification method using attention mechanism and hybrid parallel neural network. *Applied Sciences*, 11(3):1327, Feb 2021.