MONASH University

Deep Learning-Based Demodulation For Narrowband Communication Systems

Kim Chia

Doctor of Philosophy

A Thesis Submitted for the Degree of Doctor of Philosophy at
Monash University in 2023
School of Information Technology

Copyright notice

Abstract

Direct conversion, or zero intermediate frequency (IF) receivers still remain popular for use in radio frequency (RF) communication due to advantages in terms of monetary cost, power consumption, and physical size. However, the trade-off of a zero IF receiver is that it struggles with noisy, faded, and distorted signals as the signal process chain in hardware cannot fully compensate for the error, causing a drop in received signal quality. Renewed interest in applying neural networks (NN) to the field of data and signal processing, including RF communications, stemmed from the many successes of deep learning (DL) applications across multiple sectors. Advances of DL has lead to the consideration of a new perspective of radio communication receiver architectures that allow flexibility in design, ease of system update, and improved bit-error performance. In this thesis, various NN architectures suited for demodulation task in resource-constrained RF communication systems, such as those found in land mobile radios, are examined. A novel NN demodulator architecture suitable for digitally demodulating noisy, faded, and distorted signals is proposed. The target modulation studied is the linear modulation M-order Quadrature Amplitude Modulation (M-QAM), selected as an ideal modulation to observe the impact of noise. When trained in additive white Gaussian noise (AWGN) and Rayleigh fading channel conditions, the proposed architecture is able to learn better performance than traditional demodulators, such as the coherent demodulator with improved tolerance to channel impairments. Optimisation and compression method provides added benefit to the proposed architecture whereby similar performance to other DL architectures can be achieved with only a quarter of the memory and processing power required. Furthermore, an NN architecture trained specifically for signal waveform recovery is proposed with a framework to demodulate noisy, faded and distorted signals by reconstructing the signal back into their clean state to ease symbol detection. The proposed NN models are able to gain up to 3dB improvement in SNR over an optimal demodulator. The findings of our research could be used to support the classification of noisy and distorted time series data with large classes of up to 64 classes. It is hoped that this research work will lead to a future DL-based demodulator that can support the demodulation of a wider range of modulation techniques and modulation order in this space and overcome traditional constraints in conventional receivers by the application of DL.

Declaration

This thesis is an original work of my research and contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, or any use of generative artificial intelligence technologies, except where due reference is made in the text of the thesis.

Signature: _____

Print Name: Kim Chia

Date: _____

Publications during enrolment

K. Chia and V. M. Baskaran, "Deep Learning-Based Demodulation of Radio Signal," 2022 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Penang, Malaysia, 2022, pp. 1-4, doi: 10.1109/ISPACS57703.2022.10082826.

K. Chia, V. M. Baskaran, K. Wong, M. L. Sim and C. H. Chee, "Recurrent Network with Attention for Symbol Detection in Communication Systems," 2022 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Penang, Malaysia, 2022, pp. 1-4, doi: 10.1109/ISPACS57703.2022.10082803.

Acknowledgements

I acknowledge the use of https://www.grammarly.com/ to Grammarly, an automated proofreading tool. The output from these was used to assist with grammar, punctuation, and clarity in my thesis. Grammarly's assistance was limited to English language refinement and did not influence the substantive content or structure of my work. In addition to using Grammarly for proofreading, I also acknowledge the use of https://chatgpt.com to ChatGPT, a generative artificial intelligence chatbot. The output from these was used to help summarise and improve some sections of my writing during thesis revision post-submission. However, all content in this thesis was originally written by myself prior to ChatGPT's existence. I declare that ChatGPT's assistance was limited to language refinement and summarisation without influencing the substance or structure of my work. One primary example of the prompt I used to generate a response is by asking "my original paragraph rephase the paragraph above that introduces how an rf communication system works in a scholarly, academic and precise manner", where the output of this response will be taken as a reference.

This research is funded and supported by The Collaborative Research in Engineering, Science and Technology Centre (CREST), Monash University Malaysia and Motorola Solutions, which would not have been possible without their aid in finance and resources. I am most grateful to be provided with the opportunity to conduct research in the area of deep learning and signal processing and contribute to knowledge.

There are many parties who helped make this research possible. In particular, I would like to thank my industrial advisor, Dr. Sim Moh Lim for his encouragement and guidance, and for generously sharing his knowledge in mathematics and signal processing. I must also thank my main supervisor and advisor Dr. Vishnu Monn Baskaran for his supervision and guidance in this research as well as the many discussions we have had to ensure the research project is on track. Not to forget Dr. Mohamed Hisham and my co-supervisor Dr. Wong Koksheik whom have contributed many helpful discussions and ideas for improving our publications. Many thanks to my industrial partner Dr. Chee Chong Hin who provided me with information on common issues in land mobile radios and the challenges in the signal process chain. Thank you also to Dr. Tan Chee Keong and Dr. Trung Le, for their helpful feedback on blending the areas of communications and deep learning. I would also like to extend my thanks to my colleague Mr. Ian Wong who provided me with dataset which he collected, and explained to me in details on the process of signal generation and simulation. I am also grateful to Ms. Wan Adleen, the Monash Graduate Research Office Administrator, for her kindness and help in making everything run so smoothly. I am most grateful to Dr. Jasbir Kaur who made this research project possible by gathering the right partnership, grant and formation of the

project team, whose word of encouragement I will always treasure. My friends HK, Boey and Tay have supported me throughout difficult times of my life, they shared the same interests in signal processing and deep learning, therefore discussions and ideas exchanged with them have helped shape this thesis.

Contents

List of Figures

List of Tables

Abbreviations

ADC	**A**nalogue-to-**D**igital **C**onverter
ASK	**A**mplitude-**S**hift **K**eying
AWGN	**A**dditive **W**hite **G**aussian **N**oise
BER	**B**it **E**rror **R**ate
CE	**C**ross-**E**ntropy
CFO	**C**arrier-**F**requency **O**ffset
CNN	**C**onvolutional **N**eural **N**etworks
DAE	**D**enoising **A**uto**E**ncoder
DBN	**D**eep **B**elief **N**etworks
DC	**D**irect **C**urrent
DEMON	**D**ecaying **M**omentum
DSP	**D**igital **S**ignal **P**rocessing
DL	**D**eep **L**earning
FSK	**F**requency-**S**hift **K**eying
GRU	**G**ated **R**ecurrent **U**nit
I	**I**n-phase
IF	**I**ntermediate **F**requency
KLD	**K**ullback-**L**eibler **D**ivergence
LO	**L**ocal **O**scillator
LS	**L**east **S**quare
LSTM	**L**ong **S**hort-**T**erm **M**emory
M2M	**M**achine-**to**-**M**achine
MAE	**M**ean **A**bsolute **E**rror
MAP	**M**aximum **A** **P**osteriori probability
MIMO	**M**ultiple **I**nput **M**ultiple **O**utput

ML	**M**achine **L**earning
MLE	**M**aximum **L**ikelihood **E**stimation
MLP	**M**ulti-**L**ayer **P**erceptron
MSE	**M**ean **S**quared **E**rror
NN	**N**eural **N**etworks
OFDM	**O**rthogonal **F**requency-**D**ivision **M**ultiplexing
PDF	**P**robability **D**ensity **F**unction
PSK	**P**hase-**S**hift **K**eying
Q	**Q**uadrature
QAM	**Q**uadrature **A**mplitude **M**odulation
RE-ATT	**RE**current-**ATT**ention Networks
ResNet	Deep **Res**idual **Net**works
RF	**R**adio **F**requency
RNN	**R**ecurrent **N**eural **N**etworks
SGD	**S**tochastic **G**radient **D**escend
SISO	**S**ingle **I**nput **S**ingle **O**utput
SNR	**S**ignal-to-**N**oise **R**atio
SOTA	**S**tate-**O**f-**T**he-**A**rt
SVM	**S**upport **V**ector **M**achine
TDMA	**T**ime-**D**ivision **M**ultiple **A**ccess
UHF	**U**ltra-**H**igh **F**requency

Abbreviations

b	Bit
dB	decibel
Hz	Hertz
s	Second

Constants

Speed of Light	c	=	$2.997\ 924\ 58 \times 10^8\ \mathrm{ms}^{-\mathrm{S}}$ (exact)
Pi	π	=	3.1415926535897932
Complex Value	j	=	$\sqrt{-1}$

Symbols

a	distance	m
P	power	W (Js^{-1})
ω	angular frequency	rads^{-1}
φ	phase	rads / $^\circ$

Chapter 1

Introduction

1.1 Background

Our lives cannot be separated from radio frequency (RF) technologies. For many years, RF became the critical technology to enable communication. Reliance on pervasive radio communication systems can be seen across various domestic, commercial, industrial, public service and military sectors. In specific sectors, RF communication plays a critical role in productive activities to remain operating. One key technology that is in use is called narrowband radio. Narrowband radio operates over a very narrow bandwidth in the range of a few kilohertz in the licensed RF spectrum to a few hundred megahertz in the unlicensed RF spectrum.

Despite being relatively limited in bandwidth compared to broadband or wideband radios seen in newer technologies such as millimeter-waves, narrowband RF continues to be used and preferred in various applications such as LoRa [1], Sigfox [2], IEEE 802.11ah [3] etc. Compared to wideband signals, narrowband signals exhibit superior propagation characteristics that allow longer range transmission and better penetration through obstacles such as buildings, walls, and foliage. Due to their propagation characteristics, narrowband radios typically consume less power than their wideband counterparts. This power efficiency translates into longer battery life for devices utilising narrowband communication. Consequently, this makes narrowband radios ideal for applications requiring

[1] https://www.lora-alliance.org
[2] https://www.sigfox.com
[3] https://standards.ieee.org/findstds/standard/802.11ah-2016.html

comprehensive area coverage and reliable communication in challenging environments, such as remote monitoring, outdoor telemetry, rural communications and machine-to-machine (M2M) communication, where power availability is limited or extended operational duration is required.

Narrowband radios use a smaller portion of the radio frequency spectrum compared to wideband radios. As the radio spectrum is a finite resource, efficient utilisation becomes crucial. While several techniques and mitigation can be done to ensure multiple narrowband channels can coexist within a given frequency band, increasingly saturated RF channels have resulted in a low signal-to-noise ratio. A crowded spectrum environment heavily impacts the quality of end-user communication and increases the challenge of maintaining signal quality [2]. This is especially important in crowded spectrum environments, such as urban areas or regions with a high density of RF systems operating simultaneously.

An RF communication system comprises essential components, including a terminal and a propagation channel, typically the air medium. The terminal can function as a transmitter, receiver, or transceiver for transmitting and receiving analogue or digital data. Simply put, the transmitter encompasses a carrier wave generator, a modulator, and an amplifier. Conversely, the receiver includes an amplifier, a demodulator, a decoder, and an output transducer. Modulation and demodulation processes play a vital role in these systems. The use of quadrature amplitude modulation (QAM) for efficient data bandwidth in narrowband RF communication has introduced additional design complexities, particularly in the performance of transmitters and receivers. RF communication systems utilising QAM generally employ linear transmitters and coherent demodulators.

RF communication systems today are not only required to have more throughput due to the high demands of different applications. There is also a strong demand for cost and power-effective RF communication systems with a consistent performance under various environments, on top of a priority to control the size of the transceiver itself. Transceivers with homodyne, or direct conversion, also known as the zero intermediate frequency (IF) architecture is one that satisfies the aforementioned requirements. Zero IF receivers are preferred over conventional superheterodyne or IF receivers. Due the received signal being amplified and filtered at baseband rather than at a specific IF, zero IF receivers do not require a separate image filtering commonly found in superheterodyne receivers

that can be costly and bulky in size. However, the choice of a zero IF architecture is in fact sub-optimal.

Zero IF receivers face issues such as mixer imperfections (in-phase and quadrature imbalance) [3], oscillator imperfections (direct current offset) [4], white noise, and fading [1]. To mitigate both the effects of hardware impairments and channels effects, reliance of digital signal processing (DSP) has become instrumental in mitigating these hardware impairments and channel effects. However, transceivers are still subjected to performance limitations in the signal processing chain that can lead to information corruption and difficulties in demodulation and detection of received signals.

The degradation of demodulator performance becomes more pronounced with higher-order modulation schemes. Receiver sensitivity plays a critical role in RF communication systems, directly impacting the communication range. Doubling the receiver sensitivity theoretically yields the same range extension as doubling the transmitter power. However, environmental factors and noise sources can limit the optimal sensitivity of the receiver. The complexity, limitations, and trade-offs in architectural designs motivate research into using deep learning (DL) as a replacement for conventional hardware solutions in order to enhance receiver performance. Leveraging DL in the DSP chain and data-driven methods holds promise in achieving improved receiver performance with simpler architectures without the trade-offs associated with conventional approaches. This avenue of research holds potential as a key technology for next-generation RF communications.

1.2 Statement of Problems

As modern society develops, requirements for RF communications to achieve stability, robustness and reliability become increasingly challenging, especially for narrowband RF with finite spectrum. Channel effects are a bottleneck that limits the performance of a communications system. Due to the randomness, sparsity, intensiveness and complication, conventional state-of-the-art (SOTA) methods have several drawbacks in combating the channel effects. Apart from that, narrowband RF communication is already so well-established that it has reached a plateau where any further significant performance

improvement cannot be realised with hardware optimisation alone. To this day, demodulation of the linear modulation signal is still performed using a coherent demodulator applied with correlation [5] whose concept first came about in 1978 [6].

As channel effects in RF communication systems are a lengthy topic, this research focuses on addressing fundamental problems associated with channel effects. In particular, the problems faced in the process of demodulation in RF communication systems.

I. Channel Effects and Signal Distortions. Noise within the system, referred to as system intrinsic noise, is typically random in distribution. Mitigating or suppressing system intrinsic noise is challenging due to its randomness. However, the intensity of system intrinsic noise is typically not severe for digital RF communications systems, which have larger variations in electric levels compared to analogue systems. In contrast, external noise, originating from deterministic signals, may be distributed randomly and can have intensities far exceeding that of system intrinsic noise. External noise significantly impacts digital RF communications systems. However, the rules governing the distribution of external noise are often easier to determine, allowing for the development of more effective mitigation methods.

Fading, on the other hand, poses a significant challenge as it leads to a loss of signal power without reducing noise power. Linear modulation techniques relying on the amplitude of the signal wave, such as amplitude-shift keying (ASK) and QAM, are particularly affected by fading. The time-varying delay and attenuation caused by fading cannot be readily corrected, unlike other channel impairments such as noise. To combat the effects of a dynamic channel like fading, transmitter signals are often diversified over multiple channels and coherently combined upon reception to reduce the probability of experiencing a fade. However, fading effects change faster than adaptations can be made, resulting in degraded bit error performance during symbol detection. These randomised perturbations in RF signal makes estimation and compensation difficult.

II. Constrains in Conventional Demodulation. During waveform recovery, where unwanted frequency components are filtered out, and the signal voltage is maintained at a detectable level, the receiver's hardware itself can also introduce noise and distortions. Factors such as thermal noise injection from low noise amplifiers (LNA) that are supposed to increase signal-to-noise ratio (SNR), and nonlinearities caused by local oscillator (LO) leakage in mixers contribute to harmonic/intermodulation distortion [2].

Addressing these issues prior to symbol detection adds further complexity to receiver design, as detection of such signals leads to high bit error rates.

Demodulating and detecting the desired signal requires optimal sensitivity and dynamic range, which present trade-offs in conventional receivers [7, 8]. For instance, a passband must be wide enough to accommodate fluctuations in the carrier caused by factors like Doppler shift or transmitter frequency drift instead of only narrow enough for the desired signal. This makes it challenging to detect impaired signals due to unwanted frequency components also able to pass through the filter. Achieving the appropriate balance between passband width and gain poses significant design complexities for RF receivers.

III. Challenges in Time Series Classification. Deterministic signals can be represented by explicit mathematical expressions, allowing for the estimation of missing or future values. In contrast, random signals exhibit unpredictable variations, making it difficult to derive suitable mathematical representations for predicting future values. The complexities introduced by real-world factors, such as noise and fading, further hinder the development of precise quantitative expressions. Consequently, achieving accurate classification or reconstruction of random signals remains a challenging problem.

In RF signal demodulation, the goal is to detect the received signal into one of the predefined symbols corresponding to the modulation scheme used. Each class represents a specific combination of amplitude and phase, and the demodulator's task is to identify which symbol the received signal belongs to. The sequential nature of the time-domain RF signal is crucial in demodulation, as the sequence, context, temporal dependencies and relationship with neighbouring symbols affect the outcome of the symbol detection. All of which aligns with the characteristics of a time series multi-class classification task.

One of the challenges of multi-class time series classification is that most DL algorithms assume the features are independent of each other. That is often not the case in a time series, especially RF signal data, as a value of the series at one time point is dependent on the preceding and succeeding values. While it is understood that time series feature extraction is necessary to identify patterns, it is not clear what features are useful to the classifier; therefore, what was typically done is a deeper and wider network in hopes to create more features where useful features eventually prevail. The computational cost associated with constructing such a model is often very high and unsuitable for the purpose of a low-level signal processing task.

Practical classification problems often contain substantial geometrical structure in the class distributions that manifests as identifiable clusters or patterns. The underlying structure can be difficult to uncover among a vast amount of feature samples in a high-dimensional space due to the impact of noise and distortions resulting in poor quality features generated by the DL model. Consider the time-domain signal data, where in challenging scenarios and channel impairments in the signal, the features representing different symbols may heavily overlap. The time-domain signal consists of only two features attributing to amplitude or power over time, yet is impaired by noise and distortions with randomised distribution and power. The subsequent features generated and learned by the hidden layers can have unintended effects. Learning is limited when it is impossible to fit the model to a highly noisy feature space.

IV. Opportunities with Deep Learning. Due to the complexity of recovering signal impaired by channel is high when dealing with contradictory properties of each signal process, there is great value in finding a way to demodulate without the need of prior estimation and compensation for the impairments. Applications of DL in the DSP chain could be the key technology for next-generation cellular communications. In particular, baseband signalling which is the demodulation of the baseband signal and symbol detection in zero IF transceivers that use narrowband channels. Signal processing with DL can perform symbol detection directly on raw baseband signal, hence alleviates the need to do corrections for unavoidable analogue impairments prior to symbol detection. As such, the focus is turned to DL-based demodulation that identifies patterns in seemingly chaotic time series data to potentially increase the performance of transceivers with simpler architecture without trade-offs from its conventional counterpart.

1.3 Objectives of Research

This research work aims to explore the application of DL in RF communications, with a focus on demodulation tasks. The objective is to provide a unified and principled approach to the selection of DL architecture design and training methodologies, aiming to improve the runtime performance of DL models as well as the clarity of inner workings rather than treating them as black boxes.

The development of a common DL architecture capable of demodulating various modulation orders will offer benefits such as reduced hardware development costs, greater design flexibility, and a shift in the design effort from low-level hardware to a higher-level, system-oriented task. The research seeks to address the trade-offs between spectral efficiency, power efficiency, and receiver sensitivity in conventional RF communication by leveraging DL solutions that are robust to impairments like phase noise and DC offset that are common in low cost receivers.

A significant challenge in demodulation is achieving the distinctiveness of each class group within the feature space to establish clear boundaries. Noise and distortions in the input data can lead to overlapping regions between classes, causing misclassification and high bit-error rates. Thus, the research focuses on enhancing the DL model's ability to handle noisy and distorted data, allowing accurate classification in challenging scenarios through learning discriminative features.

Therefore, the formalised objectives are listed as follows:

1. To model and design of new DL method that improves demodulation of noisy, faded and distorted QAM signal by reducing bit-error compared to conventional method.

2. To train DL-based demodulator with consistent performance even in low SNR scenarios.

3. To optimise the new DL method to increase BER performance while minimise runtime complexity and memory usage.

The objectives 1 and 2 ride on the opportunities in DL to solve the problems caused by channel effects and complexities that occur during demodulation. Objective 2 leverages on DL models' data-driven methodology and learning abilities to provide a better solution than conventional demodulators that is not constrained by mathematical convenience. Objective 3 attempts to overcome problem that arises in a demodulation task that can be defined as a time series classification.

1.4 Contributions of Research

A high-level description of the contributions of this research, highlighted in point form and categorised by topic.

DL-based demodulation:

1. Motivation of the selection of DL-based demodulator architecture by comparing the inner works of conventional demodulator design based on probabilistic models to the architectures, activation functions, and training objectives of modern neural networks.

2. Current SOTA DL demodulators are examined and compared to the ideal demodulator derived in RF communication textbooks, serving as a benchmark for the proposed solutions.

3. A carefully designed DL model architecture, tailored to the unique characteristics of noisy, faded, and distorted RF signals, leads to the formalisation of two novel solutions: the recurrent-attention networks (RE-ATT) model for direct signal-to-symbol demodulation and the Denoising Autoencoder (DAE) model for signal recovery with a classifier of choice. Both solutions have improved performance over other SOTAs and the optimal zero IF receiver for demodulating noisy, faded and distorted signals.

4. Experimental results demonstrating the viability and performance of the proposed architectures for various modulation orders used in current wireless standards are presented, achieving consistent performance for 4-QAM, 16-QAM, and 64-QAM signals.

Optimised Learning:

6. Model compression methods such as zero-weight pruning and quantisation are applied to the RE-ATT to achieve an optimised learning model in terms of training and inferencing speed.

7. Optimisation method such as stochastic gradient descent parameterised by momentum term with a decaying momentum schedule is applied to both the RE-ATT and DAE models, resulting in improved performance and simpler training

with smoother convergence. This eliminates the need for manual hyperparameter tuning and achieves low complexity and more efficient learning.

Improved classification performance:

8. Challenges in the field of DL for the proposed architectures and the shortcomings preventing DL models from achieving better performance are examined. The lack of ability to learn discriminative features in signals with low SNR is identified as a significant problem.

9. A novel regularisation method, the separability metric, is derived and applied to existing objective functions of classifiers to increase intra-class and inter-class separability, leading to improved performance in the proposed classifier RE-ATT.

1.5 Thesis Structure

The thesis is comprised of six chapters, with the thesis structure outlined as follows. Chapter 1 covers an introduction to the research and briefly describes the domain's background. The research problems are highlighted with their corresponding aims and objectives. The contributions of the research will also be mentioned in this section. Finally, the outline and contents of the thesis will also be briefed in Chapter 1. Chapter 2 describes the literature review on three different domains - propagation channel, signal processing, and artificial intelligence (ML/DL), where the majority focus is placed on the SOTAs in DL-based demodulation and symbol detection of time-domain digital baseband signal, as well as the relationship of signal data with various impairments to the DL models' ability to extract salient features. Chapter 3 details the first proposed method of this research to demodulate signals that accurately detect symbols for signals with linear modulation, where it is susceptible to noise and distortions from its surrounding environment. A comparative study of the performance of several spatial and temporal SOTA DL models over the noisy and fading channel is recorded. The SOTA models, as well as the golden standard in DSP, serve as a benchmark for the proposed model. The implementation details, outcome, performance evaluation and experimental analysis with discussion are presented along with a brief of Chapter 3. Chapter 4 exhibits three methods used for the compression, optimisation and fine-tuning of the

proposed method described in Chapter 3. Chapter 5 focuses on the second proposed method in this research that serves as an alternative to the first proposed method to demodulate signals using a two-steps method with the results compared to the latest SOTA. The implementation details, outcome, performance evaluation and experimental analysis with discussion are presented in Chapter 5 as well. Chapter 6 notes the conclusion and potential future works as well as the afterthoughts on the outcome of the research.

Chapter 2

Literature Review

This chapter contains some definitions and knowledge in radio propagation and digital signal processing, as well as a review of all the SOTAs within the domain of signal demodulation.

2.1 Signal

A signal is defined as a medium through which information is transmitted from a source to a destination terminal. Signals can be categorised as either analogue, which exhibits instantaneous variation corresponding to the original information energy, or digital, where predetermined code pulses or variations represent symbols from a defined set [9]. The utilisation of digital transmission offers several advantages over analogue transmission:

1. Enhanced signal-to-noise ratio: In analogue systems, noise and interference accumulate with distance, whereas digital systems are immune to such degradation.
2. Improved signalling capability: Digital transmission allows for a more efficient and reliable representation of signals.
3. Simplified multiplexing: Time division multiplexing, utilised with digital signals, is simpler compared to frequency division multiplexing employed in analogue systems.

4. Straightforward implementation of digital switching: Digital systems facilitate easier switching mechanism implementation.

5. Uniform representation and processing: In digital systems, various types of signals in different signalling applications (e.g., cellular telephony and television) can be represented and processed in a consistent manner during transmission and switching.

A typical RF signal is composed of a sinusoid carrier, $s = s(t)$ formulated with the cosine function and modulated with digital baseband data. The carrier is defined by amplitude $A = A(t)$, phase $\varphi = \varphi(t)$, frequency $\omega = \omega(t)$ and time t such that:

$$s \triangleq A\ cos(\omega\, t + \varphi) \tag{2.1}$$

2.1.1 In-phase and Quadrature Signal Components

In-phase and quadrature signal components, often referred to as I and Q components, are essential elements in RF communication systems. I and Q components are extracted by multiplying the received signal with the local oscillator signal, the in-phase component and the out-of-phase component by $\pi/2$ radians, respectively. Over time, the correlation between I and Q averages to zero, ensuring that each component remains distinct. The orthogonality of these components ensures minimal interference during extraction. Using trigonometric identities, a baseband signal $s = s(t)$ can be decomposed into in-phase s_I and quadrature s_Q components, with amplitudes $A_I = A_I(t)$ and $A_Q = A_Q(t)$:

$$s \triangleq A\ cos(\omega\, t + \varphi) \equiv A_I\ cos(\omega\, t) + A_Q\ sin(\omega\, t) = s_I + s_Q \tag{2.2}$$

whereby

$$\begin{aligned} s_I &= A_I\ cos(\omega\, t) \\ s_Q &= A_Q\ cos(\omega\, t - \pi/2) \\ &= A_Q\ sin(\omega\, t) \end{aligned} \tag{2.3}$$

The complex form of a modulated baseband signal can be expressed as:

$$s = AI + AQj \tag{2.4}$$

where j is the complex constant.

2.1.2 Representation of Signal

One important consideration in analysis and statistical modelling is the representation of data, in the case of signal demodulation, the representation of signal. Signals can be represented in the time domain, frequency domain, or both.

Time domain representation provides insights into how a signal varies over time, including its amplitude, phase, and frequency characteristics. This representation is valuable for analysing the temporal behaviour of signals, particularly in the presence of noise, distortions, and other time-varying phenomena. On the other hand, frequency domain representation offers a perspective on the spectral content of a signal. It enables the identification of frequency components and their contributions to the overall signal. This representation is useful for analysing the spectral characteristics, such as frequency composition, power distribution, harmonic content, and periodicity of signals. However, frequency domain representation often requires complex mathematical operations, like the Fourier transform, to obtain the frequency content from a time domain representation.

The choice between representing a signal in the time domain or frequency domain presents a challenging decision due to the distinct advantages offered by each domain. However, when utilising DL, which exhibits the capability to extract high-dimensional features from lower-dimensional inputs, it becomes essential to evaluate whether the additional cost incurred by converting the time domain to the frequency domain justifies the potential performance improvements. Recent research supports the notion that DL models can effectively operate directly on time domain signals without the need for explicit conversion to the frequency domain. In this research, study is mainly done in time domain and DL models are utilised to learning the underlying components in the signal without incurring additional cost to convert a time domain signal to frequency domain signal.

RF signals in the time domain are typically represented as a time series data, where sequence of amplitude values vary with time. Each signal contains multiple symbols, and each symbol represents one or more bits of information. Signal data can be represented as an input vector with a dimension of $2 \times L_{sps}$, where L_{sps} is the number of samples per symbol. The input vector contains the real and imaginary parts of complex baseband time domain data by interleaving or concatenation. Such data representation is simplistic and aids the learning of feature representation from each individual component.

A method to represent digital modulation signals in a visually comprehensible way is through plotting of a constellation diagram. The signal is portrayed as a scatter plot in the complex plane, using the x-axis, the horizontal or real axis representing I, and y-axis, the vertical or imaginary axis representing Q, to display the signal at symbol sampling instants. Comparable to a phasor diagram, the angle of each point on the constellation diagram is measured counterclockwise from the x-axis, signifying the phase shift of the carrier wave relative to a reference phase. Moreover, the distance from the point of origin of the diagram represents a measure of the signal's amplitude or power. In essence, the constellation diagram provides valuable insights into the distribution of signal points in the complex plane, and it is particularly useful for comprehending channel effects on the signal as it captures perturbations to the amplitude and phase.

2.2 Propagation Channel

The radio propagation from the transmitter to the receiver is influenced by various factors that can lead to the degradation of received signals [2]. These factors encompass distance path loss, slow fading or shadowing, characterised by a lognormal distribution representing power variation, as well as fast fading, resulting from the random power fluctuations caused by moving transmitter or receiver antennas and their surroundings. Additionally, noise in the form of white or pink noise and various types of interference such as co-channel, adjacent channel, and close-in/far-out interference contribute to signal degradation. Within this section, the primary focus lies on examining the constraints imposed on the dimensioning of receivers, which are influenced by the characteristics of the propagation channel — the electromagnetic propagation medium

between the transmit and receive antennas. Notably, these channel characteristics have a significant impact on demodulating signals with linear modulation.

TABLE 2.1: shows several mitigation techniques for combating the effects of both signal distortion caused by fading and loss in SNR caused by white noise [1].

Challenge	Distortion	Loss in SNR
Types	**Frequency-Selective**	**Flat and Slow Fading**
	Adaptive equalisation	Error-correction coding
	Spread spectrum	Uncorrelated signal estimation
	Pilot signal	
	Fast Fading	**Diversity in Space/Time/Frequency**
	Transmit redundant signal	Interleaving
	Coding and interleaving	Bandwidth expansion
		Spread spectrum
		Spaced antennas
		Polarisation

2.2.1 Fading

The wireless propagation channel exhibits much more differences and unpredictability compared to wired channels due to several factors, such as delay spread (time dispersion) and Doppler spread (frequency dispersion). Time dispersion occurs when the coherence bandwidth is smaller than the modulation bandwidth or when the channel is band-limited. Time dispersion can manifest as signal distortion, causing spreading of modulation symbols in time domain which results in inter-symbol interference (ISI). This is because different delayed copies of a signal overlap and interfere with each other due to multipath propagation, making it challenging to correctly demodulate the received signal. Frequency dispersion is evident through frequency shifts. It is a consequence of mobility, either of the transmitter, receiver, or due to systematic ionospheric movements that causes spreading in the signal. While reflections from regular layers typically experience frequency shifts less than 1 Hz, scatter-mode signals at low latitudes have been reported to exhibit shifts of up to 20-30 Hz. In practice, fading occurs as a consequence of variations in propagation paths caused by movements of terminals or varying environmental conditions [2, 9].

Multipath propagation arises when transmitted signals are reflected by obstructions within the environment that can be any physical obstruction including buildings, foliage,

natural geographic landscape, vehicles and etc, resulting in multiple signals arriving at the receiver with random phase offsets due to different path lengths. This leads to fading, where reflections superimpose constructively or destructively [10].

The degree of fading is determined by the time dispersion, measures the extent to which the received signal's power or amplitude fluctuates over time. A signal that experiences a high degree of fading will exhibit large and rapid fluctuations in signal strength, making it more challenging for a receiver to reliably detect and demodulate the signal. The rate of fading is determined by coherence time, the time over which the channel remains constant, defined as the inverse of the Doppler spread that measures the speed where channel characteristics change. Fast fading occurs when such changes are higher than the symbol rate of modulated signal, while slow fading occurs when changes are slower than the symbol rate of modulated signal. In cases of frequency selective fading, the delay spread exceeds the symbol duration. Conversely, fading is flat when there is no dispersion, and the symbol duration is significantly longer than the maximum excess delay spread, which will equally affect all frequencies in the signal and potentially causing deep fades exceeding 30 to 40 dB. Coherence time is given by:

$$\mathcal{T}_c = \frac{1}{2\pi f_d} \tag{2.5}$$

Maximum Doppler spread occurs when the receiver is moving directly towards or away from the source of transmission. The maximum Doppler spread will be determined by the velocity of mobility, v, and the carrier wavelength, λ, given by:

$$f_d = \frac{v}{\lambda} \tag{2.6}$$

Numerous statistical models have been developed to facilitate the understanding of communication channel behaviors. One such model, known as Rayleigh fading, offers insights into the impact of the propagation environment on radio signal transmission. According to the Rayleigh distribution, the amplitude of a signal undergoes random fading when traversing a communication channel. This fading phenomenon typically occurs in scenarios where no strong, direct line-of-sight communication between the transmitter and receiver is established, and that multipath propagation environment exist. Under

this model, transmitted signals follow independent and identically distributed Gaussian distributions, characterised by equal variance and a mean of zero. This can be represented in a random complex rectangular format, encompassing both real and imaginary components. The simulation of fading in RF communication systems often employs Jake's Rayleigh fading model, which provides a framework for describing the normalised channel behaviour. It can be expressed as:

$$h(t) = \sum_{i=1}^{L} A_i \cdot \cos(2\pi f_i t + \varphi_i) \tag{2.7}$$

where $h(t)$ represents the time-varying channel response. L is the number of multipath components, A_i represents the amplitude of the i-th multipath component, f_i represents the Doppler frequency of the i-th multipath component, t is the time variable and $varphi_i$ represents the phase of the i-th multipath component. The probability density function (pdf) of the Gaussian distribution of mean $\mu = 0$, given by:

$$p(A) = (A/\sigma^2) * exp(-A^2/(2\sigma^2)) \tag{2.8}$$

where $p(x)$ represents the pdf, A is the amplitude of the signal while σ is the scale parameter of the Rayleigh distribution. σ is related to spread or dispersion of x by calculating the root mean square value of x and is determined by the characteristics of the fading channel.

Due to multipath propagation, signals received will be characterised in several reflections, diffraction patterns, and scattered waves. The signal amplitude and phases become. Therefore, Rayleigh fading makes it very difficult to perform coherent detection as the fading process results in rapid amplitude and phase changes in the received signal.

Mitigation of fading is often conventionally solved by diversity techniques, grouped into three main categories: space, time, and frequency diversity on either the transmitter or receiver's end. When a receiver has the option to select from an array of signals received at various spatial, temporal, or frequency points, it can potentially circumvent the reduction in SNR attributed to fading. This selection can be executed through various methods, each presenting a distinct trade-off between power consumption and SNR.

Space diversity technique involves either the transmitter or receiver using multiple antennas that are separated for transmission or reception to create an independent signal, and the receiver selects the signal with the strongest strength among the others. There are several space diversity topologies, such as single input multiple output (SIMO), multiple input single output (MISO), and multiple input multiple output (MIMO) systems. Implementation of space diversity increases the complexity, cost, and power consumption of the hardware due the requirement of multiple antennas circuitry, as well as the need for accurate channel estimation and feedback. It is also difficult to incorporate more than one antenna at the portable unit due to the size constraint.

The time diversity technique combines channel coding with interleaving to transmit repeated information over the same channel. It can be implemented through time-hopping spread spectrum (THSS), time-division multiplexing (TDM), and automatic repeat request (ARQ). However, while this technique is effective in fast fading environments, it offers little protection against slow fading environments. Time diversity technique is more cost-efficient as it requires minimal hardware changes. However, it increases latency and overhead of transmission, reduces throughput and bandwidth efficiency, and is dependent on the coherence time of the channel.

When the multipath delay spread is large, signals transmitted over different frequencies will experience different levels of fading. This fact is exploited in frequency diversity techniques when fading is frequency-selective. Frequency diversity uses multiple carrier frequencies or bandwidths to transmit information over different frequency channels. Frequency diversity can be achieved by techniques such as frequency-hopping spread spectrum (FHSS) and orthogonal frequency-division multiplexing (OFDM). Frequency diversity technique consumes more power as all received signals must be tracked, has limited availability of spectrum resources, an increased susceptibility to inter-channel interference, and requires for synchronisation as well as coordination among different frequency channels.

2.2.2 Noise

Additive White Gaussian Noise (AWGN) is a fundamental and frequently encountered model for background noise within the domain of RF communications. It stems from the intrinsic properties of communication systems, arising from phenomena like molecular

thermal motion and the movement of discrete electric charges within electronic components. Due to its inherent physical origins, AWGN is a persistent presence and unlikely to be eradicated. AWGN finds widespread use in the modelling, analysis, and practical simulation of communication systems, particularly in the context of channel and noise characterisation. A specialised form of AWGN is band-limited white noise, characterized by a flat noise power spectrum density within a constrained operational bandwidth [9].

AWGN can be characterised as an aggregate outcome of a huge number of random events that adheres to the statistical properties of a Gaussian distribution, with its pdf given by:

$$p(v) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{v^2}{2\sigma^2}} \tag{2.9}$$

Consider an AWGN channel, with a noise vector of $n \sim \mathcal{N}(0, \sigma_z^2 \mathbf{I}_{\mathcal{N}_s})$, contains uncorrelated zero-mean complex noise with variance σ_n^2. The receiver observes the transmitted signal in the presence of additive zero-mean white Gaussian noise $v(t)$ with variance $\sigma_n^2 = \mathcal{N}_0/2$, where $\mathcal{N}_0$ is the two-sided power spectral density. The output of the channel model assuming ideal symbol synchronisation in the receiver is therefore:

$$r = se^{j\varphi} + n \tag{2.10}$$

The channel SNR is $1/\sigma_z^2$, and the ratio of average-symbol-energy-to-noise-spectral-density is given by:

$$\frac{E_s}{\mathcal{N}_0} = \frac{\mathcal{N}_s/f_s}{\sigma_n^2/f_s} = \frac{\mathcal{N}_s}{\sigma_n^2} \tag{2.11}$$

where sample rate $f_s = \mathcal{N}_s/T_{sym}$ is the rate at which the waveform samples that are transmitted serially through the channel $\mathcal{N}_s$ during the symbol period T_{sym}. The scalar phase offset $\varphi \sim \mathcal{N}(0, \sigma_\varphi^2)$ is used to model the receiver coherency, with $\sigma_\varphi = 0$ represents perfect coherency while $\sigma_\varphi = 2\pi$ represents complete noncoherency. φ is assumed to be a constant within each T_{sym}, but each symbol is allowed to vary independently.

2.3 System Modelling

One fundamental characteristic of real-world channels is their time-varying nature, leading to a delayed response to the input signals. This inherent inertia within the channel or medium gives rise to distortion effects which can impact the output of the channel. To comprehensively study such effects, the Linear Time Invariant (LTI) channel model is commonly employed. Within this model, the channel's response to any input solely relies on the Channel Impulse Response (CIR) function of the channel.

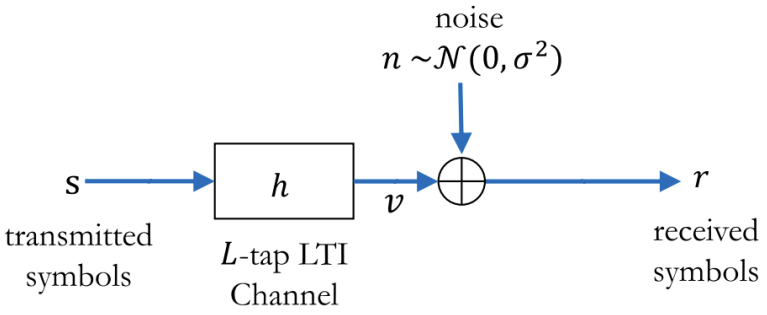

FIGURE 2.1: A block diagram of the LTI channel model

The CIR is typically represented as a finite-length sequence and is denoted by the term $h = [h_0, h_1, h_2, ..., h_{L-1}]$, where h_0 corresponds to the CIR at the symbol sampling instant 0. h_{L-1} denotes the CIR at the symbol sampling instant $(L-1)T_{sym}$. In essence, this type of channel can be effectively modelled as a Tapped Delay Line (TDL) filter, also known as a Finite Impulse Response (FIR) filter. Specifically, for the analysis of this research, only the CIR at the symbol sampling instances is considered. A flat fading channel model can be formulated with the equation:

$$r = H \otimes d + \backslash \tag{2.12}$$

where d is the vector of transmitted symbols, $\backslash$ is the AWGN vector, r is the received signal vector, and H is the CIR or channel gain matrix. y can be expressed as the linear convolution of d and H at the symbol sampling instances. In the case of a SISO or MIMO channel, the elements of H are the channel gains between transmit and receive antennas. For SISO channel, H only has a single path, whereas in MIMO channel, H has M rows and N columns to represent N transmit and M receive antennas.

SNR is the ratio of the required signal power to the noise power as given by:

$$\begin{aligned} \frac{S}{\mathcal{N}} &= \frac{\text{Required signal power}}{\text{Noise power}} \\ &= 10\ log_{10}\ \frac{\text{Required signal power}}{\text{Noise power}}\ dB \end{aligned} \tag{2.13}$$

The assessment of a system's performance can be quantified using the signal-to-noise ratio (SNR), ideally maximised for optimal results. The Shannon-Hartley Law [11, 12] provides a mathematical framework for determining the theoretical upper limit of data transmission rates, and it is directly related to the SNR of the system and can be determined by the formula:

$$\mathcal{C} = B\ log_2\ 1\ +\ \frac{S}{\mathcal{N}}\ \text{bps} \tag{2.14}$$

where $\mathcal{C}$ is the channel capacity, B is bandwidth, and $\mathcal{N}$ is the random noise power in watts while bps stands for bits per second [9].

According to the Nyquist theorem [13], the minimum bandwidth required to send a complex signal as a series of waveform samples through the propagation channel is equal to f_s. The channel bit rate is $R_b = B/T_b$, where T_b bit transmission time. The spectral efficiency of the system, measured in b/s/Hz, is

$$\gamma = R_b = \frac{f_s}{\mathcal{N}_s} \tag{2.15}$$

2.4 Conventional Radio Receiver

Receivers with homodyne (also known as zero IF or direct-conversion) architecture are widely used in mobile RF communication devices due to its simplistic circuitry, compact size, power and cost efficiency [7, 14]. In homodyne receivers, all of the potential in-band channel are frequency translated from the carrier directly to the baseband. As its intermediate frequency (IF) is translated to near-zero frequency, the desired signal is also the image of itself. This essentially eliminates the need of bulky and off-chip image reject filters, such as the IF SAW filter, required to tackle conventional image problems in receivers with heterodyne or superheterodyne architecture. Also, with the desired spectrum being directly downconverted to baseband, channel selection can be done with

a simple analogue low-pass filter, or digital signal process techniques. Moreover, the energy from the undesired channel is removed with on-chip filtering at the baseband. Therefore, homodyne receivers rely only on low pass filters and baseband amplifier to suppress out-of-channel interferers which may create large intermodulation distortion in the front-end, keeping cost and complexity low.

At the simplest conceptual level, homodyne receiver comprises of an RF front-end receiver, an analog-to-digital converter (ADC), and a digital processor [7, 15]. The RF front-end typically consist of an antenna, a band-select low pass filter, a linear RF low-noise amplifier and a down-mixer. The front-end serves to filter the frequency range to obtain only the desired range, amplify the signal and pass the signal through a power splitter prior to downconversion of the high frequency signals to near-zero baseband (can be represented in complex values with real and imaginary parts respectively). This is done in order to work with the signals at minimal power. The in-phase (I) and quadrature phase (Q) components of the baseband signal are obtained by multiplying the received signal with a reference carrier signal and a 90 degrees phase shifted copy of the reference carrier separately. It has to be perfectly synchronised in both frequency and phase to the received signal's carrier, lest there will be cross-coupling. The I and Q baseband is then passed through a low-pass filter (i.e. Butterworth, Chebyshev) to cut-off higher frequencies and separate them from the baseband spectrum before amplifying again with a variable-gain amplifier [1, 7, 14]. The ADC then samples the baseband signal at discrete time intervals which represents the centers of the modulated symbols [7] and performs quantised approximation [2, 15]. Finally, the digital processor takes the discrete amplitude values output by ADC to perform demodulation and detection with various digital signal processing technique which will be discussed in the next section.

Despite its advantages, the homodyne receiver has some plaguing problems which limit its uses. One common problem is DC offset. Since the downconverted signal extends to the zero frequency in a homodyne receiver, the DC offset voltages can corrupt the signal and saturate the baseband stage. The offsets mainly come from two sources: (i) transistors mismatch between I and Q path, (ii) LO leakage appearing at the in put of LNA, thus producing a DC voltage caused by LO self-mixing. The latter varies with time, which makes it more difficult to be removed. These offsets may also be larger than the desired signal itself. Therefore, unless removed, the SNR at the baseband will be very low. Another common problem is IQ mismatch. A homodyne receiver must

incorporate quadrature mixing that requires RF signal or LO shift by 90°. There will be phase and amplitude mismatch in I and Q path. This mismatch between I and Q path will corrupt the downconverted signal, thereby degrading the SNR. Monolithic implementation intends to improve I and Q mismatch. In practice, the amplitude mismatch is limited to -40dB while the phase mismatch is limited to 1°.

2.4.1 Quadrature Amplitude Modulation and Demodulation

Amplitude-based signalling scheme such as QAM is inherently vulnerable to performance degradation especially in a fading environment. QAM is commonly used in digital communication system such as WiFi, LMR and fibre optics [16] as it has better bandwidth efficiency. Due to these two reasons, it is suitable for theoretical studies on the performance of learning-based demodulation.

QAM signalling can be viewed as a combination of amplitude shift keying (ASK) and phase shift keying (PSK). For QAM that has an AWGN channel, when demodulated with an optimum receiver, the probability of bit error (BEP) is expressed by:

$$P_B \approx \frac{2(1-M^{-1})}{log_2 M} Q\left[\sqrt{(\frac{3log_2 M}{M^2-1})\frac{2E_b}{N_0}}\right] \tag{2.16}$$

where $Q(x)$ is the complementary error function as given in [17] equation 4 $Q(x) = \frac{2}{\sqrt{\pi}}\int_{\infty}^{x} e^{-t^2}\,dt$ and M is the number of amplitude levels in one dimension [1]. In [18], fading is not taken into consideration. Hence to obtain the BEP, it should be expressed by[1]:

$$P_{B_{\text{ray}}} = \int_{\infty}^{0} P_B(x)p(x)\,dx \tag{2.17}$$

where

$$p(x) = \frac{1}{SNR}\; e(-\frac{x}{SNR}) \quad , \quad x \geq 0 \tag{2.18}$$

In RF communications, an important requirement is to strive a balance between data throughput and signal-to-noise ratio (SNR). As modulation order M is increased, i.e. progressing from 4-QAM to 16-QAM to 64-QAM, etc. the data throughput attainable under ideal conditions increases. However, the increase in M demands a higher SNR to

[1]See [1] section 15.5.4 equation 15.38 and 15.39 for detailed explanation.

TABLE 2.2: shows the I and Q bits, the channel, output power of each phase and as well as the phase position for the corresponding value for M-QAM where M = 4. The I/Q bits determine the polarity while the C channel determines the magnitude of the signal. Assuming the data encoding is in Gray code, 00 is symbol 0, 01 is symbol 1, 10 is symbol 3, 11 is symbol 2.

I/Q	C	Output	Phase
0	0	-0.541V	$\pi/4$
0	1	-1.307V	$3\pi/4$
1	0	+0.541V	$5\pi/4$
1	1	+1.307V	$7\pi/4$

maintain comparable data throughput and low error rates as achieved in ideal scenarios. Conventional approaches to optimising data speed involve adjusting M and employing error correction techniques to maintain the desired error rate. However, the aim is to achieve error rates as close to ideal conditions as possible, enabling satisfactory data throughput even in low SNR situations.

Analysing the QAM constellation diagrams for various M reveals that as the M increases, the distance between points on the constellation decreases. Consequently, small amounts of noise can lead to significant issues. As noise levels increase due to weak signal strengths, the area covered by a point on the constellation expands. If this area becomes too large, the receiver becomes unable to accurately determine the intended position of the transmitted signal on the constellation, resulting in errors. Moreover, higher M in QAM signals introduces greater amplitude variation in the transmitted signal. This aspect necessitates the use of linear amplifiers that are less efficient than their saturation-operating counterparts for transmitter RF amplification. Furthermore, the increase in amplitude variation reduces efficiency, an important consideration for mobile equipment battery efficiency and base station power efficiency.

In the context of digital modulation signals, the transmission of information occurs through a series of uniformly spaced samples, each occupying a fixed time slot. Within each sample, the carrier wave maintains a constant amplitude and phase, constrained to a finite set of values. These samples serve as encodings for a finite number of symbols. In M-ary modulation schemes, multiple bits are combined to form symbols. During each symbol period T_s, one of the possible signals $S_1(t), S_2(t), ..., Sm(t)$ is transmitted. In this case, $M = 2^k$ and $k = log_2 M$, where k is an integer indicating the number of bits per symbol and the total number of potential signals.

The capacity bound for bandlimited Gaussian channel based on Shannon's Channel Capacity Theorem is shown in Figure 2.2. The bandwidth-efficiency plane illustrates the performance trade-offs between different modulation types between linear and orthogonal modulates. The upper right region of Figure 2.2 is known as the bandwidth-limited region, where M-QAM is located, has spectral efficiency as its key performance metric. Due to the use of linear transmitters and coherent demodulators, transceiver cost and power consumption is higher than orthogonal modulations like M-FSK. Channels often exhibit frequency-selective and dynamic characteristics, requiring forward error correction, integration of pilot symbols and interleaving across subcarriers to ensure satisfactory data throughput.

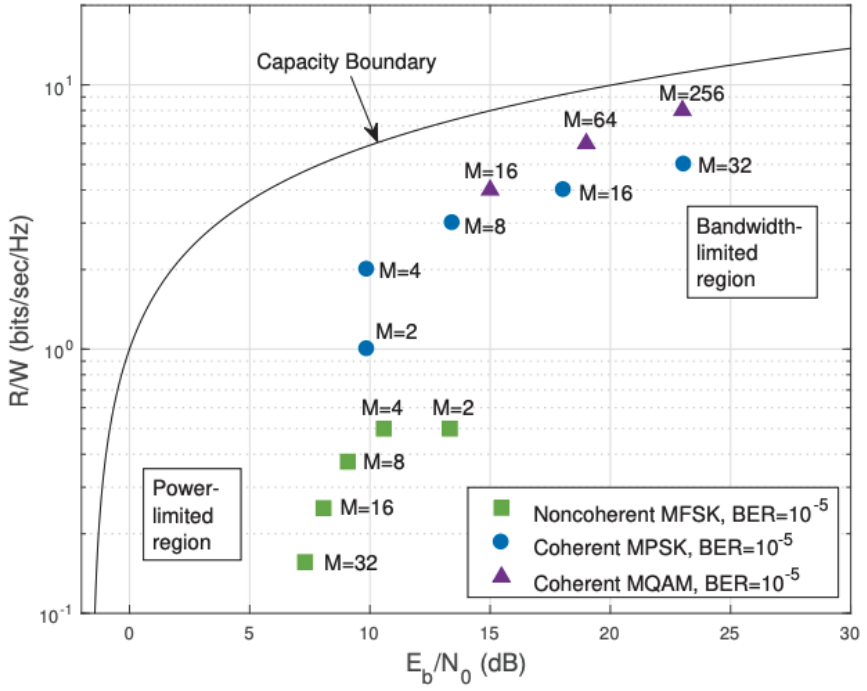

FIGURE 2.2: Illustration of the bandwidth-efficiency plane.

2.4.2 Coherent Demodulator

In RF communication, a receiver's primary task is discerning whether an observed pulse represents the desired signal that carries valid information or is merely noise occurring at specific time instances. Some strategies have be devised to tackle this problem. For a digital signal represented by symbols $s_1(t), s_2(t), ..., s_l(t)$, with a span of T_{sym}, the symbols should be orthogonal with no cross-correlation between them. This suggests a collection of correlators where each one consists of a multiplier followed by an integrator

would make an ideal receiver. The received signal is fed into one input of each multiplier, while a waveform corresponding to a unique symbol is provided to the other input. In a noise-free environment, a singular correlator would yield an output. In environment with presence of noise, multiple correlators might respond, but the signal associated with the true received symbol is expected to produce the largest output. Such mechanism aligns with the maximum likelihood principle, which is recognised as optimal for AWGN, especially when symbols have equal probability. A correlator can also be viewed as a coherent demodulator since the integrator serves as a low-pass filter. Thus, when the suitable waveform acts as its local oscillator, a coherent demodulator can operates as a correlation detector [19]. Locally generated signal $s_i(t)$ is multiplied and synchronised with $r(t)$, denoted as:

$$z_i(t) = \int_0^t r(t)s_i(t)\ dt \tag{2.19}$$

Another approach is to apply matched filter that will enhance the signal while reducing the noise voltage instead of performing correlation between each samples. Matched filters are linear filters applied to known signals with random parameters (such as amplitude and arrival time) to maximise the SNR. It has a transfer function that is specific to the desired input signal to be received. It has an impulse response $h(t) = s(T_0 - t)$ that is a delayed version of the time-reversed copy of $s(t)$. The input signal undergo convolution to obtain the output denoted as:

$$z_i(t) = r(t) * h(t) \tag{2.20}$$

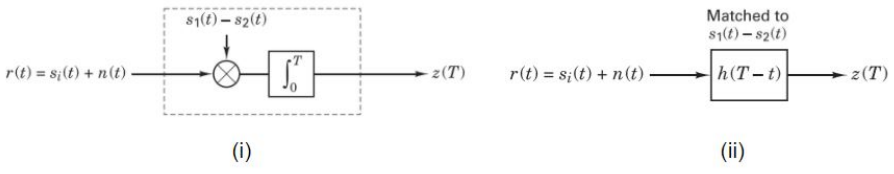

FIGURE 2.3: illustrates the block diagram where symbol to bits conversion is performed with (i) correlator and (ii) matched filter. Correlator and matched filter gives the same output despite the difference in mathematical operation hence can be used synonymously [1]

.

An optimum receiver can be designed with either detectors: correlator or matched filter following a decision-making logic with either a Maximum Likelihood Estimation (MLE) or Maximum a Posteriori (MAP) decision rule.

2.5 Machine Learning-based Demodulation and Detection

The conventional demodulation techniques described previously are predicated on the assumption of Gaussian distributions. In contrast, data-driven approaches such as machine learning (ML) focus less on assumptions and rely more on the inherent patterns within a given dataset to make predictions regarding symbol classes, achieving comparable performance to conventional methods. ML-based demodulation methods can be divided into two components: (i) a feature extractor, which learns features from the provided signal data, and (ii) a classifier, responsible for making symbol decisions.

Chen et al. [20] proposed the use of support vector machines (SVM), employing both a simple linear kernel and a radial basis function (RBF) kernel. Training these models on BPSK data corrupted by Gaussian white noise yielded a BER performance improvement of approximately 1.8 dB compared to threshold-based decision-making, even under challenging conditions of -7 dB SNR. The linear SVM also exhibited superior performance to the SVM-RBF at lower complexity. Wang et al. [21] introduced Adaptive Boosting model to classify PSK and QAM signals with varying modulation orders. Notably, they utilised real signal data collected from radio equipment, which is subject to impairments such as noise, multipath channels, non-linear distortion, interference, and frequency selective fading. Their Adaptive Boosting classifier outperformed other baselines. Siyari et al. [22] also employed SVM implemented on a software-defined radio to demodulate QAM and ASK signals in the presence of carrier-frequency offset (CFO) and Gaussian noise. Their SVM-based demodulator learned to approximate optimal demodulation boundaries and achieved a gain of up to 3 dB compared to conventional demodulation schemes.

A key challenge for ML models lies in maintaining a sufficiently low complexity to operate in near real-time while effectively discriminating between waveforms belonging to different signal categories. Since ML models for classification typically consist of distinct feature extractors and classifiers, this motivates interest in DL methods.

2.6 Deep Learning-based Demodulation and Symbol Detection

2.6.1 Motivation for Deep Learning-based Demodulation

With the recent surge of DL applications in the physical layer of RF communication systems such as automated modulation classification [23–25], RF interference detection and cancellation [26–29], a number of studies have demonstrated the feasibility of training neural networks to demodulate signals transmitted through noisy and dispersive channels. While the prominent advantage of DL over traditional ML lies in its automated feature extraction capability, there are several compelling reasons to favour DL over ML specifically for RF signal demodulation:

1. Costly nature of hand-crafted feature engineering: DL offers a significant advantage by automatically extracting high-level features from data with intricate structures and complex internal correlations, which may not be readily apparent to human experts. This eliminates the need for manual design of the learning process and mitigates the requirement for domain-specific feature expertise. This advantage becomes particularly pronounced in RF communication systems, where RF signals often originate from heterogeneous sources, are commonly corrupted by noise, and exhibit non-trivial spatial or temporal patterns.

2. Better scalability and exploitation of diverse data: Training traditional ML algorithms, such as Support Vector Machines (SVM), can be computationally infeasible under scenarios involving big data due to the requirement of storing all data in memory. Moreover, ML algorithms often demonstrate limited performance improvement with increasing data volume, reaching a plateau quickly. Conversely, DL models trained using Stochastic Gradient Descent (SGD) only require subsets of data during each training step. Furthermore, training DL models with diverse data enhances their generalisation capability.

3. Multi-objective fulfillment with a single DL model: DL models can be trained to accomplish multiple objectives, ranging from compressive representation, feature approximation to data interpolation, without requiring separate, independent

models for different tasks, as is the case with ML. This significantly lower computational and memory costs. This characteristic is particularly valuable in RF communication scenarios, where devices typically operate on low-power single-board computers.

4. Proven maturity and accuracy in computer vision: DL has already achieved remarkable success in computer vision and has repeatedly demonstrated good accuracy in tasks for which the models are trained. This track record establishes DL as a well-established field with a strong foundation.

5. Potential to utilise of unlabeled data for unsupervised learning: DL methods, such as Autoencoders (AE), Generative Adversarial Networks (GAN), and Restricted Boltzmann Machines (RBM), can be designed to leverage unlabeled data for unsupervised learning, enabling tasks such as data distribution approximation, unsupervised or semi-supervised learning, and one-shot or zero-shot learning, among others.

The application of DL in the form of multi-layer perceptrons (MLP), has been demonstrated as early as 1990s [30–32] can achieve comparable performance to traditional likelihood ratio receivers for various modulation types, including QAM [31, 32]. The objective is to investigate the potential of DL that are nonlinear in encompassing multiple demodulator functions (e.g. output pulse shapes preservation and narrowband filtering). It is proven with empirical findings that a DL model, in particular, MLPs trained using backpropagation can effectively approximate the discriminant functions optimised by the Bayes theorem for both binary and multi-class recognition tasks. The achieved performance level is comparable to optimum matched filters in the presence of AWGN. Notably, this finding extends beyond MLPs and encompass any networks architectures that minimise on a given error. However, the results are not entirely consistent across all cases, as some researchers observed improvements in network performance through training with noise, while others found that noise has adverse effects, either degrading the final results or causing distractions in the network's learning process of underlying signal features. Later research has focused on improving the performance of demodulator for signal conditions that is outside of traditional optimum detectors' assumptions, with more sophisticated DL architectures. In many recent literature, the architecture of DL

used for the purpose of demodulation revolves around direct signal-to-symbol demodulation with a mixture of architecture in neural networks, convolution neural networks, and recurrent neural networks.

2.6.2 Direct Signal-to-Symbol Demodulation

One method to approach the signal demodulation task is to treat it as a classification task. Probabilistic neural networks are typically used for classification problems. When presented with an input, the first layer computes distances of an input vector to the training input vectors. The output is a vector that indicates the correlation between the input to the training input. The consecutive hidden layers sum these contributions for each class of inputs to produce a vector of probabilities as output. Finally, the transfer function picks the maximum among these probabilities hence classifying the outcome [33]. Several popular DL architectures are explored in the following sections.

2.6.2.1 Neural Networks

Neural networks (NN), deep neural networks (DNN), or multi-layer perceptron (MLP), have long been the research interest in demodulation for their ability to approximate any function. NN are considered to be discriminative models that represent a posterior probabilitiy $p(C|x)$ where x being the input vector and C being the corresponding classes of x. Using the Bayes theorem together with marginalisation of the nuisance variables, a discriminative model can be derived. NN can also adapt to data that is not linearly separable with the addition of a non-linear activation function in each neuron. NN processes an input tensor to a resulting output tensor through a model that consists of layers of neurons interconnected through weights. Each neuron includes a non-linear activation function that determines the output of the neuron, denoted as:

$$z = f(xw + b) \tag{2.21}$$

The output of each neuron z is computed by a propagation function f of its input tensor x multiplied by its weight tensor w adding the bias term b. f can be either linear or non-linear activation (i.e. sigmoid, hyperbolic tangent, rectifier linear unit and etc). Both w

and b will be updated and learned over iterations, but need to be first initialised. The forward propagation, also sometimes known as feed-forwarding, processes information and generate output from the input given.

Backpropagation is a pivotal process in training neural networks, involving the computation of gradients pertaining to the network's hyperparameters [34]. It is applied iteratively to systematically propagate gradients across all network nodes, commencing at the output layer, where the network generates its predictions, and proceeding down to the input layer, which receives input data. After these gradients have been computed in relation to the model's weights, error is computed by comparing the generated output with the true target values. Parameter values are updated according to the error. This process continues to refine the model's parameters and producing progressively improved output predictions.

The idea of neural demodulation first came about in 1991 on [30] which talks heavily about narrowband radio propagation and statistical modelling of path loss and time dispersion. The suggestion of potential use of NN is futuristic as both DL field and computing technology is still at an infancy stage. [31] presented a performance analysis of single hidden layer NN demodulation on 16-QAM, BPSK and QPSK baseband signal data impaired with AWGN versus coherent demodulation that shown the result of NN to have an overall gain of 5.3dB over coherent demodulator. Using the method, [32] supports the work from [31] that NN is more robust than coherent demodulator that risks waveform distortion caused by phase error. [35–37] also used single hidden layer NN to demodulate PSK and QAM signals with AWGN. NN is shown to outperform the correlation receiver with gain of about 2dB and is close to the theoretical BER. In one of the most influential DL demodulation works, [23] modelled an end-to-end communications system over an AWGN channel represented as an autoencoder, where the transmitter is a NN-based encoder while the receiver is an NN-based decoder. The receiver is being trained to demodulate noisy and faded M-PSK and M-QAM signals. While the plain autoencoder's performance falls behind that of a differential BPSK with MLE and Hamming code, their other proposed autoencoder with a parametric transform layer outperforms the MLE baseline. [38] found NN to bring consistent performance improvement for underwater acoustic (UWA) OFDM in comparison to the conventional method. [39] demodulate noisy quadrature amplitude shift keying (ASK), quadrature FSK and quadrature phase shift keying (PSK) signals faster by using four parallel NN.

[40] investigated AWGN and channel interference immunity of amplitude noise shift keying (ANSK) signal demodulation with NN found that NN is able to achieve competitive results as conventional demodulation methods. [41] implemented an NN demodulator for BFSK and BASK on practical low-power microcontroller that is able to achieve stable communication within the distance of 10m at a data transfer rate of 5Kbps with a power consumption of less than 0.1 W. However, the BER is above 10^{-1} across all SNR due to being underwater (water conducts electricity that can reflect radio waves), and not having further performance improvement beyond 0dB as opposed to over-the-air RF communication. binary FSK is also observed to perform better than BASK. In general, NN models have shown BER performance that is very close to the ideal case in AWGN and fading conditions, even with a single hidden layer NN is found to be able to outperform the conventional method. In [42], NN is trained demodulate BPSK signal under AWGN and fading channel along with hardware imperfections such as frequency and phase offsets. Their NN outperformed the conventional coherent demodulator and other DL demodulator architectures.

An alternative to NN, the deep belief network (DBN), gained interest by [43, 44], where they proposed and demontrated that signal equaliser is not required to demodulate noisy and faded BPSK signals in short range multi-channel. Similarly, [45] also proposed a multi-layer DBN to demodulate noisy MSK signals with slight performance improvement over the traditional coherent demodulation at specific SNR.

2.6.2.2 Convolutional Neural Networks

CNN is initially intended as a form of regularisation to prevent NN from overfitting due to its fully connected nature. It is later found and proven to be effective for most digital signal processing tasks as it exploits the property where compositional hierarchies exist in many natural signals. For instance, hierarchies exist in speech audio from sounds to phones, phonemes, syllables, words and sentences. CNN take advantage of such hierarchical pattern in data and assemble more complex features using smaller and simpler features. CNN assumes that the data has shift-invariance properties as well as sparse representation, with shared weights it thereby reduces the number of trainable network parameters required.

There are many architecture variants for CNN, such as depth-based, multipath-based, attention-based and etc [46]. Still, the CNN architecture that is typically referred to is spatial exploitation-based. As the most popular choice of DL for pattern and object recognition, CNN has been been widely applied to a vast majority of digital data processing. The mathematical operation, convolution, however, differs from the convolution operation defined in DSP [47]. CNN uses the same convolution operation to perform feature extractions repeatedly then the results are fed into a fully connected neural network to perform classification of the input. CNN has three important key concept: (i) sparse interaction; (ii) weight sharing; (iii) equivariant representations. Due to this CNN has high affine invariance and considerably less number of model parameters needed [34].

$$Y(i,j) = \sum_{m} \sum_{n} X(i+m, j+n) \cdot K(m,n) \tag{2.22}$$

where Y is the output feature map, X is the input data, K is the convolutional kernel (sometimes referred to as filter), while i and j represent the spatial location of the output feature map. Every layer in CNN can be viewed as the marginalising of one or more nuisance variables, such as the peak, orientation, location or magnitude of symbols being classified to estimate $p(C|x)$. Similar probabilistic approaches make the basis of traditional demodulator designs that utilise MLE and MAP. Beyond the valuable insights into the architecture and activation functions of CNN for signal processing, this perspective establishes a connection with communication theory and learning-based demodulators.

Hence, [48] proposed deep CNN to demodulate noisy BASK, BFSK and BPSK signals. Lin et al. has shown that CNN can be a versatile demodulator as it can either be trained to demodulate a sampling sequence consisting of a single symbol or a long sequence consisting of multiple symbols. Their experiment suggests demodulating a single symbol is more advantageous for a CNN model. Similarly, Mohammad et al. [49] implemented a demodulator with a deep CNN for BFSK signals with AWGN and frequency-selective fading in their work. They managed to obtain an impressive result of a 10dB learning-based gain as compared to various methods such as correlation and a 2 hidden layer NN demodulator. CNN is implemented again in [50] to M-FSK signal with AWGN but the performance does not come close to the theoretical performance of a non-coherent

demodulator baseline. In [51] the authors showed that CNN outperforms typical non-coherent receivers when under AWGN-only conditions, outperforms both coherent and non-coherent demodulation methods when under AWGN-time offset conditions and coherent receivers when under AWGN-carrier frequency offset conditions for FSK signal. In general, the performance of CNN varies across different studies, whereby CNN performs better when demodulating signals with various channels and signal impairments when compared to coherent and non-coherent demodulators than an AWGN channel only.

2.6.2.3 Recurrent Neural Network

For tasks involving sequential inputs such as signal processing, RNN is often the model of choice due to its memorising capability of past inputs. RNN's memory works by maintaining a state vector that implicitly contains information of all the past elements. The single most important element of RNN is the backpropagation of error through time which is different from general feed-forward networks, as backpropagation takes place at each step or each point in time and the total gradient is the summation of gradients at each step. During backpropagation, information flows in reverse direction as matrix multiplication happens backward where the weight matrix is transposed and come to the front. As for the state vectors, the outer product remains the same as state vectors are only transposed without swapping. Also, the dimensions of matrix remain consistent throughout the calculation of gradients. In this way, error gets distributed backward while the fidelity of connections is maintained. RNN can be modelled in five different process sequences: (i) one-to-one where one unit of recurrent cell receives one input that is mapped to one single output (i.e. sample-to-symbol); (ii) one-to-many where several units of recurrent cell receive one input that will be backpropagate through time to produce a sequence of output (i.e. sample-to-bitstream); (iii) many-to-one where several units of recurrent cell receive a finite timestep sequence of input that will be backpropagate through time to produce one single output (i.e. samples-to-symbol); (iv) many-to-many where several units of recurrent cell receive a finite timestep sequence of input that will be backpropagate through time to produce a sequence of output (i.e. samples-to-bitstream classification or samples-to-audio similar to natural language processing machine translation).

Vanilla RNN tends to suffer from either exploding or vanishing gradients due to the the temporal evolution of the backpropagated error is dependent on the size of the weights. Therefore, other variations of RNNs such as the long short-term memory (LSTM) [52] and gated recurrent unit (GRU) [53] which retains the advantage of RNN without the exploding/vanishing gradient problem are more commonly applied. The common LSTM is expressed as:

$$\begin{aligned} i_t &= \sigma_i(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\ f_t &= \sigma_f(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\ c_t &= f_t c_{t_1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\ h_t &= o_t tanh(c_t) \end{aligned} \tag{2.23}$$

where, σ is the logistic sigmoid function, i, f, o and c are its input gate, forget gate, output gate and cell activation vector cells at time t respectively. x_t is the input feature vector, h_t is the hidden output vector. b_i, b_f, b_o are the bias terms respectively while $W_{hi}, W_{hf}, W_{ho}, W_{xi}, W_{xf}, W_{xo}$ are the weight matrices connecting the different inputs and gates with the memory cells. GRU, on the other hand, is expressed as:

$$\begin{aligned} i_t &= \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \\ f_t &= \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \\ o_t &= \phi_h(W_h x_t + U_h r_t \odot h_{t_1} + b_h) \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t \end{aligned} \tag{2.24}$$

where x_t is the input feature vector, h_t is the output vector, $\hat{h}_t$ is the candidate activation vector, z_t is the update gate vector and finally r_t is the reset gate vector. b_z, b_r and b_h are the bias terms respectively while W_z, W_r, W_h are the weight matrices connecting the different inputs and gates with the memory cells [53]. GRU typically uses the sigmoid σ_g activation function to forget or remember the information by outputting either 0 or 1, and the hyperbolic tangent ϕ_h activation function to overcome the vanishing gradient problem as its second derivative can sustain for a long range before reaching zero. However, the use of an alternative activation function to sigmoid is also possible provided that $\sigma_g(x) \in [0, 1]$.

Before the popularisation of LSTMs and GRUs, low complexity networks such as Elman network and time-delay NN are proposed to demodulate BFSK signal with performance similar to coherent demodulator [54, 55]. While a conventional receiver typically requires a band-pass filter to filter out-of-band noise and pulse waveform regeneration, it is not necessary to perform such operations on the data prior to demodulation with an RNN. A sequence-to-sequence (Seq2seq) RNN model is proposed in [56] connected in an encoder-decoder structure to perform symbol prediction and pulse-shaped symbol recovery on binary phase shift keying (BPSK) and QPSK signals with AWGN channel. While Seq2seq achieved better results for BPSK signals in comparison to the matched filter (coherent) demodulator, the work does not have a convincing result as the Seq2seq has visibly poorer performance for QPSK signal than the matched filter counterpart. There is no conclusive answer regarding the possible causes of the performance difference. An end-to-end baseband signal to audio output RNN demodulator with stacked bi-directional LSTM for frequency modulated (FM) signals is proposed in [57] for the first time. Their proposed method yields results better than the digital phase-locked loop-based (non-coherent) demodulator in [58]. Daldal et al. also applied bi-directional LSTM to ASK, FSK, and PSK modulation which sees a significant improvement when compared to SVM [59]. Finally, a RNN model with LSTM to demodulate orthogonal frequency division multiplexing-aided differential chaos shift keying (OFDM-DCSK) signals over AWGN and fading channels is proposed by [60]. Their model outperforms the benchmark for OFDM-DCSK systems only at higher E_b/N_o over fading channels.

2.6.2.4 Other Hybrid Architectures

There are also works that implemented hybrid architecture to obtain the advantage of each algorithm. [61] proposed a Twice-Training Network demodulator that does not require channel equaliser with a significant improvement in BER when compared to MLE detection. Inspired by [57], Wu [18] compared 3 major models - CNN (similar architecture as [48]), RNN with bi-directional recurrent layer, and a combined CNN-RNN with GRU to perform symbol detection. BFSK, QPSK and 16-QAM modulated signal have AWGN and only BFSK have Rayleigh fading channel. Their proposed solutions for faded signals perform better than non-coherent demodulator while coming close to coherent demodulator. Xie et al. proposed a CNN-RNN with LSTM DL model for non-orthogonal multiple access joint signal detection that simultaneously fulfils the

function of channel estimation, equalisation, and demodulation of noisy and faded signal [62]. From the literature, it can be observed that highly complex or deep architecture is not truly necessary. Simplistic and naive architectures have been shown to be just as competitive when constructed for the correct use case. It is also observed in the results across several previous works that linear modulation suffers in BER performance more than nonlinear modulation, such as Frequency Shift Keying (FSK) modulation under conditions where the signal's amplitude and/or phase gets distorted. In [63], they proposed a hybrid of CNN-RNN with GRU for UWA communication which has a better BER performance when compared to other models such as CNN, RNN as well as MLE.

2.6.3 Two-steps Demodulation

In many previous works, the signal demodulation task is treated as a classification task using different architectures such as NN that have been implemented in [31, 32, 35–39, 41, 43–45], CNN [18, 48, 49, 51], RNN [56, 60] and hybrid CNN-RNN [18, 62, 63], where most of the works have competitive results to the coherent demodulator in noisy and faded conditions. The models learn a set of features with definitive dimensions and map the learned features back to the correct class, where each symbol in a modulated signal is a separate class. The difficulty of learning to detect the symbols of an impaired modulated signal lies in the nonlinear structure of the feature space. Although DL models are able to provide nonlinear mapping through learning spatial or temporal features and perform universal approximation, remapping the learned features back to their respective symbols with many class anomalies and overlapping class clusters caused by randomised perturbation in the feature space can be a challenge to obtain very high accuracy that is required in the telecommunication standards. Another issue to consider is that high-rate digital processing often requires very high resolution analog-to-digital converter (ADC) that is power, cost and space (bulky chip) inefficient. A spectrally efficient modulation such as 16 or 64-QAM would require ADC with a resolution of ≥ 10 samples at high sampling rates of 10 Giga samples-per-second [64]. To minimise cost, some RF communication devices employ a direct conversion architecture whereby the front-end downconverts the received signal to near-zero frequency baseband signal I(t) and Q(t) before sampling. However, the baseband signals can be corrupted by direct current (DC) offset, I/Q mismatch, local oscillator leakage, and other distortions due to hardware imperfection [65].

To resolve the issues, a different approach is proposed with a DL-based denoising step to reconstruct a clean signal to ease symbol decision-making.

Wang et al. [21] proposed a two-steps approach that uses DBN to extract information from received signal and SVM to classify the PSK and QAM signals of various modulation orders. The DBN-SVM method, however, turns out to have less ideal accuracy when compared to Adaptive Boosting classifier. [66] also proposed CNN as channel equaliser and NN as demodulator for UWA communication. Their proposed model brings slight BER improvement over the FC-DNN model but bears significant improvement over conventional UWA-OFDM with least square (LS) channel equaliser. The authors also noted that the FC-DNN model has a non-convex optimisation and gradient disappearance problems hence less robust to complex scenarios. In contrast, CNN has a strong feature extraction ability, ensuring more accurate signal recovery. Another method implemented with an autoencoder with long short-term memory (LSTM) to achieve signal noise reduction and deep residual network (ResNet) to perform symbol detection on the denoised signal is proposed by [67]. The LSTM autoencoder model managed denoise the signal from 11.8 dB to 13.66 dB SNR while the ResNet managed to reduce BER from 10^{-8} to 10^{-11} for signal with white noise and constant phase noise. However, achieving such results requires a computationally expensive LSTM autoencoder and large size ResNet with 50 up to 152 layers which may not be suitable for practical applications as higher computational complexity is equivalent to a lower data rate. Therefore, the proposed model in this research is carefully designed to provide the best performance with the lowest complexity and size.

Table 2.3 summarises the DL-based demodulation methods from the earliest concept to the present SOTAs. The table specifies (i) paper citation arranged by year of publication in ascending order; (ii) model architecture used in the paper; (iii) signal modulation techniques; (iv) channel impairment; (v) performance measure used; (vi) benchmarks used; (vii) if the said demodulation is signal-to-symbol or a two-steps demodulation; (viii) any hardware implementation; (viiii) if it is using simulated or real dataset; and finally (x) if the proposed DL-based demodulator performs better than the conventional methods as benchmark. It is observed that NN being the most popular method, followed by RNN, while both CNN and RNN being the most considered methods in the past three years. PSK is the most commonly studied modulation. BER is the most commonly used measure of performance. Also, an interesting observation is that most

prior works demonstrated improvement over conventional methods such as the coherent demodulator, non-coherent demodulator, least square (LS), MLE and minimum mean square error (MMSE), but not when compared to theoretical BER performance which is usually the ideal case.

TABLE 2.3: shows a quick comparison of all DL-based demodulation for wireless communication applications where the papers are arranged in chronological order. Blank space indicates the item is unspecified or not applicable to the paper.

Paper	[31]	[32]	[35]	[54]	[55]	[37]	[36]	[61]	[48]	[43]	[49]	[56]	[18]	[21]	[38]	[44]	[45]	[57]	[41]	[50]	[39]	[59]	[60]	[51]	[62]	[66]	[67]	[63]
Model																												
NN	x	x	x			x	x	x		x				x	x	x	x		x		x					x		
CNN									x		x		x							x				x	x	x	x	x
RNN				x	x							x	x					x				x	x		x		x	x
AE										x						x											x	
Modulation																												
AM/ASK									x										x		x	x						
FM/FSK				x	x						x		x					x	x	x	x	x		x				
PSK	x					x		x	x	x		x	x	x	x	x	x				x	x	x		x	x	x	
QAM	x	x	x				x						x	x													x	x
Channel																												
Noise	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Fading						x		x			x		x			x							x		x	x		x
Measure																												
BER	x			x	x		x		x	x	x		x		x	x	x		x	x			x	x		x		x
SER			x					x																	x			
Accuracy											x	x		x							x						x	
Benchmark																												
Theory			x				x	x		x										x			x	x				
Coherent	x			x	x		x		x		x	x	x				x		x					x				
Non-coherent				x	x								x											x				
LS/MLE/MMSE										x				x	x			x							x	x		x
Direct Demod.	x	x	x	x	x	x	x	x	x	x	x	x	x		x	x	x	x	x	x	x	x	x	x	x			x
HW Impl.																			x	x							x	
Sim. Dataset	x	x	x	x	x	x	x	x	x	x	x	x	x		x	x	x	x		x	x	x	x	x	x	x		x
DLD > Baseline	x			x	x		x		x		x	x	x	x	x		x	x	x						x	x		x

2.7 Summary

The availability of significant computational power has enabled the development of highly precise and detailed simulation models for RF propagation channels and RF systems. These advanced simulation capabilities have led to the ease of generation of vast amounts of data for the purpose of RF system design validation. However, the abundance of data presents an opportunity beyond mere validation; it can also be leveraged for training purposes. Once an appropriate DL demodulator architecture design is identified, subsequent efforts can be concentrated on defining performance objectives and selecting suitable training data for the model.

In most recent works, DL-based demodulation and detection have gained traction, particularly in the context of direct signal-to-symbol approach. Various classifiers such as MLP, CNN, RNN, and hybrid architectures, have been employed in these efforts. On the other hand, some recent studies have adopted a two-steps approach, where signal transformation, feature extraction, and feature classification are treated as distinct functions to be modelled independently. In this approach, modular learning process is employed, where the first block extracts information from the received signal or reconstructs it, and the subsequent block classifies the output from the former block. Both the direct signal-to-symbol approach and the two-steps approach have their respective advantages and limitations to be taken in considerations for practical applications.

While previous research has primarily focused on studying demodulation and detection in the presence of AWGN, some studies have also explored non-stationary channel conditions, such as time-varying Rayleigh fading. Research studies on different approaches to signal demodulation, be it signal-to-symbol demodulation or two-steps demodulation, both have yielded promising results. The results often surpass the performance of conventional receivers or achieve comparable outcomes without requiring specific pilot sequences in the transmitted signal. The performance of DL-based demodulators has served as a great motivation to pursue further research in this direction.

Chapter 3

Deep Learning-based Demodulation: Direct Signal-to-Symbol Approach

This section contains the details of the proposed DL method to demodulate signals by treating the demodulation as a classification task. Inspired by the findings from literature review, a novel architecture namely the recurrent-attention networks (RE-ATT) model is designed to performs symbol detection on raw, time series signal data is designed. The idea and motivation behind the design is explained, tested with results tabulated.

3.1 Recurrent-Attention Networks Demodulator Design Considerations

In signal processing where sequential and time series data dominate the scene, it can be very useful to have memory component that retains information about previous inputs as the order of samples in the signal input is important and there are temporal dependencies between each sample that convey certain information such as the modulation, attenuation or amplification of signal in a specific segment and et cetera. Memory in a DL model can be achieved by constructing a recurrent neural network layer. GRU [53] is a suitable candidate to construct the recurrent layer for its ability to capture long-term dependencies due to the existence of hidden states and mechanisms to determine what

to be updated in the hidden state and what to be forgotten. Unlike vanilla RNN, gating mechanisms sufficiently prevented vanishing gradients as there will not be repeated application of recurrent weight matrix, and the product will not be directly passed through from one node to the next node [68]. The gating mechanism also prevents the repeated direct application of nonlinearity, especially for sigmoid activation whose derivative is near 0. To further reinforce the model by having information on the position of each element, bidirectional GRU can be applied to allow contextualisation of input data with respect to past and future.

The outcome of a bidirectional GRU is a complex representation and much longer than the original input sequence. If directly fed to a multi-layer perceptron, either a deep network or a high number of neurons would be necessary to transform the feature space to be separable. Hence, an attention mechanism [69] can be used to summarise the feature learned from the recurrent layer. Attention mechanisms allow the model to focus on different parts of the input sequence at different times, which can be particularly useful in signal processing tasks where certain parts of the input signal may be more important than others. In addition to improving the accuracy of signal processing models, attention mechanisms can also help to reduce the computational cost of processing large input sequences. By selectively focusing on important parts of the input sequence, attention mechanisms can reduce the amount of unnecessary computation required to process the entire sequence.

Due to these considerations, a novel architecture RE-ATT is designed with the aim to fit polynomial functions for a time series classification task. The proposed architecture is loosely based on the idea of a typical transformer introduced in [70] that has found success in a wide range of sequence-to-sequence problems such as machine translation, generation of text, music and other medias. Given the application of the proposed solution is for the application on resource-limited low-power devices, the RE-ATT architecture leverages on GRU acting as an encoder to preserve the positional information inherent in signal data that is crucial in signal processing. While this adaptation sacrifices the parallelism seen in transformer models due to the inherently sequential nature of recurrent processing, it empowers RE-ATT to excel in tasks demanding precise temporal information. The attention mechanism that follows helps to capture contextual information within a sequence.

3.1.1 Recurrent-Attention Networks Architecture

The proposed RE-ATT has four main layers that forms the model itself: (i) a forward pass GRU, (ii) a backward pass GRU, (iii) an attention mechanism, and (iv) the output layer where the details are provided at Algorithm 1 along with the visualisation of the architecture in Figure 3.3. The model comprise of the following two important components:

I. Gated Recurrent Unit (GRU): Although similar to LSTM, GRU is more robust than LSTM due to the absence of an output gate, the use of leaky integration to balance between the previous, new memory content and current memory content, as well as being fully exposed at every timestep [71]. GRU has achieved notable results on various sequence-to-sequence learning problems such as text-to-speech mapping, natural language translation, and music genre classification [72]. The update gate captures long-term dependencies in the input. The element-wise logistic sigmoid activation will either let no or complete flow of information through the update and reset gate. Hyperbolic tangent activation is used to overcome the vanishing gradient problem as its second derivative can sustain for a long-range before reaching zero. However, GRU can still suffer from gradient explosion due to their nonlinear dynamics that can drastically change when the parameters cross certain bifurcation points [72].

The fully gated GRU [53] layer acts as a feature extracting module whereby it takes a sequence of time series data as input. The order of the sequence that can be understood as the alignment and constellation of the signal will still be conserved with GRU. Temporal features vector consisting of high-order sequential regularity of the signal data outputted by the forward and backward GRU in the recurrent network are concatenated and passed to the attention mechanism [69] to be transformed to a context vector.

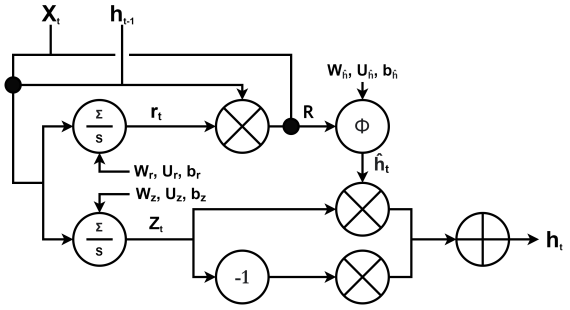

FIGURE 3.1: Illustration of a full GRU where R is the reset gate, Z is the update gate, $\hat{h}$ is the candidate activation and h is the output state. The 3 parameters learned by GRU are the weight W, recurrent weight U and bias b. S is the sigmoid activation function while ϕ is the hyperbolic tangent activation.

A fully gated recurrent unit as shown in Figure 3.1 has the equation defined at Equation 2.24. The parameters are trained, calculated and updated by the backpropagation of error. This process is performed to obtain the output of the current timestep, can be described as:

$$o_t = f(U_{x_t} + W\, f(U_{x_{t-1}} + W\, f(U_{x_{t-2}} + W\, f(U_{x_{t-3}+\ldots})))) \tag{3.1}$$

Both W and b will be updated and learned over iterations, but will be first initialised randomly. Also, the same cell is applied to all x_T so that the weights are shared across the samples in a symbol.

GRU models the distribution of X_j by predicting the probability of x_{t+1} given $\hat{h}_t$ and x_t. The eigenvalues are inputted in chronological order during processing of an input sequence. The final output is only returned when the processing of last element in X_j is finished. Bi-directional implementation of GRU enables X_j to be processed in both forward $\overrightarrow{f}$ (past states $\overrightarrow{h_1}, ..., \overrightarrow{h_T}$) and backward $\overleftarrow{f}$ (future states $\overleftarrow{h_1}, ..., \overleftarrow{h_T}$) in the same timestep, as seen in Figure 3.3 [53]. h_t of both layers are concatenated into the final vector output H and passed to the next layer. H contains annotations of each sample $[h_1, h_2, ...h_T]$ by summarizing information from $\overrightarrow{f}$ and $\overleftarrow{f}$, therefore incorporating the contextual information.

However, not every feature in the context vector is relevant or helpful for classification because noise and fading are randomised perturbations in the signal. These randomised perturbations cause low-class separability; thus, it is important to have an attention mechanism to weigh each feature by a score to emphasise or drop certain features. The

introduction of the attention mechanism filters out irrelevant information by selecting segments of relevant features.

II. Attention Mechanism: The attention mechanism inspired by the work from [69] which is initially designed as an interface to provide the decoder with information from every encoder hidden state. With attention mechanism in place, the model is able to selectively focus on valuable parts of the input sequence while learning the association between them. Attention mechanism has been applied to time series prediction [73], document/text classification [74, 75], context-based text information detection [76] and text summarisation [75] with success. Attention mechanism encourages dissimilarities across different filters in the convolution layer. This encourage divergence of features from the convolved feature map by transforming potential feature candidate to a salient one.

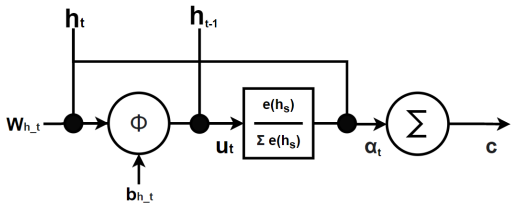

FIGURE 3.2: Illustration of the attention mechanism where ϕ is the hyperbolic tangent activation, W_{h_t} and b_{h_t} is the weight and bias learnt from output from recurrent layer, u_t is the "scores" of each timestep ranked according to the importance, h_s is the source and target to be evaluated for their fitness, α_t is the outcome of the evaluation and finally c is the weighted sum outcome.

The workings of a global attention mechanism is fairly simple. To compute the relevance between the inputs, or the scores, dot product is first applied to the learnable weights, W and current input from GRU, h_t with added bias b_w, denoted as

$$u_t \;=\; tanh(W_{h_t}\, h_t \;+\; b_{h_t}) \tag{3.2}$$

u_t is a hidden representation of h_t. The importance of the sample is then computed as the similarity of u_t with a sample level context vector h_s, where h_s is learned based on h_{t-1} and h_t. The weight α_t of each h_t is computed by

$$\alpha_t \;=\; \frac{exp(h_s)}{\sum_t exp(h_s)} \tag{3.3}$$

Finally, the context vector c is computed as a weighted sum of each h_t in H, which represents the information relevant to the current timestep,

$$c = \sum_t \alpha_t h_t \tag{3.4}$$

c can be understood as the aggregated information of the alignment vectors from previous step.

Algorithm 1 Recurrent-Attention Networks

Require: input sample x, timestep T

initialise W_z, W_r, W_h, W, U_z, U_r, U_h, b_z, b_r, b_h, b
for t=1, ..., T **do**
    procedure Forward GRU
    Update gate $z_t = sig(W_z x_t + U_z \overrightarrow{h_{t-1}} + b_z)$
    Reset gate $r_t = sig(W_r x_t + U_r \overrightarrow{h_{t-1}} + b_r)$
    State activation $\overrightarrow{\hat{h}_t} = tanh(W_h\, x_t + r_t \odot U_h\, \overrightarrow{h_{t-1}} + b_h)$
    Output $\overrightarrow{h_t} = (1 - z_t) \odot \overrightarrow{h_{t-1}} + z_t \odot \overrightarrow{\hat{h}_t}$
    procedure Backward GRU
    Update gate $z_t = sig(W_z x_t + U_z \overleftarrow{h_{t-1}} + b_z)$
    Reset gate $r_t = sig(W_r x_t + U_r \overleftarrow{h_{t-1}} + b_r)$
    State activation $\overleftarrow{\hat{h}_t} = tanh(W_h\, x_t + r_t \odot U_h\, \overleftarrow{h_{t-1}} + b_h)$
    Output $\overleftarrow{h_t} = (1 - z_t) \odot \overleftarrow{h_{t-1}} + z_t \odot \overleftarrow{\hat{h}_t}$
    Obtain feature vector $h_t \leftarrow$ concatenate $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$
    procedure Attention Mechanism
    Weigh feature importance $u_t = tanh(W_{h_t} h_t + b_{h_t})$
    normalise importance weight $a_t = \frac{exp(u_t)}{\sum_t exp(u_t)}$
    context vector $v_t = a_t h_t$
end for
Output symbol probability $\hat{D} \leftarrow softmax(W \sum_T v + b)$
return $\hat{D}$

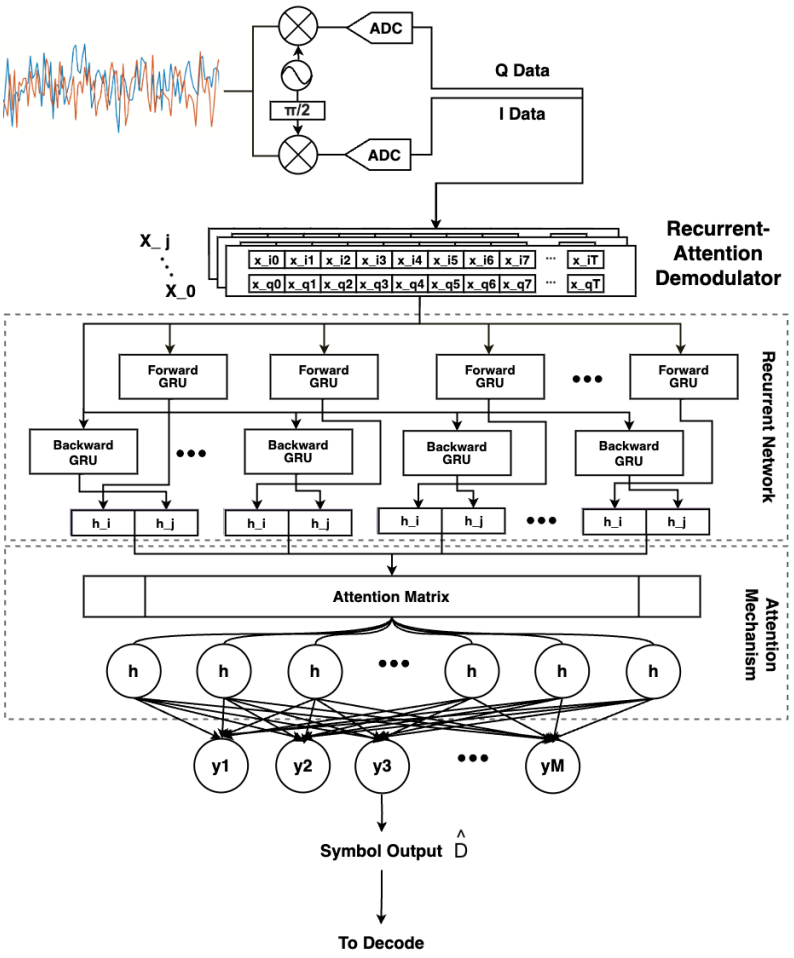

FIGURE 3.3: The block diagram of the receiver and the model architecture of the proposed RE-ATT.

Temporal features vector consisting of high-order sequential regularity of the signal data outputted by the forward and backward GRU in the recurrent network are concatenated and passed to the attention mechanism to be transformed to a context vector. The attention mechanism collects the overall feature candidates globally and considers the relation between feature candidates to calculate the salience of the temporal features from GRU. Finally, the context vector is passed to an output layer with softmax output layer to determine the learned vector's class membership probability.

Both GRU and the attention mechanism have $O(n)$ complexity where n is the length of the input sequence as they require sequential operations. The output FC layer has $O(4)$ complexity. Hence the total complexity for the RE-ATT model is $O(n^3 + 4)$.

3.2 Experiment Setup and Methodology

Experiment setup for the RE-ATT model including the dataset, system modelling, implementation details and evaluation metrics are detailed.

3.2.1 Dataset

RE-ATT aims to demodulate raw, time-domain baseband signal without the need for error compensation, low pass filtering or low noise amplifier typically implemented to pre-process the signal such as in the optimal receiver prior to symbol detection. Hence, a dataset X is generated to mimic real life signal used for training and testing of DL models. X is obtained from a simulated model of a radio transmission with the configurations shown in Table 3.1:

TABLE 3.1: Dataset Configuration

Modulation	4, 16 and 64-QAM
Coding	Uncoded
Channels	AWGN with -5 to 20dB E_b/N_0, Rayleigh flat fading with single path delay
Receiver Impairment	DC offset by $dc = 1db$, IQ imbalance of $g = 1dB$, $\varphi = 10$

The configuration are selected to closely emulate signals in real environment that are received by a DCR under fading conditions. To replicate signal qualities found in real

environment, randomised integers are generated using a pseudorandom algorithm with uniform distribution to avoid class imbalance problem. This randomly generated integers are also used as the ground truth label for training and validation. The sequence of integers, d, also known as symbols, are converted to bits then generated as M-QAM modulated baseband signals, s, where $M \in \{4, 16, 64\}$ and $M = 2^k$. The signal has a carrier frequency of 800 MHz and a symbol rate of 12.5 kHz. The complex-valued received signal (timestep omitted) can be denoted as:

$$X = s * H(c) + \mathcal{N} \ + dc + \varphi \tag{3.5}$$

Let $X \in R^d$ be a d-dimensional received signal, s the complex-valued transmitted signal, $H(c)$ the Rayleigh slow and flat fading channel, $\mathcal{N}$ the AWGN channel, dc the DC offset and ϕ the IQ imbalance. s is added with random noise of at specified strength to create data with E_b/N_0 ranging from -5dB to 20dB. The dc and φ are assumed to be constant over the symbol period. $Y \in \{-1, +1\}$ denotes the true signal. Both X and y will be analog-to-digital converted, undergo normalisation and symbol-rate sampling to obtain the real component I and the imaginary component Q vectors to be used as RE-ATT's input. The resulting baseband is sampled at 10 samples per symbol.

For each E_b/N_0, 1 million symbols are generated. This makes 41 million symbols per dataset.

3.2.2 System Modelling

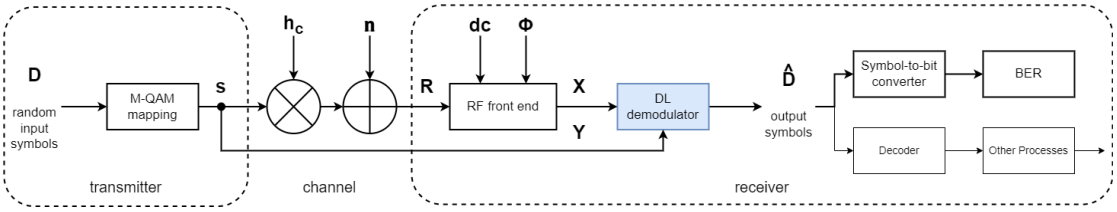

FIGURE 3.4: A block diagram of the simulated end-to-end signal transmission for a narrowband single-input-single-output (SISO) system. The flow of this research from dataset generation to performance measure follows the path highlighted in bold. The research solution is applied at the demodulation process highlighted in blue.

SISO simulation design using M-QAM modulation is shown in Figure 3.4. The implementation of the system consists of M-QAM modulation, upsampling, filtering, upconversion, and normalisation in transmitter side. On the receiver side, downconversion,

downsampling and M-QAM demodulation are performed. The demodulation phase integrates an NN architecture that is the proposed RE-ATT.

The system is modelled following these steps. First, data, or the information bits, is randomly generated, such that $d_i = [d(1)u(2)...d(n)]$ where n being the number of bits generated. Then, the symbols are formed by $k = log_2\ M$.

$$v = \sum_{i=1}^{k} u(i)\, 2^{ki} \tag{3.6}$$

The symbols are then processed as the modulation inputs, producing complex numbers corresponding to M-QAM signal. Upsampling involves passing previous process' symbols to a Squared Root Raised Cosine (SRRC) filter with an oversampling rate of 10. Subsequently, the signal undergoes upconversion, where each sample from the output of the pulse-shaping filter is multiplied by both cosine and sine signal samples, given by

$$\exp(\sqrt{-1} \times 2\pi f_c t) \tag{3.7}$$

where f_c is the sampling frequency.

The transmitted signal then passes through Rayleigh fading and AWGN channel, $\mathbf{n} \sim \mathcal{N}(0, \sigma^2)$ before reaching receiver's end as the received signal, r. The Rayleigh fading channel model has a single path delay with 0 dB gain, and Doppler shift frequency of 110 Hz. The AWGN channel model creates an average signal-to-noise ratio from -5 dB to 20 dB in increments of 1 dB. Then, R is downconverted to X where $R = s\ *\ h_c + \mathbf{n}$.

The channel is assumed to be a single path flat fading channel, modelled as a simple time delay without any additional filtering or attenuation. Therefore, the received signal experiences a constant time delay, without distortion or multipath effect caused by multiple delays with different amplitudes.

3.2.3 Implementation Details

From the 41 million symbols, the generated dataset is split into a ratio of $4 : 1 : 5$ for training, validation and testing. Pipelining of dataset is done by zipping the I and Q sequence of scalars with labels containing the symbol class ground truth. Each tensor in the dataset are reshaped to fit the dimension required by the model such that

(batch, samples, features). Mini-batches of 128 is used for introducing some noise in each gradient update to avoid being stuck in the local minima, to improve generalisation performance and achieve a relative speedy convergence and costing less memory. Prior to training, each batch also will be shuffled so that no two iterations over the entire sequence of training iterations and epochs will be performed on the exact same X. This allow the optimiser to have higher chance of escaping the local minima.

TABLE 3.2: The recommended configuration for recurrent-attention networks model.

Configuration	Description
Optimiser	Stochastic gradient descent with momentum decay [77], no scheduling or learning decay is required.
Learning Rate	The optimal learning rate is the largest learning rate that does not cause divergence of the training criterion, set at 10^{-2}.
Weight Initialiser	Glorot/Xavier initialisation [78] with normal distribution used in GRU, where the $\mu = 0$ and $\sigma^2 = \frac{1}{(input\_dimension + num\_of\_neurons)\,/\,2)}$
Weight Regulariser	L1 [79] and L2 [80] regularisers are used in GRU to curb exploding gradient. L1 ensures unimpactful parameters are driven to zero to encourage sparsity of the parameter values. L2 regularisation penalises large values more strongly and corresponds to a Gaussian prior.
Loss function	Sparse categorical CE is used so that labels need not to be encoded from binary to one-hot encoding.

For classification task, the model is trained to determine the symbol class value associated with the input vector x. While many research often treated demodulation as regression by minimising on the MSE, a more modern approach is to use an information-theoretic view that the network outputs represent posterior probabilities of each class. This can be achieved through applying the cross-entropy (CE) loss function to minimise on the estimated the posterior probabilities of each symbol class from RE-ATT's output. Empirically, by measuring the difference between the true distribution and the predicted

distribution, CE is defined as:

$$\begin{aligned} \mathcal{L}_{CE} &= -\sum_{j=1}^{M} p(Y_j)\ log\ \hat{D}_j \\ &= -\sum_{j=1}^{M} log\ (\hat{p}\ (p(Y_j) \mid X_j\ ;\ W)) \\ &= -log \prod_{j=1}^{M} \hat{p}\ (p(Y_j) \mid X_j\ ;\ W) \end{aligned} \tag{3.8}$$

where M is the modulation order and also the range of the symbol class, $p(Y)$ is the distribution of the ground truth signal while $\hat{D}$ is RE-ATT's output predicted distribution.

During offline training, RE-ATT calculates the derivative corresponding to variable parameters that are updated by backpropagation. During the updates, the differences between the each sample and output pair will decrease gradually and converge to a specific threshold. The model is trained with 100 epochs with a batch size of 128. Cross-validation is performed on the model. Table 3.2 records the details of the RE-ATT model's training configurations.

Signal generation and the conventional optimal zero-IF receiver (OPTIMAL) are both implemented in MATLAB. RE-ATT is implemented in Python using TensorFlow library. Experiments are carried out on a machine with an Intel Core i7-9700K CPU and Nvidia GeForce RTX 2080 GPU.

3.2.4 Evaluation Metrics

Standard metrics from the literature are used to evaluate the performance of DL model which is the accuracy and BER to measure the average probability of incorrectly identified symbol due to channel and receiver's imperfection. The accuracy is obtained from the amount of correctly predicted symbols over the total amount of prediction, while BER is denoted as

$$BER = \frac{E(t)}{N(t)} \tag{3.9}$$

where $E(t)$ is the number of symbols received in error over time, t, while $N(t)$ is the total number of symbols transmitted over time t. In a noisy channel, BER is often expressed

as a function of the normalised signal-to-noise ratio measure that is energy per symbol to noise power spectral density ratio E_b/N_0.

RE-ATT will also be compared to the following popular SOTA based on the research on previous works:

- Deep Neural Networks (DNN) [39]
- Convolution Neural Networks (CNN) [49]
- Recurrent Neural Networks (RNN) [18]
- Convolution-Recurrent Hybrid Networks (CNN-RNN) [18]

OPTIMAL is also used as a benchmark for fair performance comparison, defined by an architecture consisting of DC compensator, blind IQ estimator, matched filter and MLE detector. The combined functionality of DC compensator, blind IQ estimator, matched filter addresses signal impairments (DC offset, IQ imbalance) and optimises SNR, whereas RE-ATT is expected to handle DC offsets, IQ imbalances and noise filtering in an integrated manner. As minimising the negative log of the joint likelihood is equivalent to maximising the joint likelihood, therefore, determining the model parameters by minimising CE is comparable to MLE detector when performing symbol detection.

3.3 Experiment Results

The following benchmarks are used to compare with the results of RE-ATT: (i) The theoretical evaluation of BER for uncoded QAM signals (ANALYTICAL), (ii) the coherent demodulation method via match filter and MLE (OPTIMAL) (iii) the deep convolutional neural network model (CNN) by Mohammad et al. [49], (iv) the hybrid convolution and recurrent neural network (CNN-RNN) by Wu [18], and, (v) the recurrent neural network (RNN) with 128 hidden units from Wu [18]. ANALYTICAL for Rayleigh and AWGN channel assumes a receiver with maximal-ratio combining scheme. The architecture and fine-tuning for RNN and CNN-RNN follows [18] exactly, leaving any unspecified parameters at default while the same is also done for CNN. Each model is trained separately and tested on the dataset generated as described previously.

Monte-Carlo simulation is performed to obtain the average of the results for a more accurate overall performance. Through Figure 3.5, 3.6 and 3.7, it is found that RE-ATT has the best performance over all SOTA and conventional benchmark while maintaining low complexity seen in Table 3.4. Note that ANALYTICAL is a theoretical BER that does not take DC offset and IQ imbalance into account by assuming a perfect receiver [3]. For 4-QAM signal, RE-ATT is able to achieve a BER performance surpassing ANALYTICAL up to 0.5dB. For 16-QAM and 64-QAM signal, RE-ATT performs better on low SNRs where higher SNRs see less improvements.

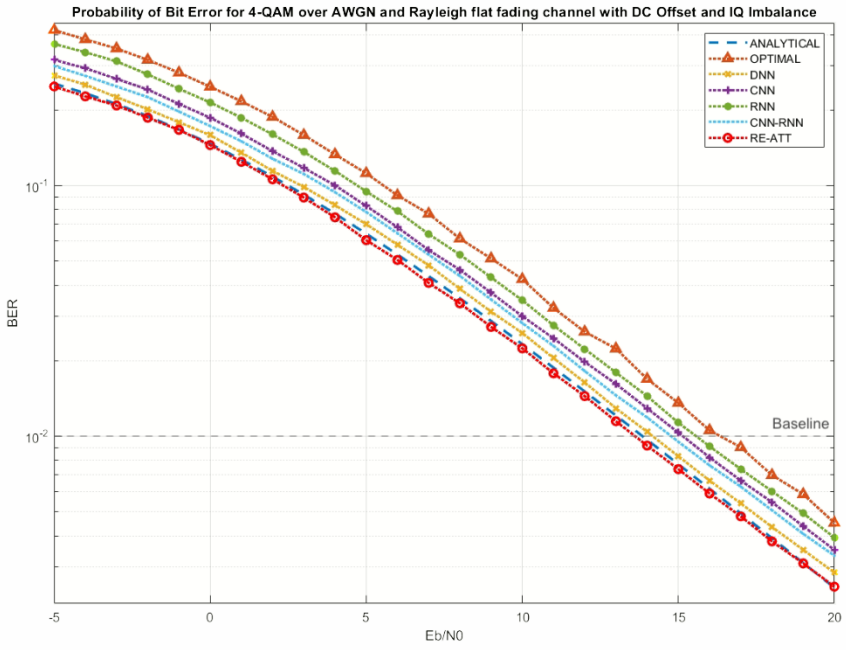

FIGURE 3.5: BER plot for performance comparison over 4-QAM signal.

The symbol accuracy of all demodulator is also calculated, where $SymbolAccuracy = \sum_{n=-5}^{20}(1 - SymbolErrorRate) * 100\%$. The symbol accuracy of each demodulator on each modulation order has been plotted in Figure 3.8, shows RE-ATT suffers the least performance degradation over increased modulation order while some demodulators such as RNN and CNN suffers exponential performance degradation over increased modulation with ANALYTICAL serving as control and OPTIMAL serving as benchmark. The symbol accuracy degradation also translates to the BER performance degradation as observed in Figure 3.5, 3.6 and 3.7.

In Table 3.3, the four implemented benchmarks are compared to the proposed solution as well as the dB gain from the theoretical analysis for each demodulator. The overall

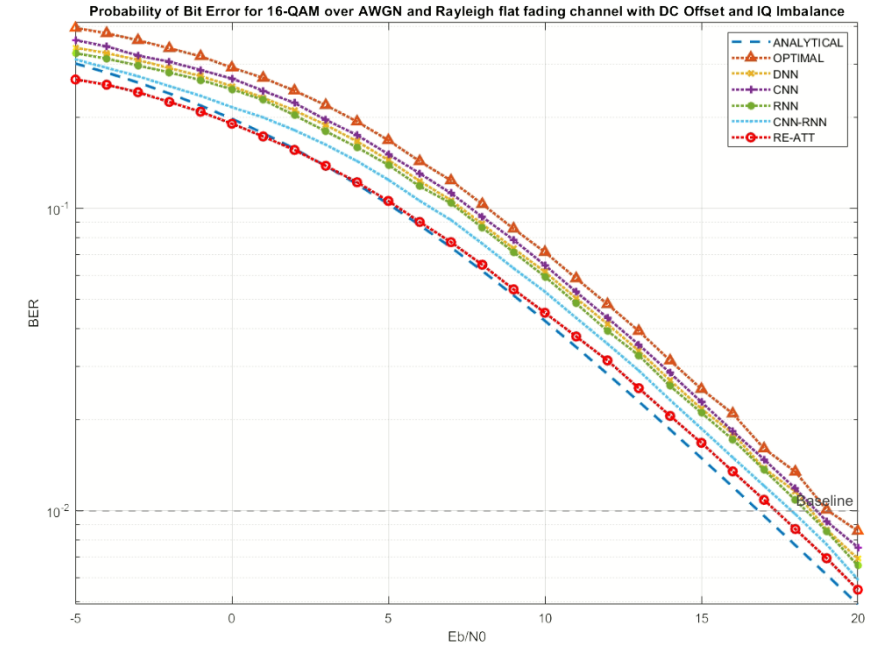

FIGURE 3.6: BER plot for performance comparison over 16-QAM signal.

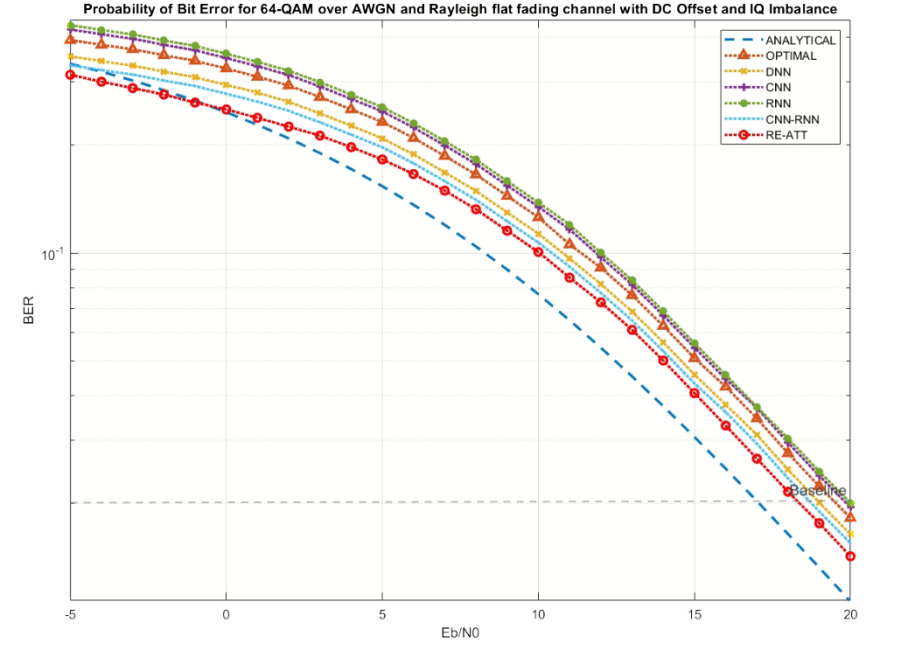

FIGURE 3.7: BER plot for performance comparison over 64-QAM signal.

accuracy across all E_b/N_0 for AWGN signal shows that RE-ATT is still at a competitive level as OPTIMAL albeit the 0.2dB difference when measured at a bit error of 10^{-2}. When looking at the performance comparison for faded signal, RE-ATT's advantage becomes more apparent. The best case performance for RE-ATT across all dataset when

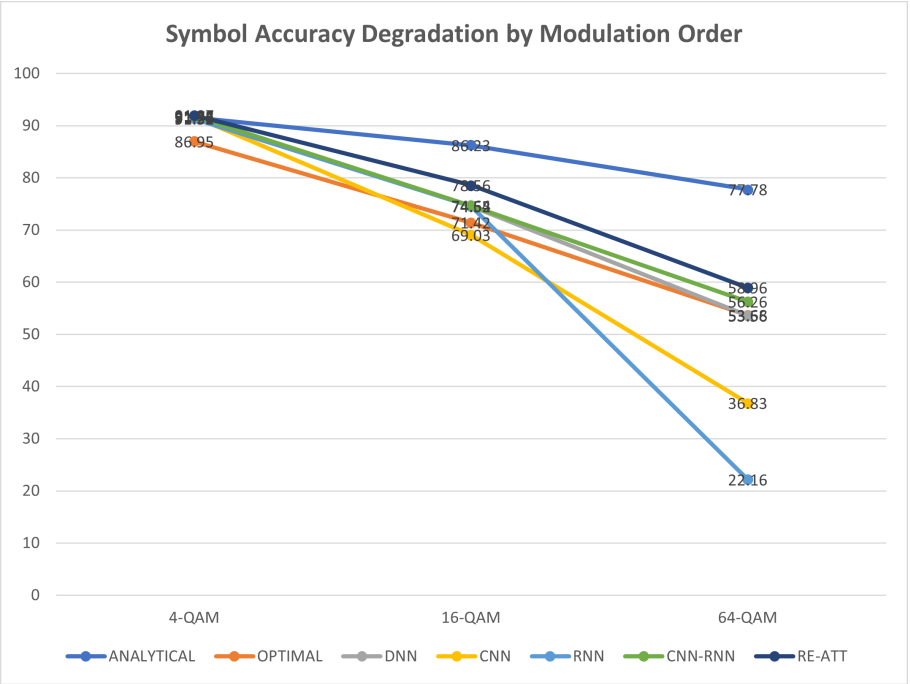

FIGURE 3.8: Symbol accuracy for each DL-based direct signal-to-symbol demodulator across QAM signal with various modulation order.

compared to OPTIMAL can go up to 3dB at high SNR conditions especially observed from Figure 3.5 and 3.6, while the averaged performance gain from ANALYTICAL is 0.2dB at 10^{-1} BER and 1dB at 10^{-2} BER. RE-ATT is shown to bear satisfactory results over OPTIMAL as well as other learning-based demodulator.

Unlike the results shown in [49], the implementation of SOTA on the same signal data from this research does not lead to a drastic learning-based gain of 10dB. It is speculated that the signal data used by [49] sampled at 1MHz with 100 samples per symbol, with an undisclosed carrier frequency and channel bandwidth, is highly oversampled. This allows a more effective gain of received SNR due to lowered noise floor as noise spreads out over more frequencies. As the way CNN's convolution layer works can be understood as similar to a matched filter, the classifier is able to benefit from a smoothened signal with increased SNR. While [18] also followed the same configurations as [49] for their signal data, there is no result observed to outperform correlation for faded signals, including their implementation of CNN which sees a much higher number of filters than CNN by [49].

Given that both GRU and attention mechanism also rely on past and present observations to compute meaningful feature map to be classified, it is theorise that RE-ATT

TABLE 3.3: Performance analysis for each demodulator of various benchmark methods for AWGN and Rayleigh flat fading datasets with 4-QAM, 16-QAM and 64-QAM signals, respectively. For non-learning method- the ideal BER ANALYTICAL and the OPTIMAL receiver; for learning-based methods- single layer NN, CNN, CNN-RNN, RNN and RE-ATT (proposed). The evaluated metrics are the symbol test accuracy, the weighted average of precision and recall (F1) and the E_b/N_0 at 10^{-2} BER. E_b/N_0 should be the lower, the better.

Wireless Channel		n=AWGN, h=Rayleigh		
Attributes		Symbol Acc (%)	F1 Score	E_b/N_0 @10^{-2}
4-QAM	ANALYTICAL	92.55	N/A	14.1
	OPTIMAL	92.62	0.926245	16.0
	NN	92.81	0.928119	14.2
	CNN	92.82	0.928245	15.2
	CNN-RNN	92.78	0.927772	15.0
	RNN	92.83	0.928328	15.3
	RE-ATT	**92.51**	**0.9251473**	**13.8**
16-QAM	ANALYTICAL	89.73	N/A	16.7
	OPTIMAL	87.23	0.872327	19.0
	NN	88.35	0.883492	18.8
	CNN	88.21	0.882079	18.9
	CNN-RNN	88.58	0.885872	17.4
	RNN	88.35	0.883548	18.8
	RE-ATT	**89.54**	**0.895412**	**16.9**
64-QAM	ANALYTICAL	86.32	N/A	20.3
	OPTIMAL	84.22	0.842227	19.8
	NN	84.05	0.840522	19.7
	CNN	83.19	0.831898	20.1
	CNN-RNN	84.15	0.841387	19.5
	RNN	83.55	0.835487	20.0
	RE-ATT	**93.93**	**0.939312**	**18.0**

will suffer in performance if sampled at Nyquist rate. 10 samples per symbol is ample for RE-ATT to demodulate. Applying attention to the input has an almost denoising effect as certain interesting segments is given a higher weightage than others.

3.4 Complexity vs Performance

While computationally expensive models may not be an issue in other applications, it is an important consideration to compare the gain in performance that the model is able to benefit from every additional cost to complexity for potential embedded applications. A model with either deeper architecture or larger amount of neurons always translates to higher complexity. This is especially true for recurrent networks.

As depicted in Figure 3.9(i), it is evident that the RE-ATT model exhibits exponential growth in both the total number of parameters and the FLOPs, in tandem with the exponential expansion of neuron units within the recurrent layer. The proliferation of parameters directly corresponds to an augmented demand for memory storage, whereas the escalation in FLOPs is directly associated with amplified training and runtime requirements. Conversely, the performance assessment, quantified by symbol test accuracy, reveals a comparatively modest increment of less than 1% for each exponential augmentation in the number of GRU neuron units, as illustrated in Figure 3.9(ii).

While it can be observed that there is a positive relationship between the number of neuron units and the accuracy of the model, the gain in accuracy for the highest amount of neuron units is minimal. This is also agreed by [18] where an input sequence of 100 samples per symbol is fed into models with different neuron units of 8, 16, 32, 64, and 128. Hence, rather than increasing the number of neuron units, model optimisation will make a more fitting solution to achieve similar results. For this use case, the proposed

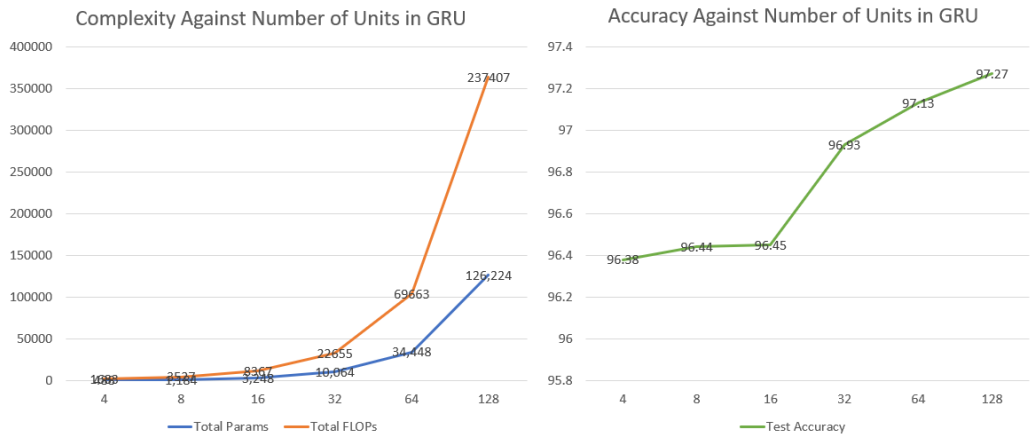

FIGURE 3.9: Comparison of computation cost and performance for different amount of neuron units (horizontal axis) in a recurrent layer.

work builds the model with 32 GRU neuron units in each layer, due to considerations of complexity versus performance.

Table 3.4 compares the complexity of each DL-based demodulator. The total number of trainable parameters gives a rough estimate of the size of the architecture. The power consumption of a digital block depends heavily on the number of arithmetic operations required to perform each task. Hence, it is necessary to minimise the architectural complexity and the number of parameters required for training as they are relative to FLOPs.

CNN-RNN is visibly larger and more memory-consuming than other DL-based demodulators while the RE-ATT has the advantage of having the simplest architecture. The estimated processing time per bit where processing rate = time taken / (total number of symbols x number of bits) is calculated from every 1 million symbols multiplied by 2 bits then divided by the prediction time taken per 1 million symbols. The FLOPs obtained for each model is the pure floating operations within each model layer per second without parallel or batch processing. It is possible to further reduce the operations and increase runtime via weight pruning.

While it is noticed that the FLOPs count for CNN is different from as stated in [49], it could be due to the difference of implementation within each layer and the pipelining. The FLOPs count obtained is solely for Tensorflow API implementation.

TABLE 3.4: Each DL-based demodulator is evaluated on the total number of parameters, the total floating point operations (FLOPs), its approximated peak heap memory usage measured at mega bytes and the approximated data rate measured at kilo bits for every modulation order. The proposed RE-ATT has a significantly lower total number of parameters, lower FLOPs and higher data rate than CNN, RNN and CNN-RNN. Although the overall complexity is not as low as DNN, RE-ATT is able to achieve a better symbol accuracy than DNN by 3% observed in Figure 3.8 which also translate to better BER performance.

Modulation	4-QAM				16-QAM				64-QAM			
DL-Based Demodulator	Total Params	Total FLOPs	Approx. Peak Mem. Usage (MB)	Approx. Data Rate (kb/s)	Total Params	Total FLOPs	Approx. Peak Mem. Usage (MB)	Approx Data Rate (kb/s)	Total Params	Total FLOPs	Approx. Peak Mem. Usage (MB)	Approx. Data Rate (kb/s)
DNN	6,424	12,604	≤ 1	163.8	9,760	19,312	2	77.1	25,984	51,904	4	29.5
CNN	20,678	40,438	7	158.2	20,786	40,702	13	69.3	21,218	41,758	21	26.7
RNN	241,964	215,808	12	133.4	242,072	215,808	21	63.3	242,504	215,808	31	21.4
CNN-RNN	119,060	238,040	15	135.9	119,168	238,304	25	65.8	119,600	239,360	37	23.5
RE-ATT	8,280	27,184	5	161.4	10,464	37,292	8	74.6	15,936	55,952	14	29.1

Although CNN has a deeper architecture than RE-ATT, convolution operations are performed in parallel while GRU needs to be processed sequentially. Mathematically speaking, matrix multiplication in convolution layer is also more efficient than dot products performed iteratively in a recurent layer. This caused the runtime complexity of RE-ATT 3 times more than CNN. Nevertheless, RE-ATT has a slight advantage in process speed over RNN and CNN-RNN due to the attention layer being an effective replacement for a conical fully connected neural network.

Still, compared to the other SOTA which are built with deeper networks, RE-ATT has the advantage of size and performance.

3.5 Ablation Study

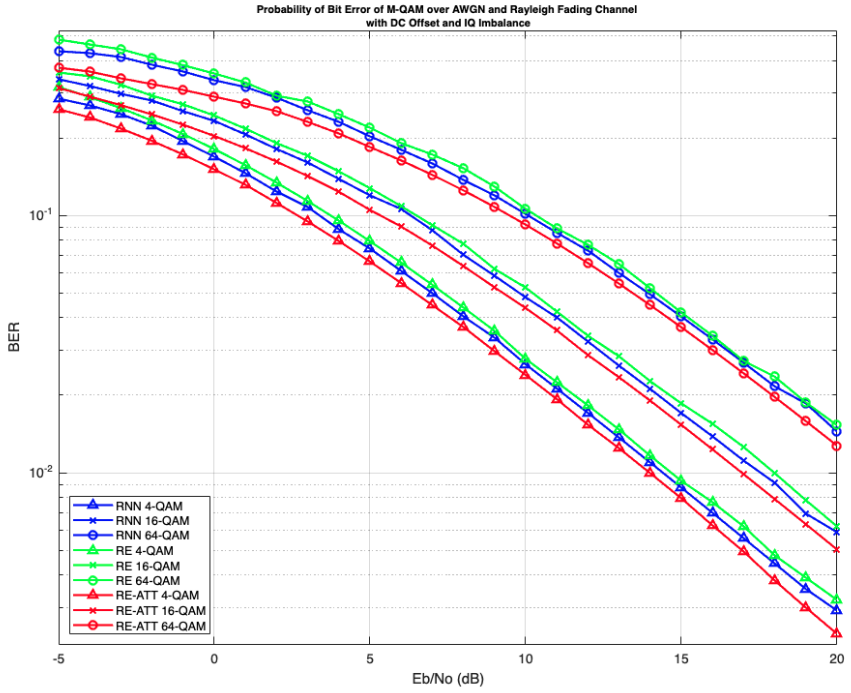

FIGURE 3.10: Ablation done on 4-QAM, 16-QAM and 64-QAM dataset with results where the log average bit error rate plot of bi-directional GRU model (RNN), recurrent network model (RE) and recurrent-attention networks model (RE-ATT) at normalised SNR from 0dB to 40dB obtained after Monte-Carlo simulation. All three models have the same amount of GRU unit and use the same optimisation.

An ablation study is conducted to further test the impact of attention mechanism in a recurrent model by comparing between two identical models: a recurrent neural network with bidirectional GRU and two feed-forward layer (RNN), a similar recurrent network

(RE) without the feed-forward layer and RE-ATT. Different from the benchmark model RNN with 128 hidden units, the RNN employs the same amount of GRU units and optimisation as RE-ATT. The same is done for RE. In such controlled environment, RE-ATT improves by an average of 1.0dB at 10^{-1} BER and 0.9dB at 10^{-2} BER in comparison to RNN for noisy and faded signal while gaining an overall improvement of 1.3dB at 10^{-2} BER and 1.5dB at 10^{-3} BER over RE as seen in Figure 3.10. The attention layer does provide solid improvement by increasing the weight on important segments and diminish the weight of redundant segments.

3.6 Summary

Given a problem of demodulation as a classification task with time series input data and symbol output, the search of a DL model that can most effectively extract salient features is conducted. Four different DL classifier models taken from the literature are trained to demodulate M-QAM signals with noise, flat fading, IQ imbalance and DC offset. The dataset containing signal with ample complexity and practicality for narrowband signal transmission is generated from the simulated model. A new classifier architecture has been proposed to perform direct signal-to-symbol demodulation on time series signal data. The proposed RE-ATT has shown that it is able to achieve better performance at a lower complexity in comparison to other deep architectures suggested in literature and conventional method. An average of 3dB gain when compared to OPTIMAL across different modulation order can be achieve by RE-ATT. RE-ATT also has several advantages as a demodulator such that is not limited to fixed-length vector input, has the lowest parameter count compared to most DL-based architectures suggested in literature and is able to achieve good performance at a complexity comparable to conventional methods.

Chapter 4

Low Complexity Optimisation

Unlike high-level applications such as computer vision tasks or generative tasks, applications of deep learning in low-level or the physical layer, especially on low-power devices within wireless communication systems are characterised by limited computational resources, limited memory capacity and constrained energy budgets. Optimising neural network models for these devices is motivated by the need to minimise resource consumption, which is critical for prolonging battery life and ensuring energy-efficient operation. Apart from that, deploying optimised models also reduces latency by maintaining low runtime complexity and more scalable batch processing by low memory consumption. Lower memory complexity can help reduce memory access times, which can impact inference or training speed, leading to lower latency and improved performance. This is because high memory usage can limit the number of processes that can be run concurrently on a cluster or the ability to use parallel processing efficiently. Therefore, low complexity optimisation comes into play to ensure neural network models that can run efficiently during inference.

Low complexity optimisation does not just reduce model complexity as it often goes hand-in-hand with strategies that can also enhance model performance. By simplifying and optimising the model, it is possible to achieve power-efficient deep learning models that are capable of maintaining or even improving their performance. Optimising deep learning methods to increase performance while minimising runtime complexity and memory usage is crucial for low-power RF communication devices, where both computational and memory resources are limited. There are several strategies

can be employed to achieve this goal. Reducing the precision of the model's weights and activations through quantisation reduces memory usage without significant loss in performance. Another method is to implement quantised inference using fixed-point or integer arithmetic, which can significantly reduce the memory bandwidth and computational requirements during inference. Pruning techniques helps to remove redundant or less important parameters, leading to a more compact model. Parameter sharing, tensor factorisation and approximation of weight matrices using low-rank factorisations such as singular value decomposition (SVD) or tensor decomposition are also viable ways to compress the model size and reduce memory usage without sacrificing performance significantly.

As efficient neural network model architecture designed with a focus on low BER performance has been explored in the previous chapter, in this chapter, the focus is placed on techniques to three goals: reduce the number of unnecessary parameters and the precision of model weights to lower memory and computation complexity, reduce the inference time through reduced amount of operations and memory access, and increase the model's performance.

4.1 Model Compression with Weight Pruning and Quantisation

Although RE-ATT is already designed with a shallow architecture with a relatively low number of parameters, it is still estimated to requiring $O(10^5)$ of storage just for the purpose of demodulation. This is not acceptable on device with limited memory. Furthermore, where inference invokes arithmetic operations and memory access in the millions, the power consumed by the processor will dissipate heat, drain battery capacity may also exceed the device's thermal limits.

Introduction of sparsity in the model is an important factor due to this. One way of achieving this is to apply magnitude-based weight pruning [81–83] that removes near-zero weights and zero weights as these less salient connections are unlikely to provide valuable contribution to the final output [84]. This creates a sparse network with fewer parameters. The magnitude indicates the importance or contribution of a weight to the final output. The magnitude-based weight pruning method is simple to implement and

computationally efficient, as compared to other pruning techniques like optimal brain damage, due to being more general and does not require calculating second derivatives or Hessian matrices. It has been shown to reduce the total number of parameters in the model while maintains model accuracy in practice across different network architectures in literature studied by [84].

Algorithm 2 Application of magnitude-based weight pruning to a trained model

Require: pretrained model F, target sparsity S
1: initialise sparsity s
2: **while** s ¡ S **do**
3: Calculate magnitude $|W|$ of all trained weights
4: Sort weights W by magnitude $|W|$
5: Prune p smallest magnitude weights by setting them to 0
6: Update sparsity $s = s + p$
7: Fine-tune model to recover lost accuracy
8: Update weights W
9: **end while**

Pruning is applied to the recurrent layer, where the weights from this layer are mostly concentrated at zero with a normal distribution [83]. A sparsed model is benefiting in disk compression due to the zeroes being skipped during inferencing. Given that categorical cross-entropy loss is used for multi-class labelling, skipping zero values can speed up execution and save some memory by avoiding logs and sum over zero values in one-hot encoded vectors.

Quantisation is another method to compress the model's memory consumption during inference. Quantisation of the model from 32-bit floating point into the IEEE standard for fixed-point 16-bit floating point[1] is adopted to compress the model's memory consumption during inference. Both methods significantly reduces model size, making it more feasible to run a deep learning model on a memory-constrained device with less processing capabilities. The results of the model compression can be found in section 4.5 Table 4.1 without negative impact on the model's performance.

4.2 Momentum Decay for Optimiser

Numerous variations of the gradient descent optimisation algorithm exist, where they fundamentally involves the computation of the partial derivative of the cost function

[1] https://www.tensorflow.org/lite/performance/post$_t raining_q uantization float16_q uantization$

concerning each adaptable parameter. Following this, the algorithm adjusts each parameter in the direction that leads to a decrease in the overall cost. The selection of an appropriate gradient descent optimiser is crucial for training neural networks effectively, as different optimisers involves multiple hyperparameters such as learning rate, learning rate decay, momentum, and weight decay, which require proper tuning for optimal performance. Among adaptive optimisers, Adam [85] has become a popular choice for neural network training.

AdamW [86], a variant of Adam with decoupled weight decay from gradient-based update, includes a penalty term in the loss function that allows weight decay to discourage the use of excessively large weight values during training. Decoupling weight decay from the gradient update step allows AdamW to overcome the inequivalence between L2 regularisation and weight decay in the standard Adam implementation. Weight decay is applied after controlling the parameter-wise step size, ensuring it does not contribute to the moving averages and remains proportional to the weight itself. This decoupling also enhances the independence of optimal settings for the learning rate and weight decay factor, reducing the criticality of striking a perfect balance between the two hyperparameters for improved performance.

However, the implementation of decoupled weight decay in gradient descent optimisations has its drawbacks, as elucidated in Decoupled Weight Decay Regularisation (DWDR). Firstly, DWDR introduces an additional computation term in the weight update equation, involving the regularisation strength λ multiplied by the weights. This extra computational overhead may lead to a slight increase in complexity compared to standard weight decay with L2 regularisation, which only involves the sum of squares of the weights. Secondly, tuning the regularisation strength λ in DWDR remains a challenging task, despite the improved independence between the learning rate and λ. Extensive hyperparameter tuning may be required to determine the optimal value for λ. Thirdly, DWDR's performance can be influenced by the learning rate schedule adopted during training. Consequently, selecting an appropriate learning rate schedule that complements DWDR becomes pivotal for achieving superior results.

To enhance performance and reduce dependency on extensive hyperparameter tuning, applying a simple technique to the momentum parameter instead of the weight parameter shows promise in boosting overall performance and convergence speed during gradient

descent optimisation. Additionally, when aiming for state-of-the-art results, stochastic gradient descent (SGD) is preferred, as models trained with Adam have been observed to not generalise as well as SGD [86]. SGD with momentum offers advantages over Adam in scenarios with sparse or varying gradients, providing a consistent and effective learning rate update, making it more suitable for handling such conditions. Moreover, SGD with momentum is memory-efficient, requiring fewer parameters compared to Adam.

SGD conditioned with a momentum term offers several benefits. In particular, it is observed to have faster convergence compared to plain SGD. By incorporating a momentum term that accumulates a weighted average of past gradients, the optimiser gains momentum and moves more decisively along the steepest direction. This acceleration aids in escaping local minima and saddle points, resulting in quicker convergence compared to the standard plain SGD optimisation. Additionally, SGD with momentum reduces oscillations during training, especially in highly irregular or noisy loss landscapes, enhancing stability and improving efficiency in reaching the optimal solution. Moreover, SGD with momentum exhibits less sensitivity to the learning rate hyperparameter than plain SGD. Typically, a low learning rate of 10^{-3} can be too slow to train for a plain SGD, yet a high learning rate of 10^{-1} can cause a plain SGD to diverge into infinity. SGD with momentum can effectively handle larger learning rates without diverging during optimisation, allowing for more significant flexibility in tuning the learning rate and potentially speeding up training while enhancing generalisation. However, constant momentum may result in instability, even if successfully landing in minima.

In light of the aforementioned limitations, there is a motivation to seek methods for enhancing performance and reducing the dependency on extensive hyperparameter tuning. One promising approach involves applying a simple technique to the momentum parameter instead of the weight parameter to boost overall performance during gradient descent optimisation. The momentum term, accumulating a weighted average of past gradients, accelerates gradient descent optimiser along the steepest direction, aiding in escaping local minima and saddle points, leading to quicker convergence. To address this, application of momentum decay is proposed by introducing the decaying momentum schedule [87] to SGD with momentum (SGD-m) shown in Algorithm 3. SGD-m with decaying momentum schedule has been shown to be less sensitive to hyperparameter tuning while achieving competitive error in various experiments [87]. Table 4.1 shows the performance comparison before and after optimisation.

Algorithm 3 Learn with Stochastic Gradient Descend with Momentum Decay

Require: model hyperparameters Θ, no. of iterations I, learning rate η, momentum β, accumulated momentum decay rate , stochastic gradient ∂

1: initialise β,
2: **for** i=1, ..., I **do**
3: Apply decay to momentum $\beta_i = \beta_{init} \cdot \frac{1 - \frac{i}{I}}{(1-\beta_{init}) + \beta_{init}(1-\frac{i}{I})}$
4: Update gradient with momentum $_{i+1} = \beta_i \, _i - \eta \, \partial i$
5: Update model hyperparameters $\Theta_{i+1} = \Theta_i - \eta \, g_i + \beta_i \, _i$
6: **end for**

Decaying momentum reduces the momentum parameter over time during training of a neural network. The momentum parameter controls how much the update to a weight depends on previous updates. In general, for a given set of parameters denoted as $w = w_0, ..., w_N{}^T$, the momentum update rule is govern by the velocity update and parameter update defined as:

$$\begin{aligned} v_t &= mu * v_{t-1} - \eta * \nabla J(\Theta_t) \\ w_{t+1} &= w_t + v_t \end{aligned} \tag{4.1}$$

where v is the momentum vector, mu is the momentum parameter, η is the learning rate, ∇ is the gradient, and w are the weights. With momentum, the weight updates depend partly on previous updates. This helps accelerate training in flat regions of the loss surface and prevent oscillations, but can cause overshooting optima or explosions if momentum accumulates too much. Decaying momentum means reducing mu over time. For example, decaying linearly from 0.9 at the start of training to 0.0 at the end. This helps the model to adapt to the changing contours of the loss surface during training. Early in the training, high momentum accelerates progress. As the surface becomes more curved throughout the progress, lower momentum allows more precise optimisation. This is akin to a discrete-time recursive filter with low-pass transfer function. The DC gain of the filter yields an amplified learning rate when the partial derivative slowly changes and vice versa when the the partial derivative changes quickly. Due to noise or oscillation near minima of the loss landscape, the low-pass characteristics of momentum will dampen the learning rate. Furthermore, the parameter update considers both the current gradient and the accumulated past gradients, meaning that contributions are weighted.

The decaying momentum schedule proposed in [87] decays momentum non-linearly from an initial value to 0, formularised by:

$$\beta_t = \beta_init * (1 - t/T)/(1 - \beta_init + \beta_init * (1 - t/T)) \tag{4.2}$$

where t is the current step, T is the total steps, and β_init is the initial momentum value.

The decaying momentum schedule can help improve model training in several ways:

- **Adaptivity to curvature.** Early in training, high momentum can help accelerate progress in relatively flat regions of the loss landscape. As the loss landscape becomes more curved in later iterations, lower momentum allows more adaptation to local curvature and avoidance of overshooting optima. Decaying momentum from high to low allows training to benefit from both high and low momentum regimes.

- **Annealing exploration to exploitation.** High momentum encourages exploration of the loss landscape early in training. Low momentum encourages finer exploitation of optima later in training. Decaying momentum smoothly transitions training from an exploratory phase to an exploitative phase.

- **Avoid oscillations.** High, constant momentum can sometimes lead to oscillations around optima later in training. Decaying momentum reduces this oscillatory behavior and allows finer convergence.

- **Reducing effective learning rates.** Decaying momentum has a similar effect to decaying learning rates, where the effective learning rate reduces over time, allowing finer optimisation. However, decaying momentum allows the actual learning rate parameter to remain constant.

- **Limit gradient accumulation.** With constant high momentum, gradients can accumulate excessively over many timesteps. Decaying momentum limits this accumulation, preventing exploded or stale gradients.

4.3 Loss Regulariser for Maximum Class Separability

The process of training a deep learning model involves adjusting the coefficients to minimise the error, often measured by the difference between the predicted values and the actual targets, using gradient descent optimisation. In linear models such as linear regression or logistic regression, coefficients represent the slopes or weights assigned to each feature. They define how much each feature contributes to the model's prediction. In nonlinear models such as a neural network with nonlinear activation functions, coefficients are often used to represent the weights of the connections between neurons in the neural network. These weights determine the strength of the connections and play a critical role in how the network processes and transforms input data to produce output. While the strategy for SGD optimiser in section 4.2 tackles overfitting by solving the issue of overshooting, local minima convergence and learning rate problem, a follow-up is required to further control the model complexity.

Regularisation is a technique often used to avoid the overfitting of a neural network model that has more parameters than the input data and to learn from noisy inputs with better ability to generalise. Regularisation encourages the generalisation by avoiding a perfect fit of the coefficient with the training data samples. While it is also possible to prevent overfitting by other methods such as increasing or changing the distribution of the training dataset, perform data argumentation, perform dropout, or early stopping; each method has their own pros and cons. In the context of signal demodulation, the aforementioned methods may not be ideal mainly due to the nature of signal data having different levels of SNR and channel conditions, therefore difficult to extract precise information from the signal or determine a clear validation loss threshold. Therefore, regularisation applied to the loss function of a model is investigated to help reduce overfitting and improve model performance without disrupting the learning process.

One factor heavily impacting the chances of overfitting lies in the features learned by a model. Intuitively, it is understood that discriminative features need to be generated to further improve the performance of deep learning models. However, features generated for time series signal data by hidden layers of a deep learning model are often unintelligible, therefore difficult to define a meaningful trend or pattern. Rather than trial-and-error process in generating features, a more feasible method is to encourage

the model to penalise features that are less discriminative so that the model will naturally select discriminative features over the course of training. This can be done by adding a separability metric measured by the distance between each feature and their respective class centroid (intra-class compactness) as well as the distance between each class centroid and the other class centroids (inter-class separability) [88][89]. Intra-class compactness and inter-class separability can be used as indicators to measure the effectiveness of a model to produce discriminative features. In general, it is desirable to decrease the distance between each feature and their respective class centroid, which translates to high intra-class compactnese, and increase distance between each class centroid and the other class centroids, which translated to high inter-class separability. Intra-class compactness is measured by the mean distance of all samples to the class centroid:

$$C_i = (1/N) sum_{j=1}^{N} ||X_j - \mu(i)|| \tag{4.3}$$

where X_j are the sample vectors, N is the number of samples in class i and $\mu(i)$ is the centroid of i. Smaller intra-class compactness C_i indicates higher compactness within class i. On the other hand, inter-class separability can be measured as the Euclidean distance between the centroids (mean vector) of two classes:

$$S_i j = ||\mu(i) - \mu(j)|| \tag{4.4}$$

where $\mu(i)$ and $\mu(j)$ are the centroids of classes i and j, respectively. Larger S_{ij} indicates higher separability between classes i and j.

To achieve that, a metric called the separability metric is derived where the intra-class compactness is divided by the inter-class separability while the class centroids being the mean value of a feature cluster. The algorithm can be referred at Algorithm 4.

The separability metric is a regulariser to be added to an objective function in a classification model such as the SoftMax function rather than being used as a standalone loss as per suggested in [89]. This is to ensure a convex loss that is optimisable by a gradient descent optimiser. As the separability metric is a distance-based metric, mean squared error (MSE) is used for calculating the error. The new loss shall be defined as:

$$\mathcal{L}_{RE-ATT} = \mathcal{L}_{SoftMax} + \mathcal{L}_{Separability} \tag{4.5}$$

where $\mathcal{L}_{Separability}$ can be referred at Algorithm 4. By extracting the features in the final layer of the RE-ATT model where $\mathcal{L}_{RE-ATT}$ is applied, a feature distribution map can be visualised using t-distributed stochastic neighbor embedding (t-SNE) [90]. The separability metric is also applied to DNN as a control study. From Figure 4.2 and 4.3, it is observed that the addition of separability metric has indeed increased intra-class compactness and inter-class separability as compared to a typical SoftMax. A BER performance improvement can be found at section 4.5. However, the separability metric requires additional memory to store the centroids values and longer training time with the runtime performance unaffected.

Algorithm 4 Separability Metric Regulariser

Require: label d, features y, update scale factor α, margin scale factor λ, constant ε

initialise centroids of all classes C
get current class centroid $c \leftarrow C_d$
for i=1, ..., size(y) **do**
    calculate intra-class distance $INTRA_i = \|y_i - d\|_2^2$
    calculate inter-class distance $INTER_i = \|c - C_i\|_2^2$
    update centroids of all classes $C_i = C_i - [\alpha * (C_i - y)/(1 + d)]$
end for
calculate separability metric r $\leftarrow (\sum ICSa + \lambda)/(\sum ICSe + \varepsilon)$
return r

4.4 Experiment Setup and Methodology

Experiment setup for the optimisation of the classifier RE-ATT model including the dataset, implementation details and evaluation metrics are detailed.

4.4.1 Dataset

The same dataset in section 3.2.1 is utilised for the purpose of validation of the aforementioned optimisation methods.

4.4.2 Implementation Details

Implementation of RE-ATT is the same as stated in section 3.2.3. The optimisation methods are implemented in Python using TensorFlow library. Experiments are carried

out on a machine with an Intel Core i7-9700K CPU and Nvidia GeForce RTX 2080 GPU.

4.4.3 Evaluation Metrics

Several metrics are used for the evaluation of the effectiveness and efficiency of the three optimisation methods detailed in section 4.1, 4.2 and 4.3. Runtime performance and time complexity is measured by the total number of parameters, the total number of floating point operations (FLOPs), the approximated data rate. Memory complexity is measured by the approximated peak heap memory usages. The model's performance is measured by the symbol accuracy obtained by each SNR by percentage and BER to measure the average probability of incorrectly identified symbol post-optimasation. The optimised RE-ATT will be compared to the unoptimised RE-ATT, as well as a single-layer NN to act as a benchmark.

4.5 Results

Methods to reduce complexity and improve performance of the proposed RE-ATT have been applied in a gradual manner to observe the performance gain, tabulated in Table 4.1.

It is observed that post pruning and quantisation the RE-ATT model is significantly reduced in memory usage, computation complexity and increased data rate. Pruning and quantisation achieve their purpose with a 7.6x shrink of FLOPs, an approximated 50% reduction in memory usage, significantly faster training time and decrease prediction time by nearly 10 seconds for every 1 million symbols detected.

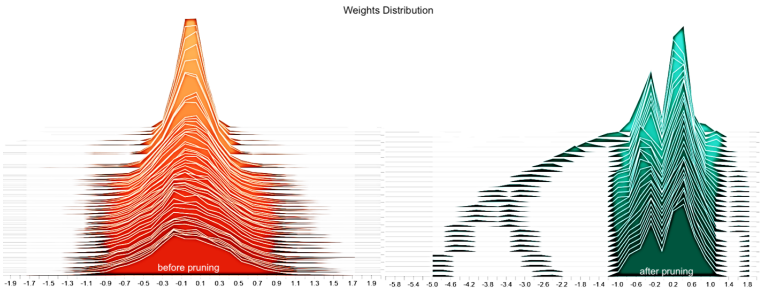

FIGURE 4.1: Weight distribution before and after pruning applied only to recurrent layer.

Although there is a slight increase of peak memory usage by 1MB, the separability metric loss regulariser increased symbol accuracy by an average of 1% while decreasing the total number of FLOPs and slightly increased data rate.

In the experiment, 2 classifier models, the proposed RE-ATT and a control group DNN, are selected to observe the effectiveness of the proposed separability metric loss regulariser. The improvement seen in Figure 4.2 and 4.3 is reflected in BER performance improvement of up to 0.5dB from their non-regularised SoftMax loss counterparts as observed in Figure 4.5.

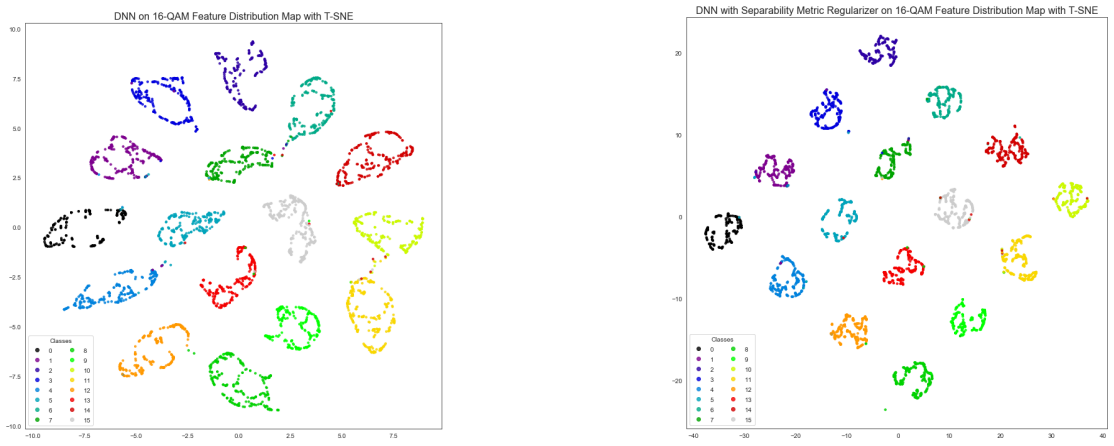

FIGURE 4.2: Comparison between two feature distribution of a pre-trained DNN mapped with T-SNE using 4000 samples from the same dataset described in Chapter 3. DNN with the loss regulariser applied (b) has a visibly improved intra-class compactness and inter-class separability than the DNN without the loss regulariser applied (a).

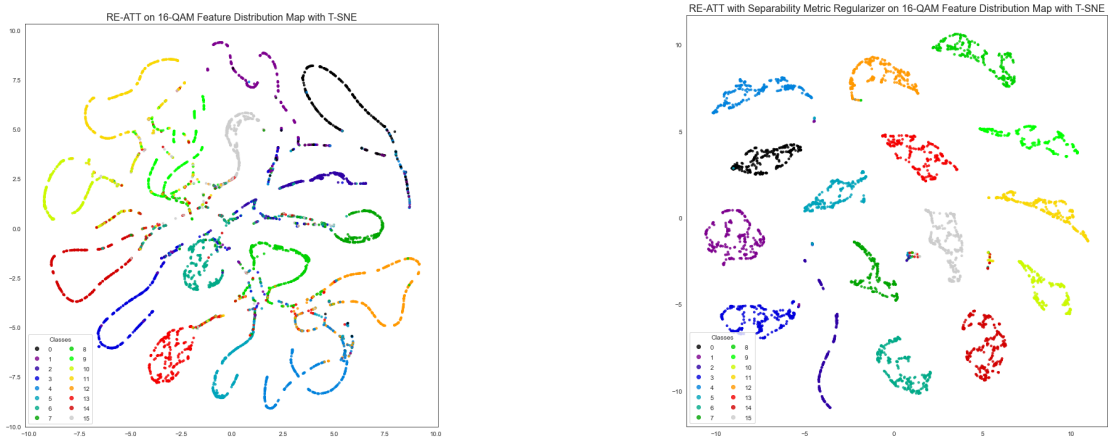

FIGURE 4.3: Comparison between two feature distributions of a pre-trained RE-ATT mapped with T-SNE using 4000 samples from the same dataset described in Chapter 3. RE-ATT with the loss regulariser applied (b) has a visibly improved intra-class compactness and inter-class separability than the RE-ATT without the loss regulariser applied (a).

Experiment model trained with SGD-m with decaying momentum yields further improvements over SGD-m in terms of trainable parameters but has significantly longer training time due to the need to update the momentum decay rate at every iteration and unchanged amount of FLOPs. Though there is an increase in training time, the total trainable parameters, overall memory usage, FLOPs and prediction time improved over Adam optimiser. It is observed that the loss landscape is a simple convex seen in Figure 4.4, where SGD-m has been observed that to diverge early on the 5th epoch.

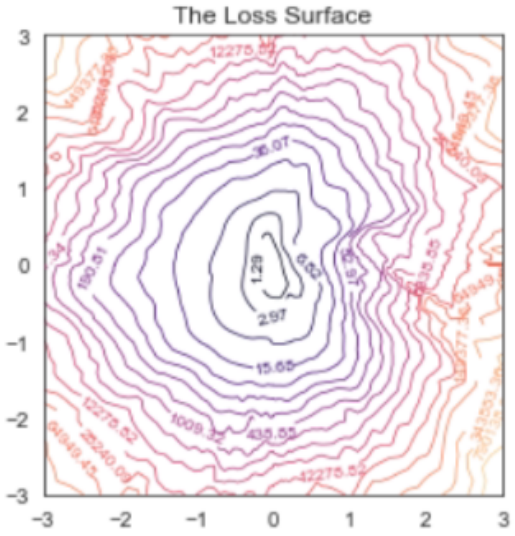

FIGURE 4.4: Visualisation of the loss landscape map for RE-ATT trained on 16-QAM Rayleigh flat fading dataset. Deeper colour indicates a steeper slope.

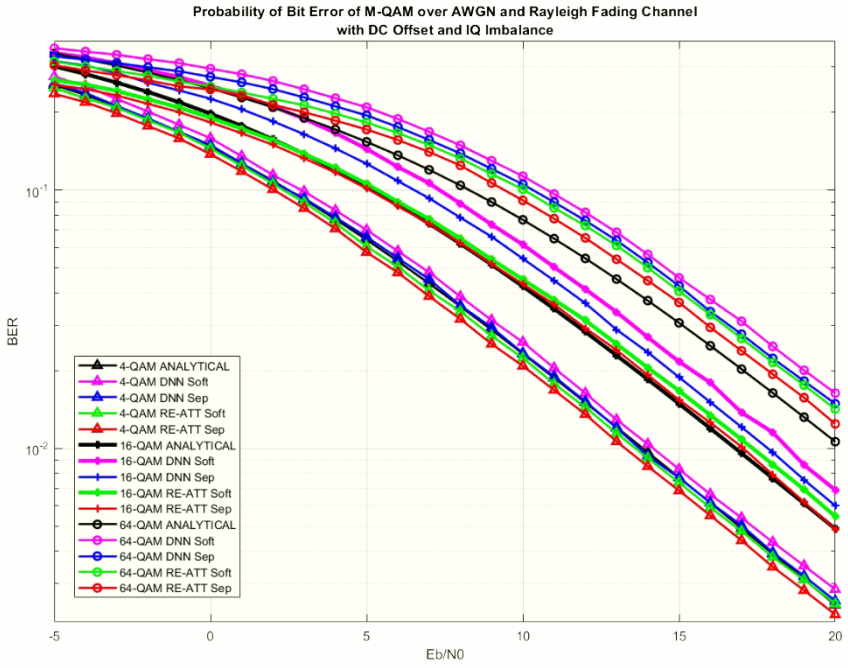

FIGURE 4.5: BER plot for performance comparison of proposed RE-ATT and control group DNN with/without separability metric loss regulariser over noisy, faded and distorted M-QAM signal. Different colours represent different demodulator baselines, where black for ANALYTICAL; magenta for DNN with SoftMax loss; blue for DNN with SoftMax loss and separability loss regulariser; cyan for proposed RE-ATT with SoftMax loss; and red for proposed RE-ATT with SoftMax loss and separability loss regulariser. Different markers represent different modulation order, where triangle marker indicates demodulators ran on 4-QAM, plus marker indicates 16-QAM and o-shaped marker indicates 64-QAM.

TABLE 4.1: The computational complexity, memory efficiency and symbol accuracy comparison of each RE-ATT model with different optimisation trained at 100 epoch on noisy, faded and distorted 16-QAM signal. Each model is evaluated on the total number of parameters, the total number of FLOPs, the approximated peak heap memory usage measured at mega bytes, the approximated data rate measured at kilo bits per second and the symbol accuracy obtained by each SNR by percentage.

RE-ATT Model and applied Optimisation	Total Params	Total FLOPs	Approx. Peak Mem. Usage (MB)	Avg.Data Rate (kb/s)	Symbol Acc(%)
with SoftMax and Adam	12,964	173,822	20	41.9	77.56
pruned, quantised with SoftMax and Adam	12,282	41,635	10	66.5	77.13
pruned, quantised with SoftMax and SGD-m	11,311	37,292	8	73.2	76.91
pruned, quantised with SoftMax and DEMON SGD-m	10,464	37,292	8	74.6	78.86
pruned, quantised with SoftMax + Regulariser and DEMON SGD-m	10,704	26,576	9	74.3	79.22

4.6 Summary

Application of pruning, quantisation model, optimiser with momentum decay and the proposed separability metric loss regulariser serve as a good combination to reduce the complexity and runtime of the proposed RE-ATT while improve classification accuracy by approximately 2%. The separability metric loss regulariser is proposed due to the characteristics of the signal have been studied and is found to have severely overlapping feature space, where the loss regulariser can help the deep learning model to learn discriminative features. The loss regulariser is also tested on a different classifier model to prove performance improvement can be achieved. The proposed loss regulariser prevails when runtime performance has a higher priority than memory usage.

Chapter 5

Deep Learning-based Demodulation: Two-Steps Approach

The advancement of DL in signal processing in recent years has posed many new ways to tackle the issue of demodulation that is discussed in the previous chapters. Various DL models have been explored in time series demodulation, where signal-to-symbol classification becomes a predominant method to demodulate signal regardless of the level of noise and distortions to which the signal is subjected, as observed from Table 2.3. However, this research has shown direct signal-to-symbol demodulation with classifiers suffers in BER performance due to low class separability and difficulty learning patterns from highly randomised data. However, it is demonstrated in this research that improvement on the issues as mentioned earlier is possible through a series of optimisations as detailed in Chapter 4; the optimisation methods are not a one-size-fits-all as they are tailored specifically for RE-ATT. There is a great interest in the search for a different approach, one such that with a DL-based modular process, where the task of demodulation is broken down into signal waveform reconstruction and symbol detection given a reconstructed clean signal. Therefore, a two-step method is proposed with a denoising step to reconstruct a clean signal waveform that eases symbol decision-making and a separate classifier to perform symbol detection. As DL has seen some successful applications in denoising data without strict requirements of accurate modelling of signal and

noise distribution [91, 92], it is theorise that employment of DL to reconstruct chaotic signals highly corrupted by channel effects and hardware impairments is very likely.

5.1 Signal Waveform Reconstruction with Autoencoder

The autoencoder (AE) is an NN architecture that consists of an encoder and a decoder, represented as $z = f(x)$ and $r = g(z)$, respectively, are usually stochastic functions $p_{encoder}(z|x)$ and $p_{decoder}(\hat{x}|z)$. An AE learns to approximate an identity function with backpropagation. The encoder $f(x)$ maps the input data x to a latent representation z, while the decoder $g(z)$ maps z back to the reconstruction of the input data r.

Geometrically, the primary objective of an AE is to acquire knowledge about the underlying manifold structure present in the input data and derive a parametric representation of the learned manifold. An AE is designed to learn an encoding map $\phi_\theta : X \to F$ and a decoding map $\psi_\theta : F \to X$ that provides a parametric representation of the input manifold Σ with the reconstructed manifold $\Sigma' = \psi_\theta \circ \phi_\theta(\Sigma)$ serving as an approximation to Σ. Considering a probability distribution μ in an ambient space X, the support of μ corresponds to a low-dimensional manifold $\Sigma \subset X$. In ML and DL, the ambient space typically refers to the higher-dimensional space containing data points or samples. In signal demodulation, an ambient space could represent the space of all symbols, where each symbol is a point in this higher-dimensional space. The ambient space is usually larger than the intrinsic or underlying structure of the data, such as a low-dimensional manifold. Still, it provides the space in which the data points can be represented and manipulated.

Moreover, the NN architecture comprising the AE also acquires knowledge and control over the distribution induced by the encoder $\phi_\theta \circ \mu$ defined in the latent space. The AE framework, through ϕ_θ, enables the learning and control of the distribution induced by $\psi_\theta \circ \phi_\theta(\Sigma)$ in the latent space. Therefore, by leveraging an AE within the context of signal demodulation, for instance, AE's capability can be exploited to learn the manifold structure and obtain a parametric representation of the underlying symbols. This facilitates the extraction of essential features and denoising of the received signal, thereby aiding in the recovery of the desired information or demodulation task.

Given a training set $S = x_i | x_i \in R^d, 1 \leq i \leq n$, an AE can be modelled by equation:

$$\begin{cases} z = f(w_e, b_e; x) \\ \hat{x} = g(w_d, b_d; z) \end{cases} \tag{5.1}$$

where w and b are the learnable parameters weight and bias term for either encoder or decoder. If $f(\cdot)$ and $g(\cdot)$ are implemented with NN, then w and b will be the weight matrices and the bias vectors concerning the encoder and decoder NN. In practice, since both $f(\cdot)$ and $g(\cdot)$ are parameterised with the parameter Θ, hence AE can also be described by:

$$\hat{x} = \mathcal{T}_\Theta(x) = f \circ g(x) \tag{5.2}$$

The network is trained to minimise the reconstruction error between the input data and the output of the decoder, such that:

$$min \sum_{i=1}^{n} \mathcal{L}(\Theta) + \lambda R(\Theta) \tag{5.3}$$

where λ is the regularisation parameter while $R(\Theta)$ is the regularisation function concerning the parameter set. To optimise the above problem, the SGD algorithm is typically used to optimise on Θ iteratively under the control of a loss function $\mathcal{L}$, typically inclines to be MSE as defined by:

$$\mathcal{L}(\Theta) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{x}_i)^2 \tag{5.4}$$

where $\theta = (w_e, b_b; w_d, b_d)$ while N is the number of samples.

The autoencoder architecture can be viewed as a type of unsupervised learning, as it learns to encode the input data in an efficient way without any explicit supervision [33]. While AE typically strives to produce an output that imitates the input in an unsupervised manner through minimisation of the reconstruction loss, it can also be trained in a supervised manner. This can be done with a given ground truth to obtain an output from a given noisy input, whereby the output is essentially a reconstructed version from the compressed feature.

Early AE was initially designed with a linear architecture, featuring a single hidden layer composed of linear activation units and no biases in both the encoder and decoder. This structural similarity to principal component analysis (PCA) models, widely utilised for dimensionality reduction and feature extraction, rendered AE valuable for similar purposes [93]. However, achieving satisfactory performance with more intricate tasks using linear AE proved challenging. As a result, several improvement strategies are typically employed to enhance the capabilities of AE [93–95]:

- Linearity-wise: Nonlinear activation functions are incorporated to augment the expressive power of AE.
- Layer-wise: Architectural variations within the encoder and decoder, such as convolutional layers and recurrent NN layers, or even stacking multiple blocks, enable the extraction of complex features.
- Depth-wise: Varying the depth of AE networks facilitates the extraction of deep features.
- Regularisation-wise: The inclusion of a regularisation term constrains the activation of hidden layer outputs, promoting robust learning.
- Loss-wise: Guiding the training process with specific task-oriented loss functions directs the AE model towards learning desired objectives.

While AE is initially proposed to achieve dimensionality reduction, it can be used for a variety of tasks. That includes dimensionality reduction, feature learning, information retrieval, denoising and anomaly detection [94]. AEs have also seen a rise of use in generative models for the generation of images, time series data, 3D textures, et cetera [91, 93]. Since then, AEs have become a widely used technique in DL for a multitude of applications. In the context of demodulation, prioritising denoising over generative tasks is both more pertinent and advantageous. Consequently, this research primarily emphasises on denoising using AE.

NN with nonlinearities will partition the input data feature space into piecewise linear regions. This partitioned feature space shapes the parametric manifold structure of data from which the DL model learns from. The parametric manifold structure obtained can be utilised in many ways, such as for the purpose of generation of a randomised

sample of Σ and denoising a noisy sample. Good understanding and efficient use of the manifold structure is the key to the success of a DL model in performing for a given task. Denoising of the signal can be viewed as a piecewise linear projection on the reconstruction manifold. Therefore, denoising is feasible with AE by reinterpreting the projection of a noisy sample onto a clean manifold representation geometrically, which corresponds to the nearest point on the manifold structure.

5.1.1 Denoising Autoencoder Demodulator Design Considerations

A denoising autoencoder (DAE) is a type of DL model that learns to approximate an identity function with backpropagation [96], mathematically represented as:

$$\hat{\mathbf{x}}_{\mathbf{IQ}} = f_{\text{dec}}(f_{\text{enc}}(\mathbf{x}_{\mathbf{IQ}}))\ \psi \tag{5.5}$$

Denoising autoencoder (DAE) is a stochastic extension to the classical AE. DAE is guided to learn the reconstruction of input data, given its noisy version [96]. The concept underlying DAE is rooted in the notion that an effective representation of data should encompass sufficient information to successfully reconstruct perturbed or corrupted variants of the original input. The general architecture of a DAE can be seen in Figure 5.1. DAE consists of three major parts: (i) an encoder f_{enc} that performs a compressed approximation of the input; (ii) the low dimensional representation output of the encoder; (iii) a decoder f_{dec} that upsamples the compressed representation. A reconstruction error loss, $\mathcal{L}$ is to be minimised by DAE through learning to achieve denoising.

In DAE, a corruption process such as dropout or randomised assignation of zero-value to the input is typically introduced. This technique aims to facilitate DAE to learn missing or corrupted values within randomly selected subsets of patterns. However, in the case of using DAE on signal demodulation and signal recovery, the process is slightly different. For a given noisy signal, the objective is to obtain its reconstructed version from the noisy, distorted and potentially compressed feature. A stochastic corruption process such as the AWGN and Rayleigh fading channel randomly perturbs the signal input, which forces DAE to predict the actual values of the randomly corrupted values

that contributes to the subsets of missing patterns. The process can be formulated as follows:

$$x\prime \;=\; H(x) + \mathbf{n} \tag{5.6}$$

where $\mathbf{n}$ represents the AWGN noise vector, the $H(\cdot)$ represents the Rayleigh fading channel.

Using a supervised learning method, the input signal will serve as the ground truth to denoise the corrupted input signal back to the clean version of the input. In short, DAE is trained to achieve denoising effects by minimising the reconstruction error between the clean input signal and the output of the reconstruction from the corrupted signal. Similar to AE, DAE can be defined as:

$$\hat{x} = \mathcal{T}_\Theta(x\prime) = f \circ g(x\prime) \tag{5.7}$$

In a classical DAE, feedforward NN is the core foundation of the hidden layers in the encoder and decoder. At the same time, the reconstruction error loss consists of an average difference between the estimated values and the actual value, such as MSE or L2 loss. While both the encoder and the decoder can be designed with varying depths and layers, ad convolutional architecture is designed specifically to the proposed method. Although feedforward NN can theoretically approximate any function by acting as a universal approximator, a carefully constructed convolutional NN is better suited to equip the model with the ability to perform the affine transformation. Convolution in the encoder serves to achieve a smoothing average effect on the input signal, while transposed convolution NN serves to reverse the distortions. The combination of the two functions is crucial for the correction of amplitude and phase distortions caused by noise, fading and DC offset. Apart from that, the properties of a convolution layer such as local connectivity and weight sharing will reduce the total number of parameters of the DL model and increase training and inference speed.

The proposed DAE follows the architecture listed in Table 5.1. n is the number of filters while k is the size of the kernel.

Table 5.1: DAE Architecture

Layer	**Parameters**
Encoder	
Min-max normalisation	N/A
Temporal convolution	filter: n^3, kernel: k, padding: True
1D Average pooling	pool size: 2, padding: True
Temporal convolution	filter: n^2, kernel: k, padding: True
1D Average pooling	pool size: 1, strides: 1, padding: True
Temporal convolution	filter: n, kernel: k, padding: True
1D Average pooling	pool size: 1, padding: False
Decoder	
Temporal deconvolution	Filter: n, kernel: k
Temporal deconvolution	Filter: n^2, kernel: k
Temporal deconvolution	Filter: n^3, kernel: k, padding: True
Temporal convolution	Filter: n, kernel: k, padding: True
Hyperbolic tangent	N/A

Such architecture incorporates convolution operations in the encoder instead of fully connected or feed-forward NN layers to extract features, and transposed convolution operations in the decoder to perform the features restoration. The normalisation layer rescales the signal values to between 0 to 1 as DL models may perceive negative values as negative weights, causing the model to disregard features learned on negative values. The convolution layers learn localised features of a given segment of signal, while also acting as a filter to smoothen the input signal. The pooling layers will further transform the signal by remapping the feature space. The deconvolution layers reconstruct the signal based on learned representations from the previous layer. Finally, the activation layer maps the values of the previous layer between -1 to 1, rescale by ψ, and outputs the reconstructed signal.

For a mono channel input, let k be the number of convolution kernel, and the i-th feature map can be formulated as:

$$h^i = act(x * W^i + b^i) \tag{5.8}$$

Weights W are shared among all the inputs which helps preserve local spatiality while biases b are broadcasted to the entire map. The asterik $*$ denotes the convolution process while *act* is the activation function. The reconstruction will be obtained as:

$$y = s(\sum_{i \in H} h^i * \tilde{W}^i + b) \tag{5.9}$$

where b is the bias per input channel or the latent map, H is the latent feature maps, and $\tilde{W}$ is the flip operation over W. Pooling and upsampling layers can be optionally added to the encoder and the decoder, respectively to contract and expand data.

Hyperbolic tangent activation function, *tanh*, is used as the activation function in the hidden layers instead of rectified linear unit (ReLU) which is typically used in convolutional layers. This is due to the nature of ReLU that cuts off all negative values will cause clipping to the signal with negative voltage which is undesirable for signal processing.

All convolution and pooling layers have a stride and dilation rate of 1. Glorot normal weight initialiser is used for all layers. SGD-m with decaying momentum as described in section 4.2 is used as it slowly reduces the gradient over time to prevent overfitting but it is possible to use other optimisers with suitable scheduling algorithms to obtain the same results.

In the output layer, a scale value ϵ is applied to the model outcome to ensure that the reconstructed signal has the correct amplitude. Hence, the output can be denoted as:

$$X' = tanh(\sum_{i \,\in\, C} c^i \; W'^i + B) * \epsilon \tag{5.10}$$

where B is the bias per input channel, C is group of latent feature maps, W' is flip operation over both weight dimensions.

The loss function is an integral part of a DAE as it guides DAE towards learning the appropriate compression of the input representation and appropriate transformations applied to the compressed representation to reconstruct the output waveform. This is to be done without any prior knowledge of original M-QAM constellation. Typically, a symmetric loss such as a quadratic symmetric loss like MSE or a linear symmetric loss like mean absolute error (MAE) will be used. However, MSE weighs large errors more

heavily than small ones as result of the squaring of each term, which causes bias towards outliers. While MAE does not have the same issue as MSE, MAE is not differentiable when the weights are 0. Therefore, even though MSE and MAE allows the encoder to be resistant to some perturbations of the input, both are unsuitable for guiding the model to reconstruct signals as close as possible to the ground truth.

To solve this problem, a loss term minimising the magnitude of the difference between the ground truth waveform and the reconstructed waveform in two-dimensional space using the Euclidean normalisation is proposed. By forcing the encoder to disregard changes in the input that are insignificant to the reconstruction of output signal by the decoder, the feature extraction will be less sensitive to small perturbations.

Therefore reconstruction loss for DAE is denoted as:

$$\mathcal{L}_{DAE} = \sqrt{\sum_{i=1}^{m} \|y_{true} - y_{pred}\|^2} \tag{5.11}$$

where y_{true} is the target or ground truth signal and y_{pred} is the predicted signal out by DAE.

Given a set of training inputs $x_1, x_2, ..., x_L$ where L is the size of the dataset. An input vector $x \in [-\infty, \infty]^d$ is first encoded into a hidden representation $z \in [-\infty, \infty]^{d'}$ via deterministic mapping, then decoded back into a $\hat{x}$ which is of same shape as x using similar mapping [96].

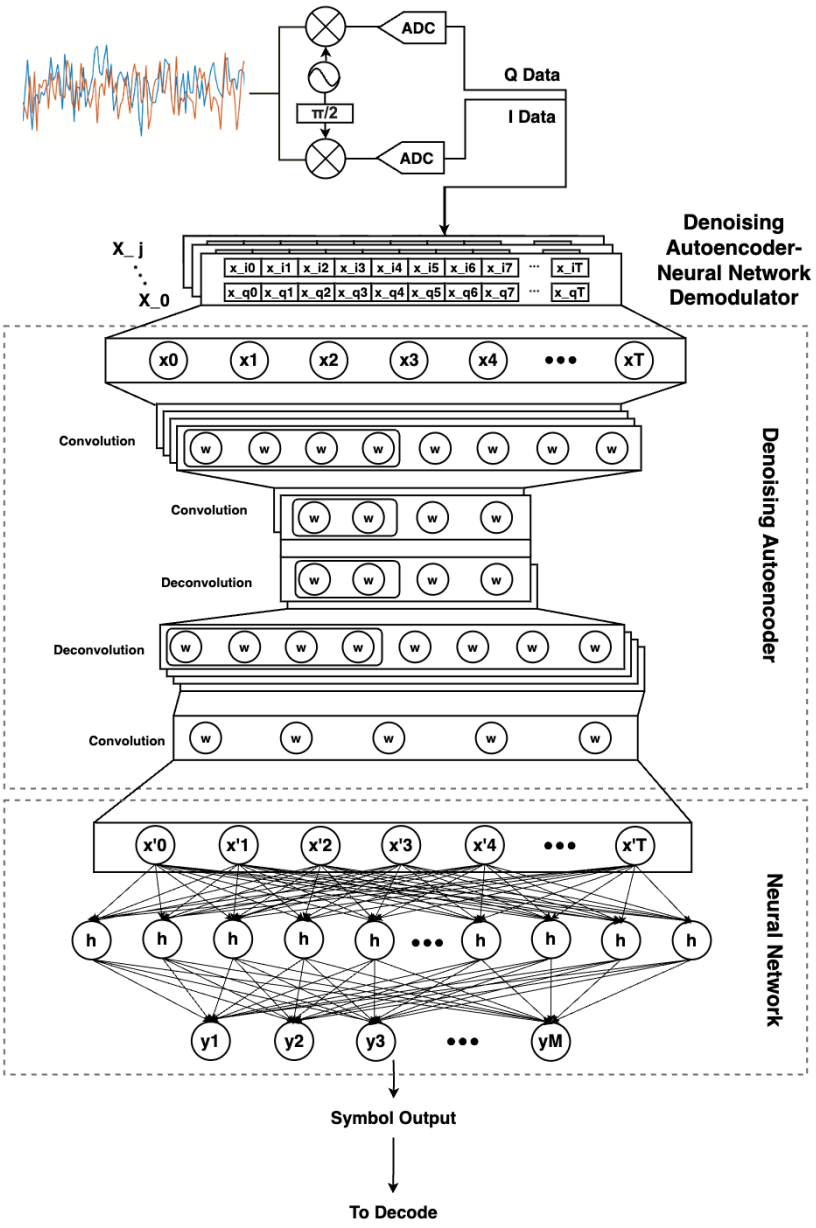

FIGURE 5.1: The block diagram of the receiver and the model architecture of the two-step demodulation that comprise of the proposed DAE with an NN classifier.

The proposed DAE consists of an encoder-decoder architecture with convolution and deconvolution operations of $O(nkdf)$ complexity where k is the window length of the kernel, d is the representation dimension and f is the number of filters in a convolution layer; while the NN consist of a single hidden layer that is $O(n)$ where n is the number of neurons. The output FC layer has $O(M)$ complexity. Hence the total complexity of the model is $O(5(nkdf) + M)$. The space complexity is $O(l)$ where l is the length of input sequence.

5.2 Experiment Setup for Denoising Autoencoder

Experiment setup for the DAE model shares many similarities to the direct signal-to-symbol demodulator RE-ATT setup.

5.2.1 Dataset

The same dataset in section 3.2.1 is utilised for the experiments of applying two-steps demodulation with the proposed DAE.

5.2.2 System Modelling

The two-steps demodulator has a similar system design as a direct signal-to-symbol demodulator described in section 3.2.2. The design also features a SISO system with a transmitter and a zero-IF receiver. The transmitter consists of M-QAM modulator, upsampler, filter, upconverter and normaliser, while the receiver consists of downconverter, filtering, downsampler and M-QAM demodulator. Differ from a direct signal-to-symbol demodulator, the two-steps demodulator integrates the proposed DAE and a classifier.

Given an input vector $x \in [-\infty, \infty]^d$ with 2 features I and Q, where d is the length symbols sequence, the proposed DAE encodes it to a hidden representation $c \in [-1, 1]^d$ via deterministic mapping. Latent representation c is then decoded back into a $\hat{x}$ which is the same shape as x using similar mapping [96]. x is finally reconstructed into $\hat{x} \in [-\psi, \psi]^d$ where ψ is the signal power level, approximating the original clean signal. A stochastic corruption process such as the AWGN and Rayleigh fading channel randomly

perturbs the input signal, forcing the proposed DAE to predict corrupted values for randomly selected subsets of missing patterns.

5.2.3 Implementation Details

Similar to the method in section 3.2.3, a uniform amount symbol class is generated followed by a shuffle to avoid class bias. A Monte Carlo simulation is performed to get an averaged performance. The system is modelled as illustrated in Figure 3.4 to generate the signal to be transmitted over AWGN noise under different SNR conditions and Rayleigh fading. The received signal is then impaired with DC offset, digitally sampled, and segmented into l input vector length with 2 channels, which are the I and Q, respectively. The dataset contains a total of 41 million symbols, from which 40% are used for training, 10% for validation, and the remaining 50% for testing and evaluation by measuring the BER.

The two-step demodulator formularised is with DAE and an NN classifier of choice. The proposed DAE is based on the concept of training a generative model with corrupted data in [97] where each layer of the DAE is trained to produce a hidden representation or higher level abstraction of the observed patterns based on the representation it receives previously by optimising a local unsupervised criterion. Yet, the proposed DAE is fully supervised where the clean signal is provided as a label along with the noisy and distorted signal as input, in which aims to reduce the difference between the label and the output. Consider a noisy and distorted observation x given a set of training inputs $x_1, x_2, ..., x_L$ with input features of x_I and x_Q, the implementation can be formularised as Algorithm 5.

Algorithm 5 Denoising Autoencoder

Require: input x, no. of kernels k, no.of layers L

procedure Encoder
initialize h_0, W_e, b_e, W_d, b_d
for l=1, ..., L **do**
    for j=1, ..., k **do**
        for i=1, ..., size(x) **do**
            Perform convolution $h_l = h_{l-1} + x_i * W_{ej} + b_e$
        end for
    end for
end for
$h_0 \leftarrow h_l$
procedure Decoder
for l=1, ..., L **do**
    for j=1, ..., k **do**
        for i=1, ..., size(x) **do**
            Perform deconvolution $h_l = h_{l-1} * W_{dj} + b_d$
        end for
    end for
end for
Perform convolution $y \leftarrow h_l * W + b$
return y

The goal of the proposed DAE is to minimise the reconstruction loss to encourage the model to transform the input feature through scaling, translation, and rotation to reverse the effects as visualised in Figure 5.3. This can be achieved by using the Euclidean distance between the true symbol and the predicted symbols. The reconstruction loss, a Euclidean normalisation [98] is given by:

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{L} \sum_{i=1}^{L} |\hat{\mathbf{x}}_i - \mathbf{x}_i|_2^2 \,, \tag{5.12}$$

where $|\cdot|_2^2$ represents the squared Euclidean distance between two vectors.

To obtain the symbol class, a simple NN can be used to map the reconstructed signals, $\hat{x}$ back to the symbols, y based on the learned relationships between the inputs and the outputs. This can be defined as:

$$\begin{aligned} f_{NN} : \hat{x} \rightarrow y \\ \text{where } f_{NN}(\hat{x}) = W\hat{x} + b \end{aligned} \tag{5.13}$$

The learning process involves adjusting the learned weights, W and bias term, b to minimise the difference between the predicted and actual outputs. This can be achieved using categorical cross-entropy loss function given by:

$$J(\theta) = -\frac{1}{L}\sum_{i=1}^{L}\sum_{j=1}^{M} y_{i,j}\log(\hat{y}_{i,j}). \tag{5.14}$$

$y_{i,j}$ is the ground-truth symbols for the i-th sample in the j-th class, while $\hat{y}_{i,j}$ is the predicted symbol of the i-th sample belonging to the j-th class. The symbols are then converted to bits to calculate the BER.

From previous results in Chapter 3 section 3.3, it can be observed that highly complex or deep architecture is not truly necessary. Simplistic and naive architectures have been shown to be just as competitive when constructed for the correct use case. Therefore for the proposed DAE model, both n and k are set at 2, $B = 128$, and $\xi = 100$. The optimiser is configured with an initial $\eta = 10^{-2}$, initial $\beta_{init} = 0.9$.

During offline training stage, each modulation order is trained separately under the same channel conditions and subjected to the same hardware impairments with the corresponding pilot symbols as the ground truth. K-Fold validation is done to prevent overfitting. During testing and inferencing, the ground truth pilot symbols are not required for the trained DL model. The proposed DAE is trained with 100 epochs. Figure 5.2 records the details of the the proposed DAE model's training configurations.

TABLE 5.2: The recommended configuration for the proposed denoising autoencoder model.

Configuration	Description
Optimiser	Stochastic gradient descent with momentum decay [77], no scheduling or learning decay is required.
Learning Rate	The optimal learning rate is the largest learning rate that does not cause divergence of the training criterion, set at 10^{-2}.
Weight Initialiser	Glorot/Xavier initialisation [78] with normal distribution used in convolution kernel, where the $mean = 0$ and $variance = \frac{1}{(input\_dimension + num\_of\_neurons)/2)}$
Weight Regulariser	None
Loss function	Euclidean Normalisation Loss

The models and benchmarks are implemented in Python using TensorFlow library.

5.2.4 Evaluation Metrics

The same evaluation metrics in section 3.2.4 is used to measure the performance of a two-steps demodulation with the proposed DAE and classifier.

The proposed DAE will also be compared to the following popular SOTA based on the research on previous works:

- LSTM-ResNet [67]
- U-Net [99]

A conventional receiver is also used as a baseline for performance comparison, characterised by an architecture consisting of DC compensator, blind IQ estimator, coherent demodulator with MLE detection as described in Chapter 2 section 2.4.2.

5.3 Experiment Results

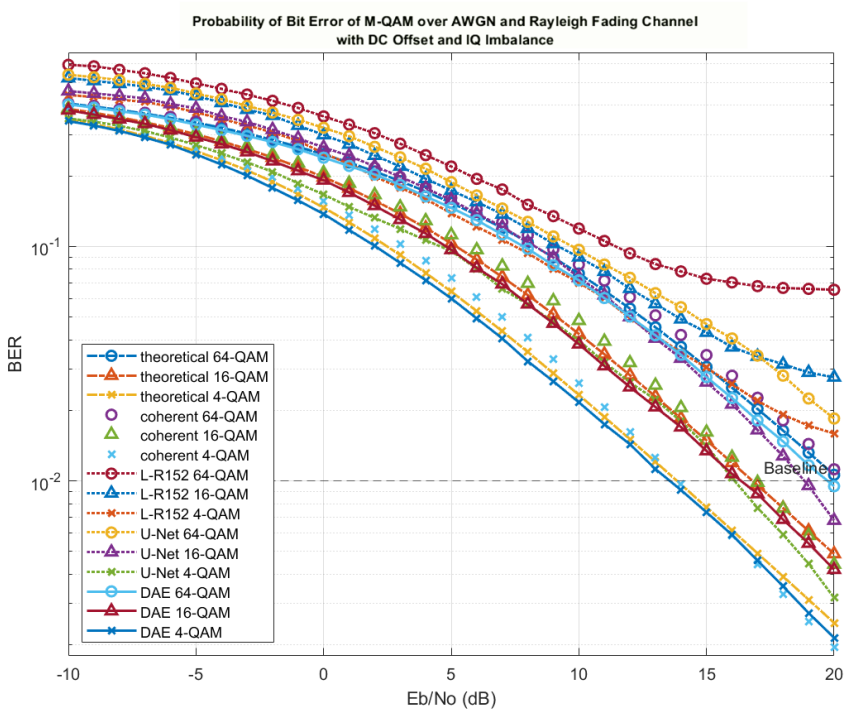

FIGURE 5.2: Performance comparison of the proposed DAE to existing methods on noisy and faded 4-QAM, 16-QAM and 64-QAM signals by BER.

The proposed DAE-NN is compared to three benchmarks, one conventional method with a coherent demodulator and two DL SOTA based on the LSTM Autoencoder and ResNet-152 and a typical U-Net. The proposed DAE-NN has a significant difference in performance compared to all other methods. In experience, randomised phase perturbation that occurs in faded signal is extremely difficult for any DL models to learn denoising and phase rotation reversal, therefore suffers in BER performance. The proposed DAE-NN demodulator has an averaged BER gain of 0.9 dB at 4-QAM, 1 dB at 16-QAM and 1 db at 64-QAM from the coherent demodulator. The mean accuracy of correctly detected symbols for 4-QAM, 16-QAM and 64-QAM are 86.93%, 91.19% and 92.47%, respectively. DAE-NN demodulator has a better performance than the coherent demodulator at lower Eb/No of 5 dB and below than higher Eb/No. The gain at 10^{-1} BER is consistently 1.1 dB throughout different modulation orders while the gain at 10^{-2} BER is 0.3 dB, 0.5 dB and 0.7 dB, respectively. From Figure 5.2, it can be observed that DAE-NN outperforms the benchmarks as well as the theoretical BER

TABLE 5.3: Performance analysis for each demodulator of various benchmark methods for AWGN and Rayleigh flat fading datasets with 4-QAM, 16-QAM and 64-QAM signals, respectively. For non-learning method- the ideal BER Theoretical and the Coherent receiver; for learning-based methods- L-R152, U-NET and DAE-NN (proposed). The evaluated metrics are the symbol test accuracy, the weighted average of precision and recall (F1) and the $Eb/N0$ at 10^{-2} BER. $Eb/N0$ should be the lower, the better.

Wireless Channel		n=AWGN, h=Rayleigh		
Attributes		Symbol Acc (%)	F1 Score	$Eb/N0$ @10^{-2}
4-QAM	Theoretical	92.55	N/A	14.1
	Coherent	92.62	0.926245	14.0
	L-R152	86.17	0.861703	> 20.0
	U-Net	92.33	0.923271	15.9
	DAE-NN	**92.47**	**0.924698**	**13.9**
16-QAM	Theoretical	89.73	N/A	16.7
	Coherent	89.27	0.892714	16.5
	L-R152	71.03	0.710296	> 20.0
	U-Net	78.04	0.780429	> 20.0
	DAE-NN	**90.19**	**0.901863**	**16.0**
64-QAM	Theoretical	86.32	N/A	20.3
	Coherent	86.11	0.861134	20.4
	L-R152	68.31	0.683131	> 20.0
	U-Net	71.57	0.715681	> 20.0
	DAE-NN	**86.93**	**0.869312**	**19.9**

which is the ideal case. DAE-NN is able to obtain a gain over the theoretical case of up to 0.8 dB.

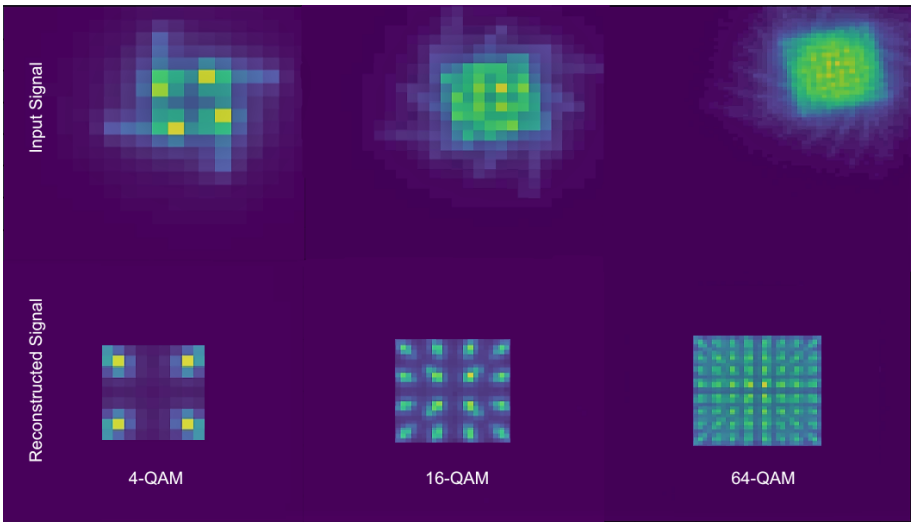

FIGURE 5.3: Visualization of the input signal (top) and reconstructed signal (bottom) at 15 dB E_b/N_o for 4-QAM, 16-QAM and 64-QAM.

5.4 Complexity vs Performance

The proposed model architectures DAE and RE-ATT combination has shown the most accuracy shown in Table 5.4 and BER performance gain across all modulation orders seen in Figure 5.4. This is followed by the proposed RE-ATT model in Chapter 3, simple DAE-NN described in section 5.2.3 and finally a vanilla NN for comparison. The addition of DAE model increases the total number of parameters by 270, total number of FLOPs by 2,881 and inferencing time by 140ms, for between 0.5dB to 2dB gain in BER performance across all SNR. The proposed DAE also has an approximated 162.8997 MIPS on NVIDIA GeForce RTX 2080 CUDA 11.7 and a peak memory usage of 580.6016 KB.

TABLE 5.4: The computational complexity, memory efficiency and symbol accuracy comparison of different approaches to DL-based demodulation trained at 100 epoch on noisy, faded and distorted 16-QAM signal. Each model is evaluated on the total number of parameters, the total number of FLOPs, the approximated peak heap memory usage measured at mega bytes, the approximated data rate measured at kilo bits per second and the symbol accuracy obtained by each SNR by percentage.

DL-Based Solution	Total Params	Total FLOPs	Approx. Peak Mem. Usage (MB)	Approx. Data Rate (kb/s)	Avg. Symbol Acc(%)
DNN classifier only	9,914	18,875	2	76.8	77.73
RE-ATT classifier only	10,704	26,576	9	74.3	79.22
DAE-NN	10,184	21,756	2	74.9	77.86
DAE-RE-ATT	10,974	29,457	9	72.2	83.94

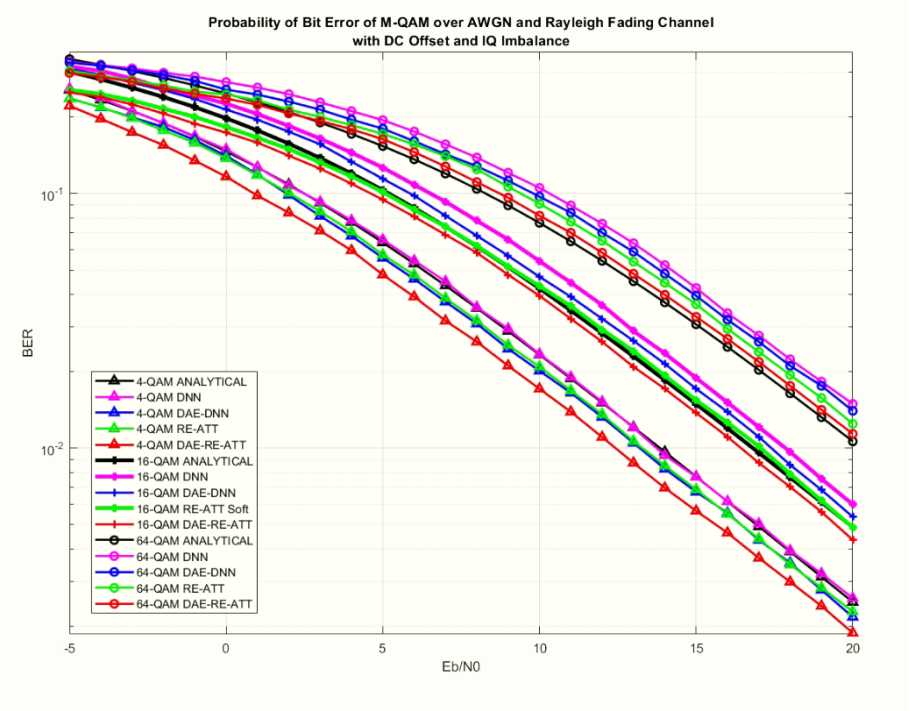

FIGURE 5.4: BER plot for performance comparison of proposed DAE applied to proposed classifier RE-ATT and control group classifier DNN versus the direct signal-to-symbol approach over noisy, faded and distorted M-QAM signal.

5.5 Ablation Study

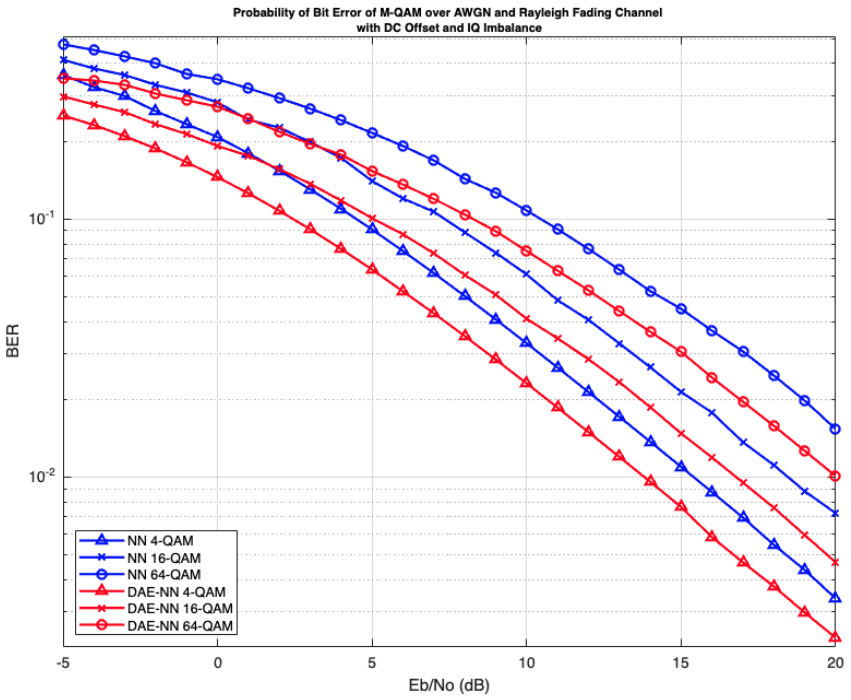

FIGURE 5.5: Ablation done on 4-QAM, 16-QAM and 64-QAM dataset with results where the log average bit error rate plot of a single hidden layer NN model (NN) and denoising autoencoder model with single hidden layer NN classifier (DAE-NN) at normalised SNR from -5dB to 20dB obtained after Monte-Carlo simulation.

An ablation study is conducted to further test the impact of DAE in a two-steps demodulation by comparing: a single hidden layer NN and DAE with NN. In such controlled environment, DAE-NN improves by an average of 2dB at 10^{-2} and 10^{-1} BER in comparison to NN for noisy and faded signal as seen in Figure 5.5. The proposed DAE does provide significant improvement by generating output signal that is less noisy which ease the classification for NN.

5.6 Summary

In this chapter, a DL-based method of signal recovery and demodulation for an imperfect M-QAM signal received by a DCR under a slow flat fading and noise channel in urban scenarios is proposed. It is shown that the proposed DAE specifically designed for handling complex-valued time series signal data with low samples per symbol, is able to reconstruct noisy and distorted signals back into their clean state to ease symbol

detection. The proposed DL-model is able to gain improvement in SNR and BER performance over a coherent demodulator with error compensation for DC offset which is common in DCR. As a modular and learning-based block, the DL-block can be trained on any modulation order, channel, frequency and samples per symbol to adapt to any existing architecture. When compared to existing models, the proposed DL model has linear computational complexity with better BER performance. Future directions of research include further fine-tuning, the introduction of other impairments as well as training on different modulation types.

Chapter 6

Conclusion

Narrowband radio frequency technologies are still preferred and relevant in today's world as it allows efficient communication with long battery life, long ranges transmission and better penetration through obstacles at a lower cost. However, narrowband systems, face many challenges that includes difficulty in achieving reliable radio communication due to environment and channel effects, as well as being bounded by the limitations of conventional demodulation techniques. Despite the maturity of traditional demodulators, deep learning application in narrowband communication systems offers a number of compelling benefits that drives this research, including better scalabilty, data-driven characteristics that gives adaptability and robustness, and offers a novel approach to addressing a longstanding issue.

This work considers quadrature amplitude modulation, a modulation in the bandwidth-limited region of the bandwidth efficiency plane in Chapter 2, which represents a large portion of high data rate applications in narrowband direct conversion receivers. Different types of advancements in deep learning architectures and the comparisons of demodulation performance between them is discussed in Chapter 2. This motivates the development of new deep learning-based demodulator architectures in this research.

The demodulation problem can be tackled in two distinct approaches, direct signal-to-symbol approach and modular approach. In Chapter 3, the focus is on defining a novel recurrent-attention networks demodulator architecture suitable for demodulation of noisy and faded quadrature amplitude modulation signals while demodulation is treated as a time series classification task. This composed of a recurrent layer to learn an

embedding of the signal and an attention layer to capture the important set of temporal features for classification. The result of this approach is compared to other state-of-the-arts deep learning-based demodulators from literature and conventional demodulator, where an averaged performance gain of 3dB across different modulation order is achieved. Next, optimisations and extensions to the recurrent-attention networks demodulator is investigated in Chapter 4, where lower complexity of up to 84% reduced floating point operations and improved accuracy of 2% is achieved through model compression, include a decaying momentum term in the optimiser, and regularisation. The other methods adopts a modular learning approach, whereby each chain of process (e.g. signal transformation, feature extraction, and feature classification) is treated as an individual subtask with different requirements and learning techniques. The naming convention "two-steps approach" reflects how this approach contains a denoising step to reconstruct the noisy and faded signal into a better representation, and a classifier learns to classify the symbols from the output of the denoiser. Further performance enhancement explores signal reconstruction using the denoising autoencoder techniques described in Chapter 5. A standalone denoising autoencoder is able to obtain an averaged performance gain of 0.8dB over benchmarks. When paired with a classifier recurrent-attention networks, the result of an averaged performance gain of 2dB across different modulation order compared to all other methods and an increase of 4.72% in accuracy compared to vanilla recurrent-attention networks.

The findings of this research demonstrate that the proposed deep learning-based demodulators effectively demodulate noisy and faded signals in narrowband systems without requiring additional transformation, filtering, or compensation of the raw time-domain signal. Both approaches consistently achieve superior bit-error performance across varying signal-to-noise ratios and modulation orders, outperforming other learning-based benchmarks which exhibit performance fluctuations under different channel conditions. Additionally, the proposed methods are shown to be efficient in terms of memory and computational complexity, as evidenced by comparative analysis with state-of-the-art benchmarks. The efficacy of these methods is further validated through comprehensive complexity and ablation studies.

Deep learning is expected to play pivotal role across various domains of data and signal processing, encompassing the dynamic landscape of radio frequency communications.

Recent developments in area of deep learning have brought in noteworthy progress concerning architectures, training methodologies, and overall performance. It is a data-centric alternative to traditional statistical signal processing used in conventional demodulation. In contrast to conventional demodulation techniques, deep learning possesses advantages including the capacity to assimilate both traditional and specialised solutions in digital signal processing, the robustness to support different modulation orders, and the ability to continuously refine and adapt its performance within real-world contexts. As research in deep learning advances, its influence on the evolution of demodulator designs is anticipated to be substantial in the future. The inherent versatility of deep learning models hold the promise of enhancing performance, inventing novel methods to solve issues and facilitating design reusability.

While the present study is primarily centred on the development of performance-complexity balanced learning-based demodulation techniques for bandwidth-limited systems, the aspiration is that this research will pave the way for a unified architecture capable of demodulation across different types of system, modulation techniques and channel environment.

6.1 Further Work

A notable limitation in the current research is its narrow focus on additive white Gaussian noise, flat fading, phase noise, and direct current offset. Future work should broaden the scope to consider additional signal impairments such as frequency offset and other distortions caused by strong close-in or far-out interference. Incorporating these factors is crucial for mitigating the effects of non-linear signal processing and communication challenges.

Although this study utilised artificially synthesised signal data modelled to real-world scenarios, it is imperative to recognise the potential differences between synthetic and real-world signal properties. For enhanced robustness and accuracy, further training of deep learning models should incorporate real channel data. Additionally, expanding the training dataset to include a wider variety of modulation types will enhance the models' adaptability and performance across diverse communication scenarios.

Another direction to focus on involves integrating the proposed deep learning demodulators into the physical layer of a direct conversion receiver. Developing a complete deep learning-based receiver capable of performing symbol detection and denoising functionalities as per the results and findings of this research will be a significant advancement. This integration will build on the current findings, potentially leading to a fully operational deep learning-based communication system with improved performance metrics.

Finally, the continuous evolution of deep learning technologies presents opportunities for exploring new architectures that promise enhanced performance with reduced computational complexity and improved training efficiency. One such architecture worth investigating is the Kolmogorov-Arnold Networks, which reportedly offer a 100-fold increase in parameter efficiency compared to typical feed-forward neural networks. Another architecture is the Spiking Neural Networks that mimic biological neurons more closely than other NNs, which can be suitable for real time and low power applications due to the event-driven behaviour of the neurons. Delving deeper into these advanced architectures could yield substantial improvements in the performance and practicality of deep learning applications in radio frequency communication systems.

Chapter 7

Appendix

7.0.1 Test RE-ATT with Non-linear Modulation Signal

Early in the research, we modelled Rayleigh channel to mimic wireless communication on a receiver that is moving away from the transmitter with vehicular speeds at 5 kmph, 50 kmph and 150 kmph to observe the performance of DL-based demodulators under different environmental conditions. The BER performance of different demodulators is as shown Figure 7.1, where the difference in BER performance between different fading speed is found to be negligible. No strong trend is observed from the results.

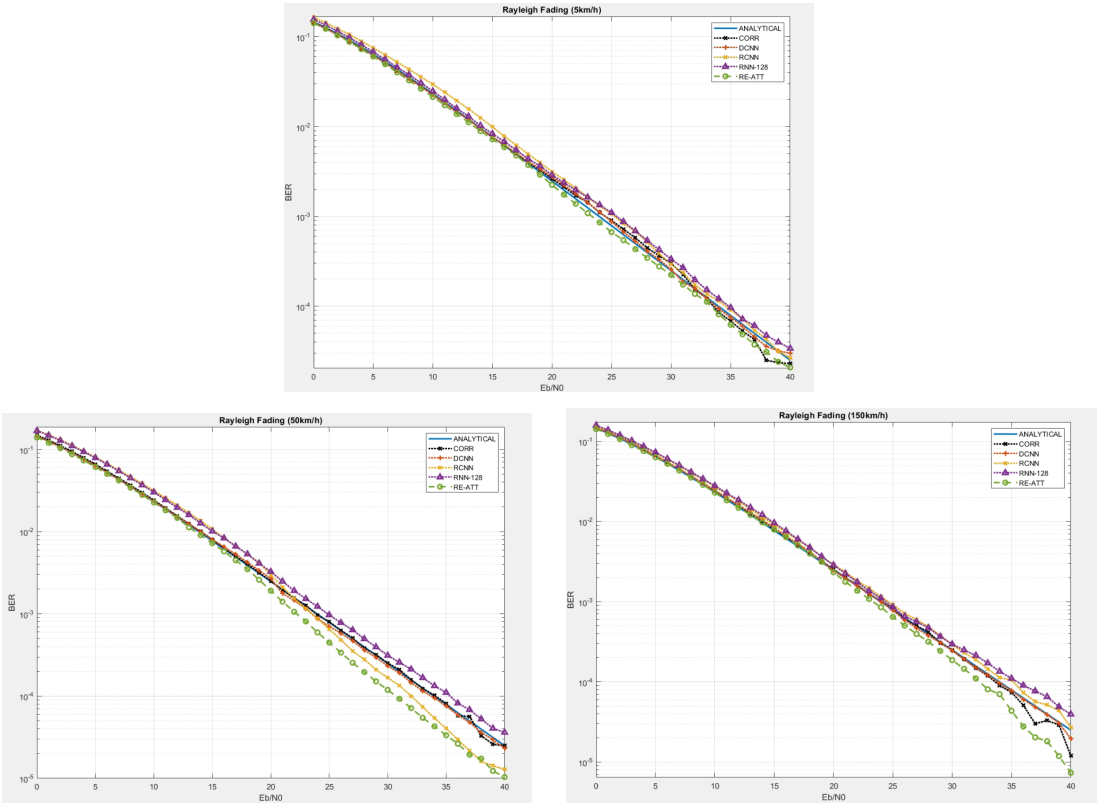

FIGURE 7.1: The BER plot of ANALYTICAL, a coherent demodulator (CORR), DCNN, hybrid CNN-RNN (RCNN), RNN-128 and RE-ATT at normalised SNR from 0dB to 40dB obtained after Monte-Carlo simulation on three different dataset with noisy and fading signal at velocity of (i) 5 kmph, (ii) 50 kmph, and (iii) 150 kmph for receiving 4-QAM signal.

7.0.2 Test RE-ATT with Non-linear Modulation Signal

Conventionally, a radio receiver will have a chain of separate processes to perform channel correction, noise reduction, filtering, decimation and offset compensation prior to symbol detection [100]. Each of the processes is viewed as individual problems to be resolved independently within their own building block. Due to the rigidity of separate block processes, signals of similar modulation type and/or at a different frequency band cannot share the same process chain. For instance, although both ASK and QAM are linear modulation schemes with double sidebands and require control of amplitude, they are unable to share the same demodulator block due to the different techniques required to detect an ASK or QAM signal [16]. Putting into data science perspective, two different modulation is as though two different domains.

Similarly, there are many works that already explored the application of DL into various types of modulation and different types of wireless channels. But in each case, the network is designed and tuned specifically for the problem at hand. This limits the impact of DL, as the effort of model design and manual fine-tuning needs to be repeated for each new task. Such characteristics would not provide DL-based solutions with a concrete edge over existing signal processing chains that utilise adaptive filters. The natural question arises: a DL model be trained to perform multi-task classification of similar nature yet vastly different in feature space?

The RE-ATT model that is successful with linear modulation scheme QAM is tested on a new dataset that differs in carrier frequency, modulation scheme and modulation order. Initial result has shown the same model that works on linear modulation signals is also able to work for non-linear modulation such as the binary FSK (BFSK), albeit the result being not ideal. Continued work is required to investigate whether a change of structure in the model architecture is necessary to achieve good feature extractions for classification or the performance could be improved with case-specific fine-tuning.

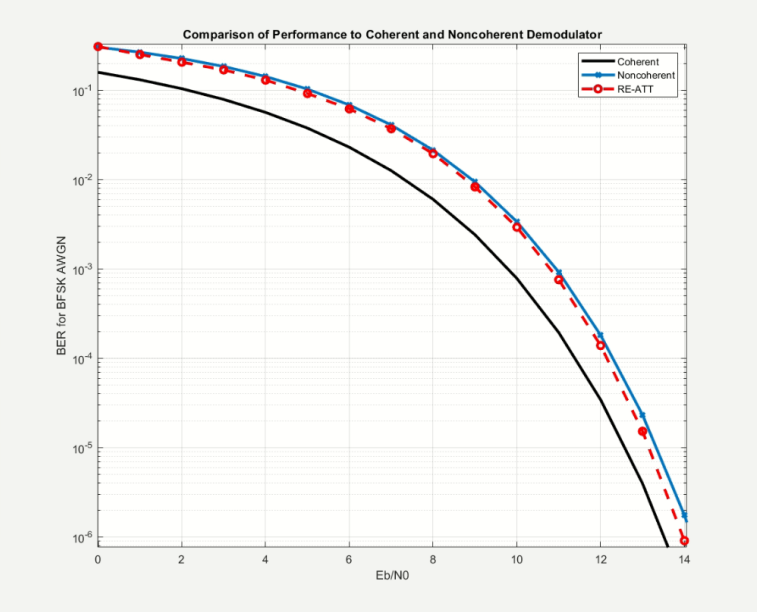

FIGURE 7.2: BER performance of RE-ATT model on BFSK signal at 400MHz carrier frequency with AWGN channel. The BER is plotted in comparison to a coherent and non-coherent demodulator.

7.0.3 Loss Function Selection for DAE

A crucial part that ensures the proposed DAE's ability to transform noisy input signal to match the clean signal is the loss function, where symmetrical loss such as MSE is typically used. In theory, minimising on MSE should be able to guide the model to learn to generate an output signal as close to the clean signal by minimising the distance between the two signals. However, symmetric loss gives same penalty for both positive and negative error. This is not ideal for dataset with many mismatched input-label pairs, where the received signal used as input is highly different from the original transmitted signal used as clean data due to channel impairments and hardware imperfections. There is a non-uniform perturbation of received signal due to the wireless channel has a normal distribution. This results in some symbol error to be higher than other symbols, creating an imbalance. Learning with nonnegative symmetric loss will cause the model to suffer underfitting and clip the reconstructed signals to between 0 and 1 rather than preserving the negative values which is crucial to form cyclic waves, this can be seen in Figure 7.4.

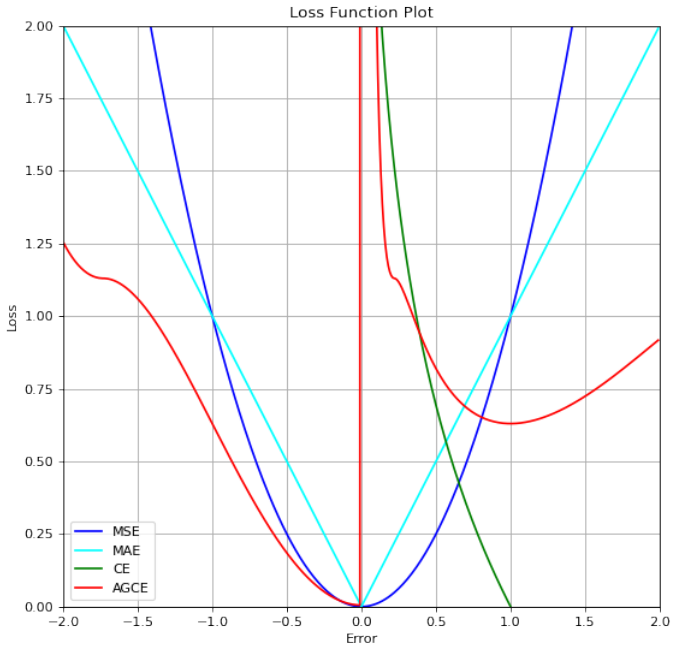

FIGURE 7.3: Loss function plot showing MSE, MAE, CE and proposed AGCE.

Hence, in our early research, we placed an emphasis on ensuring the loss function is positive with assumption that the probability of a correctly matched signal is higher than a wrongly matched pair. To achieve loss with positive bias, an asymmetric loss called asymmetric generalised cross-entropy (AGCE) loss modified from [101] and [102]

is applied tothe proposed DAE. AGCE is a negatively shifted box-cox transformation of cross-entropy (CE) and mean absolute error (MAE). AGCE has a threshold q to define the loss function by a combination of CE and MAE, and a hyperparameter α to change the shape of the symmetry. Due to it's asymmetric nature, AGCE is able to different penalty to both positive and negative error.The comparison of symmetrical and asymmetrical loss is shown in Figure 7.3.

An experiment is done to investigate the performance of the proposed DAE when trained with different loss functions. Figure 7.4 shows the visualisation of the output of each DAE trained with different loss function. The three commonly used loss for autoencoders: MSE, MAE and KL Divergence are unable to successfully reconstruction noisy signals. However, closer examination and consideration has led to the decision to discontinue research on this topic and search for alternatives. One key factor is although AGCE is also convex loss which allow learning with gradient-based optimisation, it is very difficult to train and to obtain a stable performance across different dataset. While it is possible to theorise from a high-level perspective that an asymmetric loss may benefit the model dealing with imbalance dataset by guiding the optimisation process to focus more on either positive or negative sample's gradients, it is unclear what positive or negative gradients actually represents. The use of AGCE requires meticulous fine-tuning of q without knowledge of whether to tune towards positive or negative bias. A loss with a threshold that is very sensitive to the distribution of data may not be suitable in practical application.

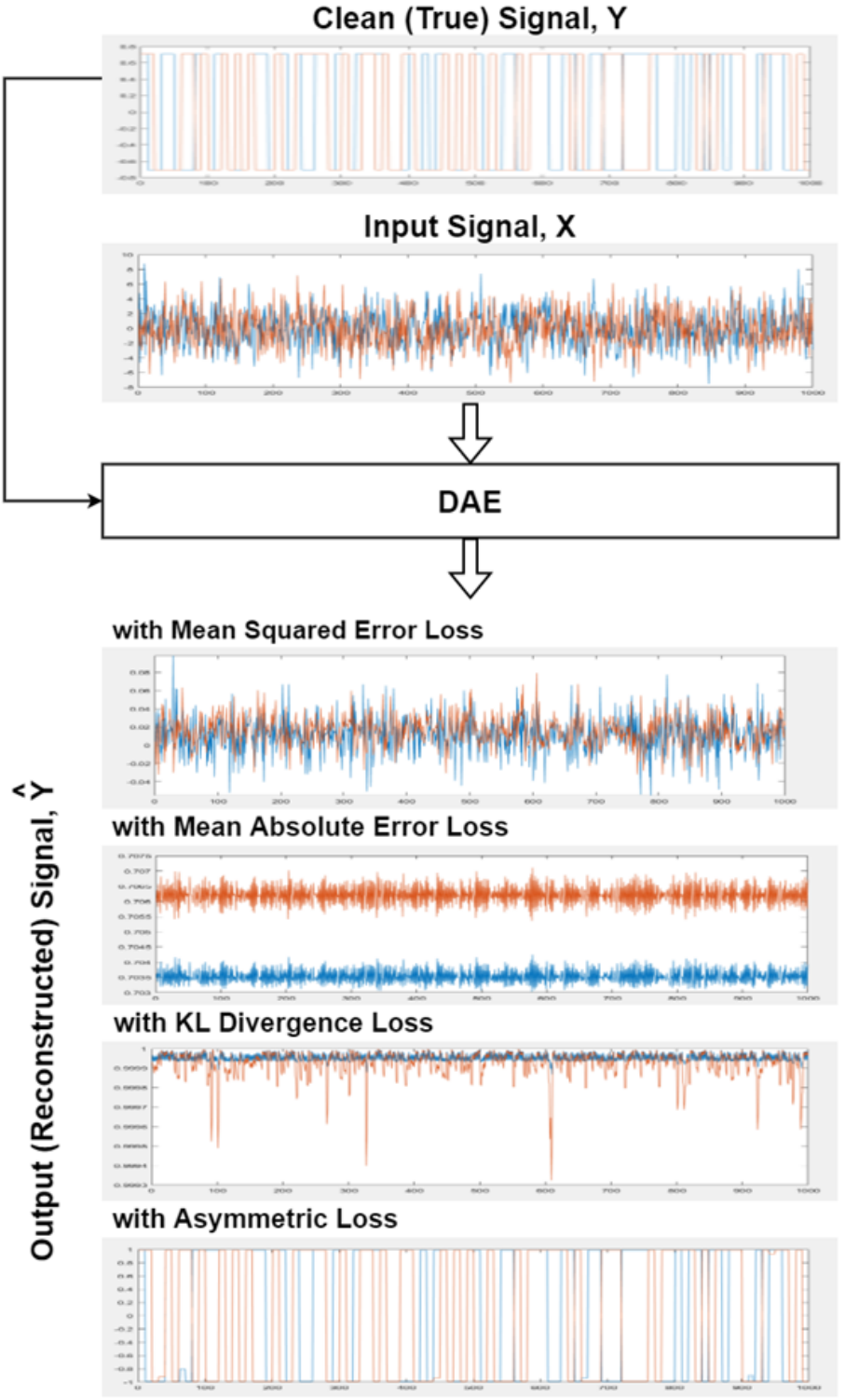

FIGURE 7.4: Visualisation of the output $\hat{Y}$ when different loss functions are applied to the proposed DAE. The proposed DAE is fed with sample input X from the same dataset described in Section 5.2.1 along with the true signal Y.

Bibliography

[1] B. Sklar. *Digital Communications: Fundamentals and Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988. ISBN 0132119390.

[2] T. S. Rappaport. *Wireless communications: Principles and practice*. Prentice Hall communications engineering and emerging technologies series. Prentice Hall, Inc., 2nd edition, 2002. ISBN 0130422320. Includes bibliographical references and index.

[3] B. Lindoff and P. Malm. Ber performance analysis of a direct conversion receiver. *IEEE Transactions on Communications*, 50(5):856–865, 2002. doi: 10.1109/TCOMM.2002.1006566.

[4] M. Faulkner. Dc offset and im2 removal in direct conversion receivers. *IEEE Proceedings - Communications*, 149(3):179–184, 2002.

[5] J. G. Proakis and D. G. Manolakis. *Digital Signal Processing (3rd Ed.): Principles, Algorithms, and Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996. ISBN 0133737624.

[6] J. C. Fletcher, J. W. Layland, W. L. Martin, A. I. Zygielbaum, R. M. Goldstein, and W. P. Hubbard. Digital demodulator-correlator, Sep 1978.

[7] J. T. J. Penttinen. *The Telecommunications Handbook: Engineering Guidelines for Fixed, Mobile and Satellite Systems*. Wiley, Giesecke Devrient, USA, 2015. ISBN 9781119944881.

[8] R. W. Stewart, K. W. Barlee, D. S. W. Atkinson, and L. H. Crockett. *Software Defined Radio Using MATLAB Simulink and the RTL-SDR*. Strathclyde Academic Media, Glasgow, GBR, 2015. ISBN 0992978726.

[9] J. Price and T. Goble. 10 - signals and noise. In Mazda F., editor, *Telecommunications Engineer's Reference Book*, pages 10–1–10–15. Butterworth-Heinemann, 1993. ISBN 978-0-7506-1162-6. doi: https://doi.org/10.1016/B978-0-7506-1162-6.50016-2. URL https://www.sciencedirect.com/science/article/pii/B9780750611626500162.

[10] P. A. Bradley and J. A. Lane. 9 - the ionosphere and the troposphere. In Mazda F., editor, *Telecommunications Engineer's Reference Book*, pages 9–1–9–19. Butterworth-Heinemann, 1993. ISBN 978-0-7506-1162-6. doi: https://doi.org/10.1016/B978-0-7506-1162-6.50015-0. URL https://www.sciencedirect.com/science/article/pii/B9780750611626500150.

[11] R. V. L. Hartley. Transmission of information. *The Bell System Technical Journal*, 7(3):535–563, 1928. doi: 10.1002/j.1538-7305.1928.tb01236.x.

[12] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x.

[13] H. Nyquist. Certain factors affecting telegraph speed. *The Bell System Technical Journal*, 3(2):324–346, 1924. doi: 10.1002/j.1538-7305.1924.tb01361.x.

[14] O. Jamin. *Broadband Direct RF Digitization Receivers*. Springer International Publishing, Switzerland, 2014. ISBN 9783319011493.

[15] Y. Akaiwa. *Introduction to Digital Mobile Communication*. Wiley, Hoboken, New Jersey, 2015. ISBN 9781119041108.

[16] A. V. Raisanen and Arto Lehto. *Radio Engineering for Wireless Communication and Sensor Applications*. Artech House, Inc., Norwood, MA, USA, 2003. ISBN 1580535429.

[17] W. T. A. Lopes, W. J. L. Queiroz, F. Madeiro, and M. S. Alencar. Exact bit error probability of m-qam modulation over flat rayleigh fading channels. In *2007 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference*, pages 804–806, 2007.

[18] W. Tian. Cnn and rnn-based deep learning methods for digital signal demodulation. In *Proceedings of the 2019 International Conference on Image, Video and Signal Processing*, IVSP 2019, page 122–127, New York, NY, USA, 2019. Association

for Computing Machinery. ISBN 9781450361750. doi: 10.1145/3317640.3317656. URL https://doi-org/10.1145/3317640.3317656.

[19] J. E. Flood. 19 - digital modulation. In Mazda F., editor, *Telecommunications Engineer's Reference Book*, pages 19–1–19–22. Butterworth-Heinemann, 1993. ISBN 978-0-7506-1162-6. doi: https://doi.org/10.1016/B978-0-7506-1162-6.50025-3. URL https://www.sciencedirect.com/science/article/pii/B9780750611626500253.

[20] X. Chen and L. Wu. Nonlinear demodulation and channel coding in ebpsk scheme. *The Scientific World Journal*, 2012, 2012.

[21] H. Wang, Z. Wu, S. Ma, S. Lu, H. Zhang, G. Ding, and S. Li. Deep learning for signal demodulation in physical layer wireless communications: Prototype platform, open dataset, and analytics. *IEEE Access*, 7:30792–30801, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2903130.

[22] P. Siyari, H. Rahbari, and M. Krunz. Lightweight machine learning for efficient frequency-offset-aware demodulation. *IEEE Journal on Selected Areas in Communications*, 37(11):2544–2558, 2019. doi: 10.1109/JSAC.2019.2933956.

[23] T. J. 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, 2017. doi: 10.1109/TCCN.2017.2758370.

[24] J. He, C. Xu, C. Yin, and P. Li. Modulation recognition of communication signals based on deep learning. In *2021 IEEE 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, volume 2, pages 527–531, 2021. doi: 10.1109/ICIBA52610.2021.9687989.

[25] X. Zhang, T. Li, P. Gong, R. Liu, X. Zha, and W. Tang. Open set recognition of communication signal modulation based on deep learning. *IEEE Communications Letters*, 26(7):1588–1592, 2022. doi: 10.1109/LCOMM.2022.3174035.

[26] H. Guo, S. Wu, H. Wang, and M. Daneshmand. Dsic: Deep learning based self-interference cancellation for in-band full duplex wireless. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019. doi: 10.1109/GLOBECOM38437.2019.9013521.

[27] Y. Ghanney and W. Ajib. Radio frequency interference detection using deep learning. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5, 2020. doi: 10.1109/VTC2020-Spring48590.2020.9129612.

[28] Y. Kurzo, A. T. Kristensen, A. Burg, and A. Balatsoukas-Stimming. Hardware implementation of neural self-interference cancellation. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 10(2):204–216, 2020. doi: 10.1109/JETCAS.2020.2992370.

[29] Y. Zhou, A. Samiee, T. Zhou, and B. Jalali. Deep learning interference cancellation in wireless networks, 2020.

[30] T. S. Rappaport. The wireless revolution. *IEEE Communications Magazine*, 29: 52–71, 1991.

[31] D. P. Bouras, P. T. Mathiopoulos, and D. Makrakis. Neural-net based receiver structures for single- and multi-amplitude signals in interference channels. *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, pages 535–544, 1994.

[32] K. Ohnishi and K. Nakayama. A neural demodulator for quadrature amplitude modulation signals. *Proceedings of International Conference on Neural Networks (ICNN)*, 4:1933–1938, 1996.

[33] I. Goodfellow, Y Bengio, and A. Courville. *Deep Learning*. The MIT Press, 2016. ISBN 0262035618, 9780262035613.

[34] I. Goodfellow, Y Bengio, and A. Courville. *Deep Learning*. The MIT Press, 2016. ISBN 0262035618, 9780262035613.

[35] K. Dejhan . Lerkvaranyu and Y. Miyanaga. M-qam demodulation in an ofdm system with rbf neural network. In *The 2004 47th Midwest Symposium on Circuits and Systems, 2004. MWSCAS '04.*, volume 2, pages II–II, 2004. doi: 10.1109/MWSCAS.2004.1354225.

[36] M. Aziz M. Ahmed Nasraden Milad and Rahmadwati. Neural network demodulator for quadrature amplitude modulation (qam). In *International Journal of advanced studies in Computer Science and Engineering (IJASCSE)*, volume 5, 2016.

[37] A. Akan M. Onder and H. Dogan. Advanced neural network receiver design to combat multiple channel impairments. *Turkish Journal of Electrical Engineering and Computer Sciences*, 24:3066–3077, 2016.

[38] Y. Zhang, J. Li, Y. Zakharov, X. Li, and J. Li. Deep learning based underwater acoustic ofdm communications. *Applied Acoustics*, 154:53–58, 2019. ISSN 0003-682X. doi: https://doi.org/10.1016/j.apacoust.2019.04.023. URL https://www.sciencedirect.com/science/article/pii/S0003682X18307400.

[39] N. Daldal, M. Nour, and K. Polat. A novel demodulation structure for quadrate modulation signals using the segmentary neural network modelling. *Applied Acoustics*, 164:107251, 2020. ISSN 0003-682X. doi: https://doi.org/10.1016/j.apacoust.2020.107251. URL https://www.sciencedirect.com/science/article/pii/S0003682X20300025.

[40] M. Kozlenko, I. Lazarovych, V. Tkachuk Valerii, and M. Kuz. Deep learning demodulation of amplitude noise shift keying spread spectrum signals. In *2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC ST)*, pages 717–720, 2020. doi: 10.1109/PICST51311.2020.9468063.

[41] W. Wang Q. Wang, R. Liu and G. Xie. An electrocommunication system using fsk modulation and deep learning based demodulation for underwater robots, 2020.

[42] A. Ahmad, S. Agarwal, S. Darshi, and S. Chakravarty. Deepdemod: Bpsk demodulation using deep learning over software-defined radio. *IEEE Access*, 10: 115833–115848, 2022. doi: 10.1109/ACCESS.2022.3219090.

[43] L. Fang and L. Wu. Deep learning detection method for signal demodulation in short range multipath channel. In *2017 IEEE 2nd International Conference on Opto-Electronic Information Processing (ICOIP)*, pages 16–20, 2017. doi: 10.1109/OPTIP.2017.8030690.

[44] S. Muthulakshmi and R. Jose. Signal demodulation without channel equalizer using machine learning techniques. In *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, volume 1, pages 1247–1252, 2019. doi: 10.1109/ICICICT46008.2019.8993353.

[45] Q. Miao, Y. Zhang, and X. Zhang. Intelligent demodulation method for communication signals based on multi-layer deep belief network. In *2019 International*

Conference on Electronic Engineering and Informatics (EEI), pages 131–134, 2019. doi: 10.1109/EEI48997.2019.00036.

[46] X. Liu N. Zeng Y. Liu W. Liu, Z. Wang and F.E. Alsaadid. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, April 2017. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2016.12.038.

[47] S. W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, San Diego, CA, USA, 1997. ISBN 0-9660176-3-3.

[48] W. Hu Y. Li X. Zhou X. Lin, R. Liu and X. He. A deep convolutional network demodulator for mixed signals with different modulation types. In *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, pages 893–896, Orlando, FL, USA, 2017. IEEE.

[49] F. James A.S. Mohammad, N. Reddy and C. Beard. Demodulation of faded wireless signals using deep convolutional neural networks. In *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 969–975, Las Vegas, NV, USA, Jan 2018. IEEE. doi: 10.1109/CCWC.2018.8301731.

[50] V. Tkachuk M. Kozlenko, I. Lazarovych and V. Vialkova. Software demodulation of weak radio signals using convolutional neural network. In *2020 IEEE 7th International Conference on Energy Smart Systems (ESS)*, pages 339–342, Kyiv, Ukraine, 2020. IEEE. doi: 10.1109/ESS50319.2020.9160035.

[51] K. Dakic, B. Al Homssi, A. Al-Hourani, and M. Lech. Lora signal demodulation using deep learning, a time-domain approach. In *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pages 1–6, 2021. doi: 10.1109/VTC2021-Spring51267.2021.9448711.

[52] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[53] D. Bahdanau K. Cho, B. Merrienboer and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches, 2014.

[54] M. Li, H. Zhong, and M. Li. Neural network demodulator for frequency shift keying. In *2008 International Conference on Computer Science and Software Engineering*, volume 4, pages 843–846, 2008.

[55] M. Moghadasi M. R. Amini and I. Fatehi. A bfsk neural network demodulator with fast training hints. In *2010 Second International Conference on Communication Software and Networks*, pages 578–582, 2010.

[56] S. Kalade, L. Crockett, and R. W. Stewart. Using sequence to sequence learning for digital bpsk and qpsk demodulation. In *2018 IEEE 5G World Forum (5GWF)*, pages 317–320, Silicon Valley, CA, USA, 2018. IEEE.

[57] D. Elbaz and M. Zibulevsky. End to end deep neural network frequency demodulation of speech signals. In Supriya Kapoor Kohei Arai and Rahul Bhatia, editors, *Advances in Information and Communication Networks*, pages 1–11, Cham, 2019. Springer International Publishing. ISBN 978-3-030-03402-3.

[58] I. Hatai and I. Chakrabarti. A new high-performance digital fm modulator and demodulator for software-defined radio and its fpga implementation. *International Journal of Reconfigurable Computing*, 2011, January 2011. ISSN 1687-7195. doi: 10.1155/2011/342532. URL https://doi.org/10.1155/2011/342532.

[59] N. Daldal, A. Sengur, K. Polat, and Z. Cömert. A novel demodulation system for base band digital modulation signals based on the deep long short-term memory model. *Applied Acoustics*, 166:107346, 2020. ISSN 0003-682X. doi: https://doi.org/10.1016/j.apacoust.2020.107346. URL https://www.sciencedirect.com/science/article/pii/S0003682X2030222X.

[60] Y. Jiang L. Zhang, H. Zhang and Z. Wu. Intelligent and reliable deep learning lstm neural networks-based ofdm-dcsk demodulation design. *IEEE Transactions on Vehicular Technology*, 69(12):16163–16167, 2020. doi: 10.1109/TVT.2020.3022043.

[61] M. Fan and L. Wu. Demodulator based on deep belief networks in communication system. In *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, pages 1–5, Khartoum, Sudan, 2017. IEEE.

[62] Y. Xie, K. C. Teh, and A. C. Kot. Deep learning-based joint detection for ofdm-noma scheme. *IEEE Communications Letters*, 25(8):2609–2613, 2021. doi: 10.1109/LCOMM.2021.3077878.

[63] S. Meng, X. Yang, J. Liu, and W. Liu. Demodulation of unitary space-time modulated signals based on deep learning. In *2023 IEEE 6th International Conference on Electronic Information and Communication Technology (ICEICT)*, pages 33–37, 2023. doi: 10.1109/ICEICT57916.2023.10245529.

[64] F. Etemadi, P. Heydari, and H. Jafarkhani. On analog qam demodulation for millimeter-wave communications. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 66(3):402–406, 2019. doi: 10.1109/TCSII.2018.2852687.

[65] M. Shah and S. Gupta. Baseband i/q regeneration method for direct conversion receiver to nullify effect of second order intermodulation distortion. In *2017 IEEE MTT-S International Microwave and RF Conference (IMaRC)*, pages 259–262, 2017. doi: 10.1109/IMaRC.2017.8611006.

[66] Y. Zhang, C. Li, H. Wang, J. Wang, F. Yang, and F. Meriaudeau. Deep learning aided ofdm receiver for underwater acoustic communications. *Applied Acoustics*, 187:108515, 2022. ISSN 0003-682X. doi: https://doi.org/10.1016/j.apacoust.2021.108515. URL https://www.sciencedirect.com/science/article/pii/S0003682X21006095.

[67] H. Arab, I. Ghaffari, R. M. Evina, S. O. Tatu, and S. Dufour. A hybrid lstm-resnet deep neural network for noise reduction and classification of v-band receiver signals. *IEEE Access*, 10:14797–14806, 2022. doi: 10.1109/ACCESS.2022.3147980.

[68] R. Pascanu, T. Mikolov, and Y. 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.

[69] K. Cho D. Bahdanau and Y. Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[70] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. K., and I. Polosukhin. Attention is all you need. In *Advances*

in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[71] K. Cho J. Chung, C. Gulcehre and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[72] Y. Fujiwara S. Kanai and S. Iwamura. Preventing gradient explosions in gated recurrent units. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 435–444, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[73] K. S. Fan Y. S. Shun and Y. L. Hung. Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, 108(8), 2019. ISSN 1573-0565. doi: https://doi.org/10.1007/s10994-019-05815-0.

[74] Z. Yang, D. Yang, C. Dyer, A. J. Smola X. He, and E. H. Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, San Diego, California, 2016. Association for Computational Linguistics. URL http://aclweb.org/anthology/N/N16/N16-1174.pdf.

[75] Y. Zhou, L. Liao, Y. Gao, H. Huang, and X. Wei. A discriminative convolutional neural network with context-aware attention. *ACM Trans. Intell. Syst. Technol.*, 11(5), July 2020. ISSN 2157-6904. doi: 10.1145/3397464. URL https://doi-org/10.1145/3397464.

[76] S. Joty J. Ma, W. Gao and K. Wong. An attention-based rumor detection model with tree-structured recursive neural networks. *ACM Trans. Intell. Syst. Technol.*, 11(4), June 2020. ISSN 2157-6904. doi: 10.1145/3391250. URL https://doi-org/10.1145/3391250.

[77] J. Chen and A. Kyrillidis. Decaying momentum helps neural network training. *CoRR*, abs/1910.04952, 2019. URL http://arxiv.org/abs/1910.04952.

[78] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors,

Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL http://proceedings.mlr.press/v9/glorot10a.html.

[79] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. doi: https://doi.org/10.1111/j.2517-6161.1996.tb02080.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1996.tb02080.x.

[80] A. E. Hoerl and R. W. Kennard. Ridge regression: Applications to nonorthogonal problems. *Technometrics*, 12(1):69–82, 1970. doi: 10.1080/00401706.1970.10488635. URL https://www.tandfonline.com/doi/abs/10.1080/00401706.1970.10488635.

[81] S. Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural network. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/ae0eb3eed39d2bcef4622b2499a05fe6-Paper.pdf.

[82] A. See, M. Luong, and C. D. Manning. Compression of neural machine translation models via pruning. In *Conference on Computational Natural Language Learning*, 2016. URL https://api.semanticscholar.org/CorpusID:2973141.

[83] S. Narang, G. F. Diamos, S. Sengupta, and E. Elsen. Exploring sparsity in recurrent neural networks. *ArXiv*, abs/1704.05119, 2017. URL https://api.semanticscholar.org/CorpusID:10135357.

[84] M. Zhu and S. Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression.

[85] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

[86] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.

[87] J. Chen, C. Wolfe, Z. Li, and A. Kyrillidis. Demon: Improved neural network training with momentum decay, 2021.

[88] Q. Ce and S. Fei. Contrastive-center loss for deep neural networks. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2851–2855, 2017. doi: 10.1109/ICIP.2017.8296803.

[89] H. Peng and S. Yu. Beyond softmax loss: Intra-concentration and inter-separability loss for classification. *Neurocomputing*, 438:155–164, 2021. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2020.11.030. URL https://www.sciencedirect.com/science/article/pii/S0925231220318178.

[90] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[91] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S. Chang, and T. Sainath. Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 13(2):206–219, 2019. doi: 10.1109/JSTSP.2019.2908700.

[92] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C. Lin. Deep learning on image denoising: An overview. *Neural Networks*, 131:251–275, 2020. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2020.07.025. URL https://www.sciencedirect.com/science/article/pii/S0893608020302665.

[93] D. Bank, N. Koenigstein, and R. Giryes. Autoencoders. *Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook*, pages 353–374, 2023.

[94] P. Baldi. Autoencoders, unsupervised learning, and deep architectures. In Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham Taylor, and Daniel Silver, editors, *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pages 37–49, Bellevue, Washington, USA, 02 Jul 2012. PMLR. URL https://proceedings.mlr.press/v27/baldi12a.html.

[95] J. Zhai, S. Zhang, J. Chen, and Q. He. Autoencoder and its various variants. In *2018 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 415–419. IEEE, 2018.

[96] Y. Bengio P. Vincent, H. Larochelle and P-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[97] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390294. URL https://doi.org/10.1145/1390156.1390294.

[98] T. Georgiou, S. Schmitt, T. Back, W. Chen, and M. Lew. Norm loss: An efficient yet effective regularization method for deep neural networks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 8812–8818, Los Alamitos, CA, USA, jan 2021. IEEE Computer Society. doi: 10.1109/ICPR48806.2021.9412171. URL https://doi.ieeecomputersociety.org/10.1109/ICPR48806.2021.9412171.

[99] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

[100] W. Leonard, A. Saunders, M. Calabro, and K. Olsen. A multi-waveform radio receiver, an example of machine learning enabled radio architecture and design. In *MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM)*, pages 634–639, 2019. doi: 10.1109/MILCOM47813.2019.9020991.

[101] Z. Zhang and M. R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 8792–8802, Red Hook, NY, USA, 2018. Curran Associates Inc.

[102] X. Zhou, X. Liu, J. Jiang, X. Gao, and X. Ji. Asymmetric loss functions for learning with noisy labels. *ArXiv*, abs/2106.03110, 2021.