Deep Learning Techniques for Next Generation Wireless Networks

A thesis submitted to The University of Manchester for the degree of Doctor of Philosophy in the Faculty of Science and Engineering.

2023

By

Mohamad Abdulaziz Al Awad

Department of Electrical and Electronic Engineering

Contents

Word Count: 29421

List of Tables

List of Figures

Abstract

Machine learning (ML) techniques have shown promising performance in solving different communication system issues. Recently, several deep learning-based end-to-end techniques have been implemented to optimize the transmitter, the channel, and the receiver blocks in one single process, thereby replacing the conventional communications system. In this thesis, we start exploring the research for the end-to-end wireless model where we used the autoencoder (AE) as a based communication system. We studied the performance of the AE with additive white Gaussian noise (AWGN) channel and compared it with the equivalent conventional communication system. Then for deep learning (DL), we introduce a DL-based detector, termed DL Index Modulation (DLIM), for IM-MIMO-OFDM using a deep neural network (DNN) in terms of error performance. Our initial results using the proposed DLIM can detect the transmitted symbols with performance comparable to near-optimal bit error rate (BER) with shorter runtime than the existing hand-crafted detectors. Next, for the Intelligent Reflecting Surfaces (IRS), we proposed an IRS-assisted end-to-end communication system that operates over AWGN channels, where the modulation and demodulation are performed by a DNN based on an AE architecture. Simulation results show the BER performance of our AE-based scheme achieves better performance gains than the existing classical IRS baseline and the AE hand-crafted baselines. Moreover, a new probabilistic model, based on the variational autoencoders (VAE), is proposed for short-packet wireless communication systems. Using this new approach, the information messages are represented by the so-called packet hot vectors (PHV), which are inferred by the VAE latent random variables (LRVs). Numerical results show that the proposed VAE, with a DL classifier, improved symbol error rate (SER) performance in comparison to the baseline schemes.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

Acknowledgements

All the praises and thanks be to Allah Almighty, for giving me the opportunity to accomplish this work.

I would like to express my deep gratitude and appreciation to those who inspired me and have given me the drive and discipline to tackle an task with enthusiast and determination during the development of this thesis. I want to extend my deep gratitude and appreciation to my Ph.D. supervisor Dr. Khairi Hamdi, for his supervision, invaluable advice, guidance, immense knowledge, patience and kind support in all the time of my academic research and daily life. Additionally, I am thankful for my collaborators, Mutasem Hamdan, Abdulmajeed Alenezi and Ahlam Al-shukaili and my friends who supported and cheered me along the Ph.D. journey.

I would like to extend my deepest respect and thankfulness to the Kingdom of Saudi Arabia's government for funding my graduate studies at the University of Manchester.

Finally, and most importantly I would like to thank my parents for their prayer, profound encouragement patience, endless guidance, generous understanding, support and for being my constant source of inspiration. Also, I must express my appreciation to my brothers and sisters for their encouragement, love, kindness and constantly supported me in all my pursuits.

Dedication

This research is lovingly dedicated
to my beloved family

List of Abbreviations

AE	Autoencoder
AESC	Autoencoder Semantic Communication
AF	Amplify and Forward
ALAE	Adversarial Latent Autoencoder
ANN	Artificial Neural Network
ASR	Automatic Speech Recognition
AWGN	Additive White Gaussian Noise
BCD	Binary to Decimal
BER	Bit Error Rate
BLER	Block Error Rate
BNNs	Bayesian neural networks
BP	Belief Propagation
BPSK	Binary Phase-Shift Keying
CNN	Convolutional Neural Networks
CSI	Channel State Information
CUDA	Compute Unified Device Architecture
CV	Computer Vision
DeepSC	Deep Learning-based semantic communication system
DCB	Decimal to Binary
DCNNs	Deep Convolutional Neural Networks
DF	Decode and Forward

DL	Deep Learning
DLIM	Deep Learning Index Modulation
DLVM	Deep Latent Variable Model
DNN	Deep Neural Network
DQN	Deep Q-Network
DR	Dimensionality Reduction
DPGM	Directed Probabilistic Graph Model
ELBO	Evidence Lower Bound
ELM	Extreme Learning Machine
EM	Electromagnetic
FC	Fully Connected
GAN	Generative Adversarial Network
GD	Gradient Descent
GPU	Graphics Processing Units
HDPC	High-Density Parity Check
IM	Index Modulation
IRS	Intelligent Reflecting Surfaces
KL	Kullback-Leibler
KNN	K Nearest Neighbors
LDPC	Low Density Parity Check
LLR	Log-Likelihood Ratios
LRV	Latent Random Variable
LSTM	Long Short Term Memory
MAE	Masked Autoencoder
MAP	Maximum Posterior
MASK	M-ary Amplitude Shift-Keying
MCMC	Markov Chain Monte Carlo
MER	Message Error Rate
MF	Matched Filtering
MFSK	M-ary Frequency Shift-Keying

MIMO	Multiple-Input and Multiple-Output
ML	Machine Learning
MLD	Maximum Likelihood Decoding
MLP	Multi Layer Perceptrons
MMSE	Minimum Mean Square Error
MSE	Mean Squared Error
MSB	Most Significant Bit
NN	Neural Network
NLP	Natural Language Processing
OFDM	Orthogonal Frequency Division Multiplexing item[OFDM-IM] Orthogonal Frequency Division Multiplexing with Index Modulation
OWAF	One-Way Amplify-and-Forward
ParVI	Particle-based Variational Inference
PCA	Principal Components Analysis
PHV	Packet Hot Vectors
PNN	Partitioned Neural Network
PU-RXs	Primary User Receiver
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase-Shift Keying
RBF	Rayleigh block-fading
ReLU	Rectified Linear Unit
RL	Reinforcement Learning
RTN	Radio Transformer Network
RNN	Recurrent Neural Network
SDR	Software-Defined Radios
SE	Spectrum Efficiency
SER	Symbol Error Rate
SGD	Stochastic Gradient Descent
SGDM	Stochastic Gradient Descent with Momentum
SISO	Single-Input Single-Output

SNR	Signal-to-Noise Ratio
SVBL	Signal Vector Based List
SVM	Support Vector Machine
TA	Transmit Antennas
TWAF	Two-Way Amplify-and-Forward
URLLC	Ultra-Reliable Low-Latency Communication
VAE	Variational Autoencoder
ZF	Zero Forcing
ZSL	Zero-Shot Learning
2DConv	Two-dimensional Convolution

Chapter 1

Introduction

In this chapter, section 1.1 states the background of the work, section 1.2 describes the motivation behind the work, and section 1.3 states the overall objectives of the thesis. The major contributions of this dissertation and the relevant publications are summarized in sections 1.4 and 1.5, respectively. Finally, the outline of the thesis is discussed in section 1.6.

1.1. Background

Wireless communication has been long sought for achieving optimal and reliable schemes for the successful transfer of data. In recent years, many experts have investigated the use of machine learning (ML) to improve performance in wireless communications. Solving complex issues in computer vision (CV), automatic speech recognition (ASR) and natural language processing (NLP) using deep learning (DL) have demonstrated significant success that has inspired communication experts to apply DL in their fields. Using an inflexible mathematical model to characterize a real-world image or language is where DL shines [3, 5–7].

Generally, the mathematical models of most communication signal processing are linear and stationary as they have strong foundations in information theory and statistics. However, practical systems have many imperfections and non-linearities. A deep learning-based communication system does not require a controllable mathematical model to optimize such an imperfection [3].

In an end-to-end manner, the deep learning-based communication system attempts to optimise the transmitter and the receiver without defining the block structure. This system can perform better or equal to the existing system due to its simple structure that includes less processing delay, less power consumption and less computing complexity and can be easily implemented in practical systems.

Based on the above background, the objective of this work is to investigate the deep neural network (DNN) for an end-to-end communications system to achieve a better or similar bit error rate (BER).

1.2. Motivation

Several critical challenges in the development of learning-based end-to-end communication systems need to be addressed before DL framework can be applied across a variety of wireless channels. According to well-known practice, the DNN weights are generally updated using the stochastic gradient descent (SGD) method, with the calculated error gradients propagated from the input layer to the output layer [8]. The end-to-end system gain is prevented when the channel transfer function is unavailable, preventing the backpropagation of the gradients from the receiver DNN to the transmitter DNN. In such a situation, the learned weights would be biased by the assumed channel function, thereby repeating the pitfalls caused by the discrepancy between the assumed channel model and the real channel. An additional difficulty associated with the end-to-end paradigm is the curse of dimensionality during training, particularly when the transmitted symbol sequence is long. To ensure a sufficient coding gain, the code block size in a communication system must be large enough. In spite of this, the number of possible codewords grows exponentially based on the size of the code block, increasing the number of undetected codewords during training.

In addition, Conventional wireless communication systems have historically relied on modelling assumptions, often Gaussian-based, linearity, and stationarity. These assumptions have proven suitable for mathematical analysis and have been manageable. However, this approach has its limitations when accurately representing real-world scenarios. In practice, there are discrepancies between theoretical models and practical systems, primarily caused by shortcomings such as non-linearities (e.g., imperfect power amplifiers or analog-to-digital converters with resolution sensitivity). These inherent issues naturally lead to a decrease in

system performance.

In light of the above arguments, this work investigates and pursues solutions to wireless communication problems by utilizing a variety of machine learning techniques, such as assisted communication systems with auto-encoders and deep learning variable Autoencoders for end-to-end optimized wireless communication systems.

1.3. Aims and Objectives

The main aim of this research is to apply and investigate different ML applications, rethink the wireless communications system design problem and contain the potential for performance enhancements in challenging communications scenarios that are hard to model with tractable mathematics. In this research, we cover various practical ML techniques to represent the transmitter, channel, and receiver as one deep neural network (NN) that can be trained as end-to-end communications before testing the performance of the model. The main objectives of this research are detailed as follows:

- To study and analyze advanced ML techniques for representing wireless communication models by black boxes that are capable of autonomously learning the entire wireless scenario and find the research gaps and challenges.
- To propose new techniques that improve the bit error rate (BER) and symbol error rate (SER).
- To study and investigate the performance of the new technique in comparison with the conventional maximum likelihood detector based on BER and SER.
- To provide a numerical evaluation of the new techniques to support the outcomes of the experiment simulation.
- To design advanced distributed ML techniques to help the state of the art of wireless communication techniques such as intelligent reflecting surfaces (IRS).

1.4. Key Contributions

The main contributions of this thesis are illustrated as follows:

- C_1 (Chapter 4): A deep learning (DL) algorithm has been applied to enhance signal detection and performance of multiple-input-multiple-output (MIMO) based orthogonal frequency-division multiplexing (OFDM) systems with index modulation (IM). The proposed detector termed deep learning index modulation (DLIM) is used as fully connected layers of a deep neural network (DNN) and adopted to achieve minimum bit error rates in IM-MIMO-OFDM over Rayleigh wireless channels.

- C_2 (Chapter 5): An IRS-assisted end-to-end communication system that operates over additive white Gaussian noise (AWGN) channels is proposed, where the modulation and demodulation are performed by DNN based on an AE architecture. The novelty of the proposed fully data-driven system without any prior knowledge lays on how effectively the system can learn with the assistance of IRS to encode and decode the transmitted $\mathbf{s}$ and the received $\widehat{\mathbf{s}}$ symbols while minimizing the mean square error (MSE) between $\mathbf{s}$ and $\widehat{\mathbf{s}}$.

- C_3 (Chapter 5): The symbol error rate (SER) performance numerical results show that our AE-based scheme achieves better performance gains than the existing classical IRS baseline [4] and the AE hand-crafted baselines [3]. Moreover, we back up these results by explaining the constellations AE learns to improve the system performance.

- C_4 (Chapter 5): The proposed AE design results show that higher SER performance can be achieved, even with using a small number of elements. This overcomes the weakness of IRS-assisted communication systems performance that have a small number of IRS elements which are mentioned in [9, 10].

- C_5 (Chapter 6): A novel end-to-end communication system is proposed that represents the symbol as packet hot vectors (PHV) and operates over binary phase-shift keying (BPSK) modulation in AWGN channels, where modulation and demodulation are performed by a deep neural network (DNN) based on a VAE architecture.

- C_6 (Chapter 6): The experiment is extended to investigate the QPSK modulation, Rayleigh, Rician fading channels, shadowing, and Doppler effect.

- C_7 (Chapter 6): While the baseline AE uses 4 and 7 channels in [3] to achieve their results. In this proposal, we efficiently use two channels only to achieve better performance than AE baseline.

- C_8 (Chapter 6): This considers a VAE with two latent random variables (LRVs), and a simple classifier can reconstruct the transmitted message by sending only the LRVs' parameters and the message error rate (MER). The result shows that the performance of our proposed system is better than that of the existing classical scheme.

1.5. List of Publications

The list of publications that have been extracted from this thesis are detailed as follows:

- P.1 (Chapter 4): M. A. Alawad and K. A. Hamdi, "A Deep Learning-Based Detector for IM-MIMO-OFDM," 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), 2021, pp. 1-5, doi: 10.1109/VTC2021-Fall52928.2021.9625569.

- P.2 (Chapter 5): M. A. Alawad, M. Q. Hamdan and K. A. Hamdi, "End-to-End Deep Learning IRS-assisted Communications Systems," 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), 2021, pp. 1-6, doi: 10.1109/VTC2021-Fall52928.2021.9625398.

- P3. (Chapter 6): M. A. Alawad, M. Q. Hamdan, K. A. Hamdi, C. H. Foh and A. Quddus(2022),"A new approach for an end to end communication system using Variational Auto-encoder (VAE)," in 2022 IEEE Global Communications Conference (GLOBECOM). IEEE, 2022, pp.1–6.

- P4. (Chapter 6): M. A. Alawad, M. Q. Hamdan and K. A. Hamdi, "Innovative Variational AutoEncoder for an End-to-End Communication System" 2022 IEEE Access.

1.6. Thesis Organization

This thesis is structured into seven chapters and has been structured as follows:

- Chapter 2 presents the background theory of Machine learning and deep learning basics and several key works of literature related to wireless communication. There we explore the potential, challenges and requirements that should be considered when we plan to build a communication system using machine learning.

- Chapter 3 discuss the concept of autoencoder (AE) as an end-to-end communication system proposed by [3] and compares it with the conventional communication system. In addition, we analyse the performance of the AE as an end-to-end learning system and compare it with the conventional communication system considering different modulation schemes with the intention of achieving a better or similar bit error rate (BER).

- In chapter 4, a DL approach is proposed to explore the effectiveness of employing a DNN for the detection of IM-MIMO-OFDM in terms of error performance.

- In chapter 5, the classical IRS-assisted communication system has been transformed into an end-to-end AE. Additionally, we proposed DNN loss function that shows the capability of optimizing the DNN parameters to reconstruct the transmitted signal by learning signal constellations representation pattern that reduces the wireless environment contamination impact on the received signal.

- In chapter 6, a novel approach uses the VAE as a probabilistic model to reconstruct the transmitted symbol by transmitting the statistical parameters of the LRVs through the physical layer.

- Chapter 7 concludes the work and describes the future of this research.

Chapter 2

Background Theory

This chapter presents the background information for several key concepts and theories that are utilized in the thesis. Section 2.1 states some of the fundamental characteristics of ML in wireless communications, including the background, basics, and techniques. The fundamental characteristics of DL are described in section 2.2. In section 2.3, a key literature review on the deep learning approach on the physical layer of the communication systems is introduced. The fundamental characteristics of the intelligent reflecting surface (IRS) are described in section 2.4. Finally, section 2.5 summarizes the chapter.

2.1. Machine Learning Techniques in Wireless Communications

2.1.1. Background

In the past decade, ML has changed the landscape of the engineering field with it is capacity of solving complex problems through different applications, such as CV and NLP that leads the research community and industry to advocate the application of ML in wireless communications. Researchers have proposed ML algorithms for several sub-tasks in the physical layer such as encoding and decoding [11, 12], modulation recognition [13, 14], channel estimation and equalization [15, 16] etc. Several ML techniques are introduced in this chapter, including supervised learning, unsupervised learning, reinforcement learning and DNN. Below, each section begins with an introduction to the learning technique and its applications in B5G networks.

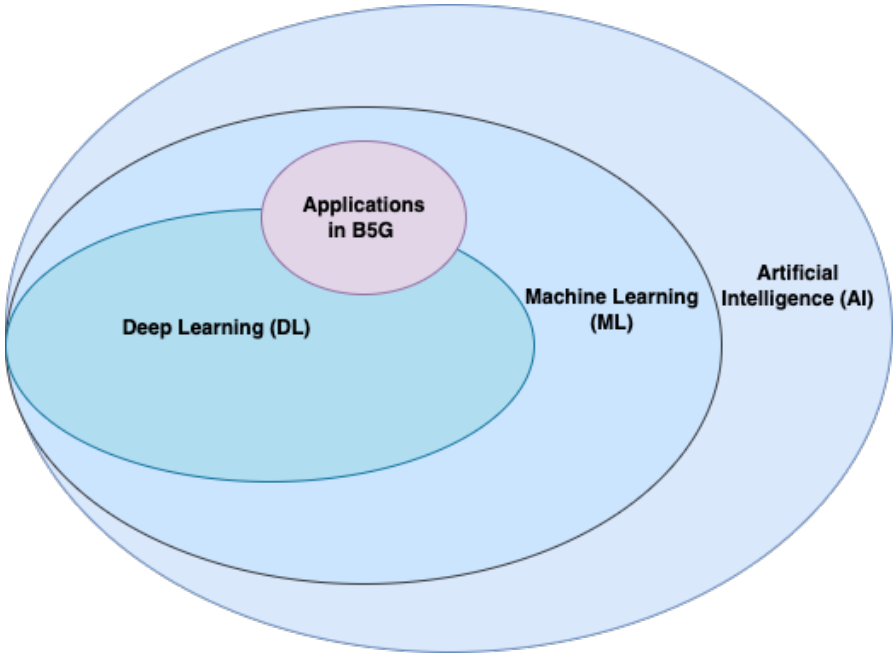


Figure 2.1: Venn diagram of the relationship between artificial intelligence, machine learning, deep learning and B5G technology.

2.1.2. Supervised Learning

Supervised learning is a ML technique that is used to label the dataset to map the features of pairs (input and output) information to reduce the label. Using this kind of technique begins with gathering data, each with a label then converted to feature vectors, and based on the training of the data analyst, the model can predict the output of any new input data set.

Applications in B5G:

Supervised learning algorithms in wireless communications are mainly utilized for the classification of points/objects. Here are some of the most common supervised learning techniques:

- **Support Vector Machine (SVM)** is one of the most often used techniques in supervised learning for classification. The goal of the SVM is to find the optimal values that leverage the data [17].
- **K Nearest Neighbors (KNN)** is also a supervised learning algorithm that is used for classification. Contrary to the other techniques, KNN use

memory for all the training [18].

2.1.3. Unsupervised Learning

Unsupervised learning is another ML algorithm where there are no labels on the data given but a function to describe the hidden pattern in data can still be found. Unlike the supervised algorithm, discovering and presenting the structure of the data is left to the unsupervised algorithm.

Applications in B5G:

The most common unsupervised learning technique in wireless communications is cluster analysis or dimensionality reduction, such as K-mean clustering and principal component analysis (PCA), respectively:

- **Clustering**: The K-Means Clustering Algorithm is used to partition the data point into K distinct clusters. These points will belong to the nearest mean of the cluster.
- **Principal Component Analysis (PCA)** is a dimension-reduction tool where the output of the model has fewer features than the input but still contains the most information about the input data set [19].

2.1.4. Reinforcement Learning (RL)

The goal of reinforcement learning (RL) is to learn the policy (function). In RL, different actions are executed in every state that has been perceived by the environment as a feature vector. Within the same environment, the agent can move from one state to another, as a different action can bring different rewards. The policy in RL takes the input from the state of the feature vector and the output is taken from the optimal action to be executed in that state. Whenever the maximum of the expected average rewards is reached, then the action is optimal [20].

Applications in B5G:

The following RL algorithms are applied in wireless communications. Below are some of the most commonly adopted reinforcement learning algorithms

- **Q-Learning**: Based on which RL agent takes an action to learn the Q values, the RL agent must interact within the environment. Starting at a tuple of a state and an action followed by a certain policy, the Q values are defined as discounted accumulative rewards. After this, the learning process where the Q values are learned and under the current state, the agent can make a decision per the largest Q value [21].
- **Deep Reinforcement Learning (DRL)**: In DRL, the interaction with the environment generates the state transition sample and stores it in the replay memory. After, it samples to train the deep Q network (DQN), and a target DQN is adapted to generate the target values, which help stabilize the training DRL [22].

Fig. 2.2 summarises some of the ML techniques and applications that have been used for future wireless communications.

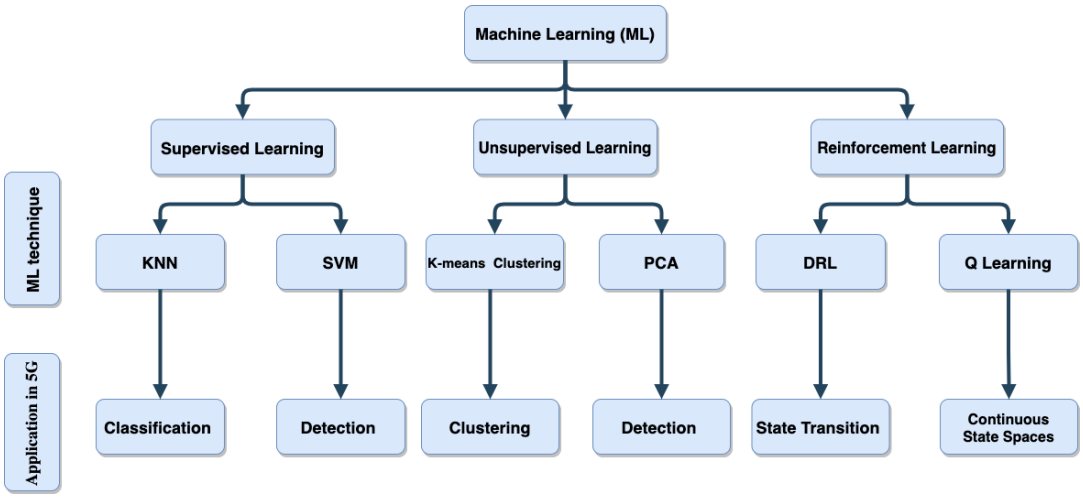


Figure 2.2: Machine learning techniques and applications [1].

2.1.5. Deep Neural Networks (DNN)

A neural network (NN) is a feed-forward network, where the first layer is the input layer and the last layer is the output layer. All the layers in between are hidden layers. The word *deep* generally refers to the network ranging from a few to more convolutional layers. The layers in deep neural networks (DNN) are fully connected by neurons. This connection provides the entire learning feature from the previous layer. Each DNN layer consists of a number of neurons,

and each input of these neurons is communicated through weight, while the output is corresponding through activation functions. There are several traditional activation functions, such as linear, ReLU, tanh, etc. Moreover, to update the neural network parameters, we use a backpropagation chain rule to compute the gradient on NN [20], [1]. The structure of a typical fully connected feed-forward NN is shown in the Fig. 2.3 below:

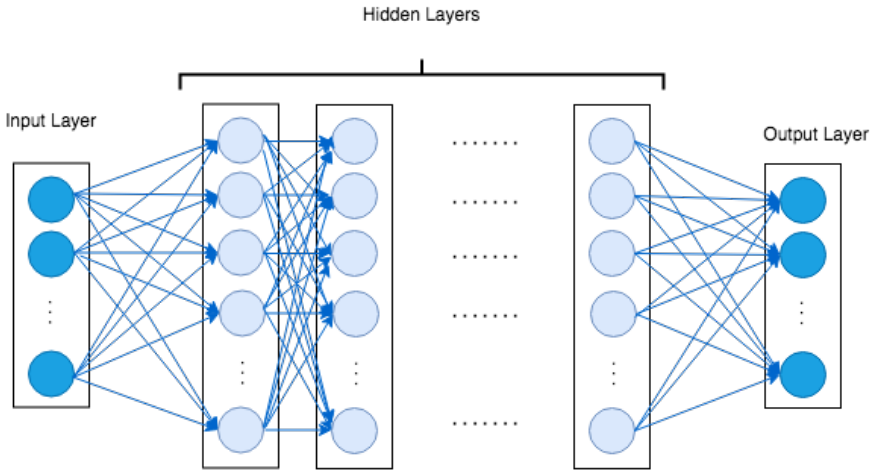


Figure 2.3: Layout of fully connected feed-forward NN.

2.1.6. Convolutional Neural Networks (CNN)

Convolution neural networks (CNN) are a special kind of feed-forward network, without losing the quality of the model, the number of units in DNN can be reduced in a CNN. Unlike any other NN layer, the working principle of the convolution layer does not employ connection weights and a weight sum. Instead, it contains a filter (convolution filter) that converts images. Convolution neural networks are mostly used with image and text processing.

2.1.7. Recurrent Neural Networks (RNN)

Recurrent neural networks (RNN) are used to generate sequences, classify and label. RNNs are mostly used for processing a text, as the texts and sentences are often sequences of characters or words. Labelling is to predict the class of each feature vector in the sequence and classification is to predict a class of whole sequences. To generate a sequence is to output another sequence somehow related to the input sequence [20].

2.1.8. Autoencoders (AE)

An autoencoder is considered a feed-forward unsupervised neural network that has three layers net. The AE is designed to learn how to de-noising the data by compressing and encoding the data before transmitting, then giving an output similar to the input by reconstructing the data at the receiver. Generally, the AE consists of four main components, as per Fig. 2.4 below:

- **Encoder:** In the encoder, the model learns how the dimension of the input data is reduced and compressed as an encoded representation.
- **Bottleneck:** The bottleneck is the layer that has the lowest dimension of the compressed representation of the input data.
- **Decoder:** In the decoder, the model learns how to be as close to the input by reconstructing the data from the encoded representation.
- **Reconstruction loss:** This step is to evaluate the performance of the decoder, and how accurate the output to the input is.

More details about AE will be illustrated in Chapter 3.

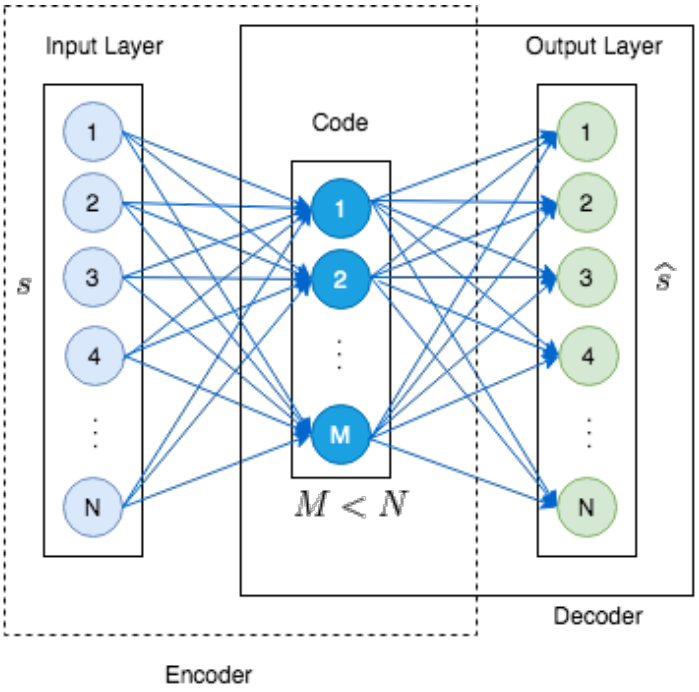


Figure 2.4: The structure of an AE.

2.2. Deep Learning Basics

The brain cells termed artificial neural network inspires the structure and functions of DL. DL is a subfield of ML, which is not programmed, though it has a concerned algorithm that helps make decisions, classify or predict based on given data. To optimise a given target from raw input data, DL uses a different number of layers to extract higher-level features. As we mention in the background for the ML algorithms, DL also expands in supervised, unsupervised and reinforcement learning [23].

The development of theory and implementation began between the 1940s-1960s, when the first DL perceptron model succeeded in training one single neuron [24]. In the 1980s, it was successful when Rumelhart et al. used the backpropagation of NNs for one or two hidden layers [23], [25]. This technological advancement has garnered the attention of many researchers to the DL approach that has eliminated computational limitations. These advantages lead to a wide range of different applications that used the DL approach.

It is possible to build a DL model from scratch, but it requires a great effort, as it is a complex task. At each layer, we must define a feed-forward function and a gradient propagation operation. In addition to that, for the training model, we must efficiently implement fast optimisation algorithms. Also, we must compute unified device architecture (CUDA) coding for the graphics processing units (GPU) parallelisation. However, due to the continuous popularity of DL applications, a large number of algorithms, tools and libraries must be developed. They are built with a high-level algorithm and massive parallel GPU architectures that quicken the training process with a huge amount of data. The high-level algorithm can be defined in various programming languages or configuration files, automatic differentiation of training loss functions through arbitrarily large networks, and a compilation of the network's forwards and backwards passes into hardware optimised concurrent dense matrix algebra kernels [3]. TensorFlow, PyTorch and Caffe are the most common and widely used libraries for DL.

2.3. Literature Review

As we mentioned above, the DL approaches aim to enhance the performance of the application on the physical layer of the communication system. We can summarise

this approach by replacing the whole block-based communication system with DL or improving and optimising certain blocks. More details about these two approaches will be covered in our literature review.

2.3.1. Block-based Communication

The conventional communication system consists of two main blocks which are the transmitter and the receiver. At the transmitter side, there are source coding, channel coding, modulation, and then transmitting over the communication channel. At the receiver side, there are demodulation, channel decoding, and source decoding as shown in Fig. 2.5. Based on different optimising algorithms that have been developed over the years for each block, this communication structure aims to achieve the optimal solution.

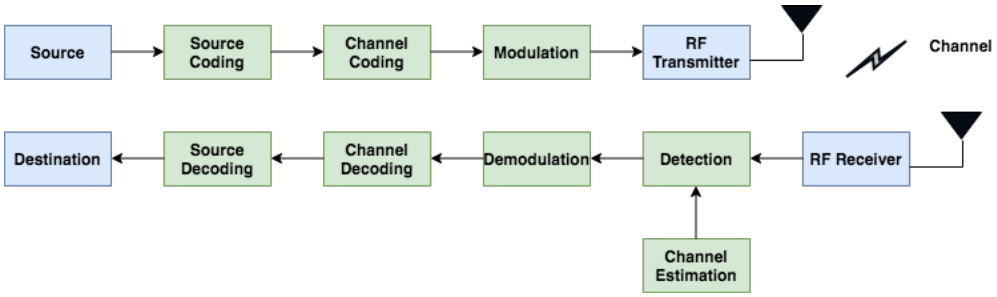


Figure 2.5: The structure of a conventional communication system.

Many studies in recent years have proposed the use of conventional ML approaches, such as small feed-forward NNs and SVM as alternative implementations for individual block tasks. Recently, many processing blocks of these DL-based algorithms have outperformed conventional communication algorithms. Also, DL-based algorithms are introduced to adapt to different complex communication environments. These DL structures and algorithms development approaches have been introduced to many of the DL applications as alternatives to the conventional algorithm, such as modulation recognition and channel decoding [26].

2.3.2. Modulation Recognition

The goal of modulation recognition is to classify or predict the pattern of different modulation schemes of the received signal, which is an important task to facilitate

communication among different communication systems. It can also be used to interfere with and monitor enemies' signals for military use. Over the years, many studies have been conducted on modulation recognition using different approaches, such as decision-theoretic and pattern recognition. Preprocessing, feature extraction and classification are examples of several common procedures. For decades, many studies have been conducted on modulation classification using different approaches, such as expert feature engineering, analytic decision tree or trained discrimination method. Some other advanced approaches have used different patterns on expert feature maps, such as the spectral coherent function with NN-based classification α-profile, combined with a NN-based classification [3].

The authors in [27], proposed a powerful modulation classifier as a NN architecture to distinguish a noise corrupted band-limited modulated signals from 13 types of analog and digital modulation schemes. From the original signal, this modulation classifier can manually extract features that characterize analog or digital modulations as well as some basic parameters, such as amplitude, frequency and phase. In order to discriminate the modulation schemes, these features are fed to four-layer NNs. To estimate the level of the M-ary amplitude shift-keying (MASK) and the M-ary frequency shift-keying (MFSK) modulation and it is identified as a second two-layer NNs. The performance of these approaches strongly depends on the extracted features due to the limited learning ability of conventional NNs [26].

In [3], the authors proposed an automatic learning feature from raw data using the superior learning capability of DNNs to optimize the end-to-end performance and the possibility to replace the artificial feature extraction. They have looked into the modulation classification problem of a single carrier modulation scheme based on the sampled radio frequency time-series data using the approach of CNN. The CNN modulation classification approach is trained data consisting of 1.2M sequences of 128 complex-valued baseband in-phase and quadrature (I,Q) samples corresponding to 10 different analog and digital single-carrier modulation schemes, which have been sent through a wireless channel with multipath fading, sample rates and centre frequency offsets. The results of their CNN-based modulation classifier approach have shown better performance compared with the other two approaches: an extreme gradient boosting with 1000 estimators and a single scikit-learn tree working on the extracted expert features in the low to medium signal-to-noise ratio (SNR) range, and with the high SNR, the performance is similar.

2.3.3. Channel Decoding

Due to the straightforward channel decoding applications of NNs, ML-based decoders have been used since the 1990s. In bit-level processing, the channel coding algorithm was conveniently treated and represented the bits in codewords or one-hot representation as the inputs and expected outputs of NNs. Moreover, as the codewords are random and sampled with help of the distribution of channel noise, the NN can learn from this noise version of the codewords and avoid the overfitting problem in each training epoch. After the training of the channel decoder, the decoding process becomes simple and with very low latency as the architecture of the NN does not require expert knowledge [26].

However, the process of learning and classifying using the DL-based algorithm is restricted due to it is dimensionality. Luckily, the DL algorithm not only provides a parallel implementation for complex computations but also provides a multi-layer architecture with realisable training methods that offer DL with outstanding learning capacity [28].

In [29], the authors proposed a fully connected DNNs-based decoder. Their goal was to enhance the performance in decoding high-density parity check (HDPC) codes using the belief propagation (BP) algorithm. Normally, using the BP algorithm, we can achieve as almost as Shannon capacity when we decode low-density parity-check codes, though not when we decode HDPC codes.

A DNNs-based BP algorithm differs from the original algorithm when weights are added. In a DNN-based BP algorithm, the weights are assigned to the edge in the Tanner graph where each variable node is connected to some check nodes. While in a DNN-based decoder, the weights share the same decoding structure of the Tanner graph.

A BP-RNN decoder, which is transformed from the RNN architecture to form a fully connected DNN-based decoder was proposed in [29]. This proposal aims to unify each iteration weight and gives feedback for the output of parity layers into the inputs of variable layers. This method's performance exceeds the plain BP method, as it remarkably reduces the number of parameters.

Another proposal in [30], based on DNN architecture, termed NN decoder (NND) aims to decode codewords of length N with K information bits. In order to observe the communication channel effect, the codeword of N length must pass through the modulation layer and the AWGN channel layer. After this step, for

a network of three layers, an N-dimensional input will generate the information of log-likelihood ratios (LLRs) of the noisy codeword, and it will be trained to output with K estimated information bits. Their proposal performs well and achieved the maximum posterior (MAP) for a random and structured code of 16-bits length, but not when we increase the number of information bits. However, with regard to the issue of one of the DNN advantages is learning the decoding algorithm. We can do that by generalising a subset of codewords for training purposes to unseen codewords when decoding structure codes. The DNN also has more advantages that make it a promising algorithm, such as one-shot decoding and parallel architecture. Moreover, the authors of this paper show that using binary cross entropy or mean squared error (MSE) as a loss function has no outstanding effect on the final results when we train with direct channel values or log-likelihood ratios (LLRs). Furthermore, they propose the existence of an optimal SNR for training to classify the codewords over arbitrary SNRs and argue that having more training epochs can lead to better performance.

In [31], the authors proposed a partitioned neural network (PNN) architecture that competes with the BP algorithm and the conventional successive cancellation. However, with a high number of sub-blocks, the performance decay restricts the number of applications for large codes. Regardless, PNN still offers a promising solution to dimensionality problems.

2.3.4. Deep Learning Based End-to-End Communication

Sending a message from the source (transmitter) over a communication channel to the destination (receiver) has some requirements in the communication system, such as block structure. The block structure helps control and enable individual analysis for each block. However, optimising each block using this structure can not guarantee that we will reach the global optimisation for the communication problem. In order to reach the improved performance of end-to-end communication, a joint optimization of two blocks or more is necessary.

Recently, O'Shea et al. introduced a novel concept by recasting the communication task as an end-to-end reconstruction optimisation task and replacing the artificial block construction of the conventional communication system [3]. Based on DL theories, this novel concept represents the simplified system as an AE system. The initial studies have shown the performance and

potential of the end-to-end method compared with the convolution system. In this section, we focused our literature review on AE-based end-to-end communication and cover some of the very recent work based on that.

2.3.5. Autoencoder Based End-to-End Communication

In [3], the authors introduced the AE first as a new design of the communication system. This system was presented as an end-to-end reconstruction task where the transmitter and the receiver are jointly optimized in a single process. In this paper, they used a DL model to learn the implementation of the transmitter and the receiver for a given channel model that optimises for a specific loss function. All of the transmitters, channels and receivers were implemented as one DNN in order to train as an AE. They represent the transmitter and the receiver as fully connected DNNs, while the AWGN channel in between is represented as a noise layer with a desired variance. Thus, the AE can be seen as a communication system that aims to learn the message s out of M possible messages $s \in M = 1, 2..., M$, in order to generate the representation of the transmitted signal x which is robust against the communication channel. At the receiver side, the original message produces $\widehat{s}$ the transmitted message s with the lowest error rate by learning from the received y.

2.3.6. Autoencoder for Multiple Transmitter and Receiver

The authors in [3], extend their work of AE to multiple transmitters and receivers, which share a common channel. This application means that the transmitter-receiver pair of each user attempting to communicate over the same channel simultaneously leads to channel interference. In this scenario, the goal of each transmitter-receiver pair is to optimize the system in order to transmit their own messages with the best accuracy. This system represents both transmitter-receiver pairs as multiple input and output NNs that are jointly optimised and minimize the weighted sum of both losses, i.e., This AE system achieves similar or better BER performance at the same communication rate than the conventional uncoded quadrature amplitude modulation QAM schemes, thereby validating the potential application of the AE model in multi-user cases as well.

2.3.7. Over-the-Air Communications Systems

Using the initial finding of [3], the authors of [32] aspired to model an AE for over-the-air transmission. Using unsynchronised off-the-shelf software-defined radios (SDRs) and open-source DL software libraries to model, train and run a NN of a communication system. The authors were able to overcome the restriction of short block lengths of the current AE for continuous data transmission and receiver synchronisation by implementing a frame synchronisation module mechanism based on another NN. They have shown a comparable BER performance close to 1dB of the learned system with a practical baseline. Moreover, to overcome training the model over the actual channel, they implemented two-phase training procedures based on the concept of transfer learning by fine-tuning the receiver part of the model to capture the actual channel effect. Finally, the study shows that it is possible to build an end-to-end communications system where the whole physical layer processing is carried out by NN.

The two-phase training strategy we mentioned in the previous paragraph has some limitations. In the fin-tuning step, it is not possible to backpropagate through the black box of the radio channel. However, it is only possible to optimize the receiver side. This restriction has been overcome in [33], where they propose an approach to the adversarial of the channel response approximation. This learning-based approach does not require an analytical model impairment in the channel. The generative adversarial approach adopted helps the model to jointly optimize these tasks:

- Channel response approximation of an arbitrary communication system.
- Learning an optimal encode and decode scheme that optimizes any given performance metrics.

This paper has shown that by using such a model, can achieve a robust performance without needing a closed-form channel model or implementation.

2.3.8. Communications System without Channel Model

The novelty in [34] is solving the shortcoming of the AEs that requires a differentiable channel model. The authors introduced an algorithm that iterates between unsupervised learning training and RL at the receiver and the transmitter, respectively. By using two different parametric functions, the transmitter and

receiver have been implemented to be independently optimised. Normally, the former approach requires a differentiable channel, and they proposed a model that does not require a channel model at all. The iterations of the training algorithm are done in two stages: at the receiver and then at the transmitter. It will not stop until the benchmark is satisfied. Since the receiver has access to the sent message, training the receiver is considered an unsupervised learning approach. While the receiver is a reinforcement learning task, it generates channel symbols that will minimize the scalar loss that had been generated by the receiver. This paper has shown that over additive white Gaussian noise (AWGN) and Rayleigh block-fading (RBF) channels, the alternating training model performs similarly to the fully supervised approach. However, it still requires an additional reliable channel during training in order to provide feedback on the losses from the transmitter to the receiver.

2.3.9. Deep Reinforcement Learning Autoencoder with Noisy Feedback

The authors in [35], used the initial finding of [34] by showing that alternating training can work even with a noisy feedback channel without performance loss to an extent to a certain limit. They have designed a system that can learn to transmit real values over unknown channels without any need to have a reliable link or a preexisting feedback link. Then, they used this feedback system for the alternating training of the AE to communicate the losses. This paper has shown that using this feedback system can achieve the same performance as if we trained the system with a perfect feedback link over AWGN and RBF channels. However, for the noisy feedback, it performs well without performance loss only up to a certain MSE and for the feedback system, up to a realistic training SNR.

2.3.10. Deep Learning Based Autoencoder for Interference Channels

One of the challenges that the data-driven end-to-end communication system faces is the robustness of the AE under an interference challenge. In [36], the authors began investigating the issue of unknown interference as it can be applied to different channel models as well as the loss functions that give the optimal solutions. The AE successfully decodes the signals without the knowledge of the interference channel at the weak interference case. Then, they investigated the performance of the AE under different levels of interference by introducing

a symmetric k-user Gaussian model, where they classified the interference to different levels from weak to very strong, based on the value of coupling parameter α. They found that, based on the interference levels and the knowledge of α that from the weak to moderate interference channels for training the AE, the system performs well up to 10% offset for α. However, to have and maintain good performance for the strong interference case, the offset for α has to be less than 5%, as well as for the very strong interference case, the offset for α has to be less than 2%.

In [37], the authors proposed an adaptive algorithm to solve the performance issues of the AE under strong and very strong interference. Based on the level of interference, the algorithm was able to predict the decoding process in real-time. This paper has shown how the adaptive learning algorithm significantly improved the robustness of the interference channel and provided an adaptable AE to real-time interference on many levels of interference. However, compared with the conventional AE with offline learning, this algorithm is more notable when we have a strong and very strong interference level.

2.4. Multiple-Input Multiple-Output OFDM with Index Modulation

It has been established that multi-carrier systems play an important role in high-speed wireless communications due to the robustness of these systems to multipath fading and interference [38]. A number of new multicarrier technologies, including generalized frequency division multiplexing and filter bank multicarrier modulation, have recently been proposed as being eligible for fifth-generation (5G) technology. Orthogonal frequency division multiplexing (OFDM) is the most widely used and well-established multicarrier modulation method that is the basis of many existing wireless standards, including IEEE 802.11 (Wi-Fi) and Long Term Evolution-Advanced (LTE-A) [39, 40].

It has become increasingly popular to study multiple-input multiple-output (MIMO) transmission techniques over the past decade due to their advantages over single antenna techniques, such as improved data rates and reduced energy consumption. One of the most promising solutions for next-generation communication systems based on MIMO spectral and energy-efficient solutions is spatial modulation (SM), which transmits information bits based on the indices

of the active transmit antennas in a MIMO system [41, 42]. The use of SM has attracted significant attention from researchers in the past few years [43–48], and it remains a hot topic in wireless communications [49].

A new multicarrier transmission technique known as OFDM with index modulation (OFDM-IM) [50] has been proposed as an alternative to classical OFDM. OFDM-IM is based on the SM concept, involving index modulation techniques for the indices of the available subcarriers of an OFDM system.

In OFDM-IM, only a subset of the available subcarriers is selected as active based on the information bits, whereas the remaining inactive subcarriers are set to zero. Therefore, information is transmitted not only using the data symbols selected from M-ary signal constellations but also by the indices of the active subcarriers. In contrast to conventional OFDM, OFDM-IM allows for the adjustment of the number of active subcarriers in the system, and this flexibility in system design enables an interesting trade-off between error performance and spectral efficiency. Furthermore, it has been demonstrated that OFDM-IM has the potential to perform better when it comes to error performance than classical OFDM when spectral efficiency is in the low-to-mid range. Considering its adjustable number of active subcarriers, OFDM-IM is not only a potential candidate for high-speed wireless communications systems, but also for machine-to-machine (M2M) communications systems that require low power consumption [50].

Over the past two years, researchers have been studying the subcarrier index modulation concept as it pertains to OFDM [50–52], and a number of studies have been conducted on it recently [53–62].

It is shown in [53] that a tight approximation for the error performance of OFDM-IM can be obtained. The OFDM-IM scheme has been generalized in [54] by introducing a more flexible method for selecting active subcarriers as a means to increase spectral efficiency further. The authors in [55, 56] address the problem of selecting an optimal number of active subcarriers. To improve the performance of OFDM-IM, subcarrier-level block interleaving is introduced in [57] to take advantage of uncorrelated subcarriers. OFDM-IM with interleaved grouping is adapted for vehicular communication in [58]. For the purpose of achieving additional diversity gains, OFDM-IM is combined with coordinate interleaving in [59]. Recently, OFDM-IM and its variants have significantly outperformed classical OFDM in terms of ergodic achievable rate [60] and coding gain [61].

In light of the advantages of both OFDM and MIMO transmission techniques, their combination appears to be a viable alternative for the development of B5G wireless networks [40]. A recently proposed multicarrier transmission technology - MIMO-OFDM-IM - is obtained by combining MIMO and OFDM-IM transmission techniques and can be considered as an alternative to classic MIMO-OFDM [62]. As part of this scheme, each transmit antenna transmits an OFDM-IM frame in order to increase the data rate at the receiver's end, these frames are separated and demodulated by a novel sequential minimum mean square error (MMSE) detector that considers the statistics of the MMSE filtered signals received.

2.5. Intelligent Reflecting Surfaces

Intelligent reflecting surfaces (IRSs) have gained significant heft because of their ability to provide reconfigurable propagation environments. Absorption, reflection, scattering, and diffraction features can be modified with time. They can be tuned to proffer wireless communication capabilities and specifically helpful in propagation environments with severe blockage. Because of reconfigurable properties, IRSs improve utility in terms of error rates, data rates, communication range, etc.

IRS-enabled wireless communication is a hot research area because of its capability to improve the efficiency of wireless systems via reconfiguring propagation environment. It enables operators to control radio wave characteristics, which helps mitigate the adverse effects of natural wireless propagation. Thus, IRS can be a crucial element in future-generation wireless communication. However, the development relies on transmission schemes and their optimization using classic tools or advanced machine-learning approaches. The impact of IRS on future-generation wireless networks can be fathomed by performance analysis, which will reflect IRSs' efficacy to perform in challenging environments. Several other areas, including cost, implementation, and energy efficiency, are essential from researchers' perspectives [63–69].

2.5.1. An Overview of IRS

IRSs are man-made surfaces with unique wireless communication capabilities. These surfaces are made of electronically-controlled electromagnetic material. An array of passive reflecting elements, classified into antenna-array-based or

metasurface-based structures, can adapt to dynamically fluctuating propagation environments [63, 70, 71]

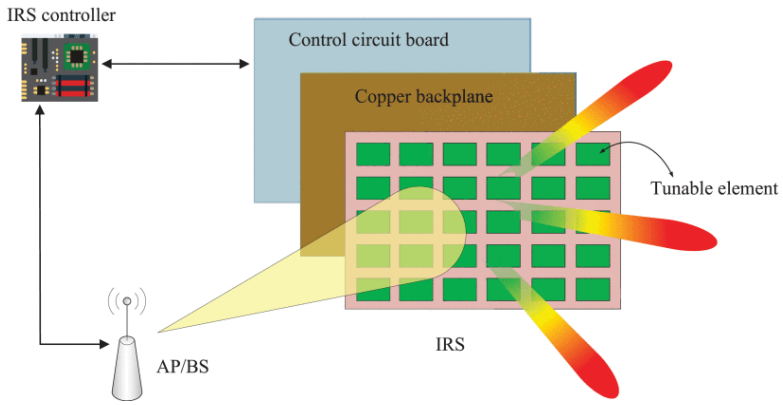


Figure 2.6: IRS architecture. [2]

The distinctive feature of IRS lies in making a controlled environment that facilitates operators to shape and control the electromagnetic response of the environmental objects [4]. Though 5G networks have been rolled out globally, millimetre-wave communication (mmWave), envisioned as a critical technology for 5G networks, is not adopted worldwide. MmWave technology has high bandwidth, which can support high data rates. However, mmWave-aided communication faces grave challenges, such as severe path loss and sensitivity to blockages by cars, trees, and other objects. IRS is an enabling technology that can circumvent these problems. IRS can be installed on large flat areas, such as walls, buildings, or outdoor signages. IRS can reflect radio frequency energy around obstructions and establishes a virtual LoS (line of sight) propagation path between the source and destination [70].

Unlike current systems where operators cannot control the environment, IRS establishes a "smart radio environment" that is software-based reconfigurable platforms. The underlying technology behind this promising concept is meta-surfaces, which are 2D equivalent meta-materials. [4]. Unique characteristics of IRS include:

- They are unaffected by noise.
- They do not need any dedicated energy source.
- They can be easily established on ceilings, indoor spaces, and even on human clothing.

- They can perform at any operating frequency.

IRS possesses several advantages for practical implementations. Its reflecting elements passively reflect the signals without any signal processing operations or transceiver hardware. It is a reason that cost of IRS is significantly lower than conventional transmitters. The passive nature of elements makes its installation easy on walls, signages, street lamps, etc. Furthermore, IRS operates in a full-duplex mode without introducing thermal noise. They gain high spectral efficiency.

2.5.2. Applications of IRS

Some promising IRS applications include IRS-aided multicell networks, IRS-aided physical layer security (PLS) networks, IRS-aided multicast networks, and IRS-aided cognitive radio networks. Cognitive Radio (CR) networks enhance spectrum efficiency by facilitating secondary users to reutilize the same spectrum with primary users. In the conventional approach, beamforming is used to improve the sum rate for secondary users. Still, gains are limited with this approach. By contrast, IRS can be deployed near primary user receiver (PU-RXs). It can mitigate the interference and improve the signal power at secondary users [72–74].

SWIPT (Simultaneous Wireless Information and Power Transfer) is a promising concept for providing cost-effective power delivery, but it is not without challenges. In SWIPT systems, information receivers (IRs) and energy receivers (ERs) have different power supply needs. ERs have higher power needs than IRs. Thus, they need to be deployed near the base station to facilitate sufficient power. Researchers have proposed to deal with this issue by deploying IRS near ERs.

2.5.3. IRS and System Performance: Interference Minimization

Research indicates that IRS enhances system performance by minimizing interferences. IRS significantly outperforms the widely used semidefinite relaxation (SDR) approach by allowing simultaneous communication by multiple users [75]. The authors in [75] have formulated an optimization-based approach for interference minimization using the system model. In addition, they have considered SDR as a benchmark solution and delineated the potential of the IRS in allowing multiple user pairs to communicate simultaneously via the same channel.

2.5.4. IRS Implementation

Literature has investigated two main implementations of IRS, i.e., traditional reflectarrays and metasurfaces. Irrespective of the implementation, IRS must be passive, i.e., it will not emit any power. IRS aims to manipulate transmitted waves, which makes it different from relaying [75].

The simplest method to implement IRS is to use a passive reflectarray. In this, each element has a limited influence on propagated waves. Still, large numbers of elements can control incident waves quite effectively. The practical implementation relies on large numbers of antenna elements. Each element in this system is like a tag in backscatter communication systems. Still, some differences exist. Backscatter communications use reflections to communicate from reflector to receiver, while IRS only helps in ongoing transmission. It does not communicate information from the reflector to the receiver. Though a single element in this system can be considered a tag in the backscatter system, the collective working of all elements creates a potent effect on the incident wave.

A more sophisticated implementation is metasurface-based implementation, a two-dimensional planar made from synthetic man-made material. Its electromagnetic features are lacking in naturally occurring elements. Typically, a metasurface comprises pixels or meta-atoms that are densely spaced subwavelength resonating elements. Their small size proffers excellent freedom in manipulating incident electromagnetic waves. Specifically, a metasurface can have fine control over a scattered electric field by imposing arbitrary quasi-continuous amplitude profiles. Earlier, metasurfaces could not be modified after fabrication because of static meta-atoms designs. Later designs facilitate modifications in the underlying meta-atom structure and the electromagnetic behaviour [75]. These tunable surfaces have gained attraction because of their cost-effectiveness and adaptability. A metasurface-based IRS is made of many tiles. Each tile is a reconfigurable surface. i.e., each individual element of the system possesses functions akin to a reflectarrays. Thus, it allows excellent flexibility in incident wave manipulation. However, most of the research studies have focused on reflectarray-based IRS. Though reflectarray can also fetch impressive results, outcomes can be better metasurface-based IRS. More experimentation is needed to explore how enhanced sophistication would help operators make performance gains.

2.5.5. Spectrum Efficient and Energy Efficient Wireless Communication

IRS technology is promising in attaining both spectrum and energy-efficient wireless communication by integrating low-cost and passive reflecting elements. By manipulating incident wave features such as phase, frequency, and amplitude, IRS reflects manipulated waves to the destination without requiring complex signal processing. [76] have analyzed the asymptomatic optimality of achievable rate in a downlink IRS system. Simulation results revealed that passive beamforming and modulation schemes could achieve asymptomatic optimality.

Multiple base stations in different cells reutilize the same scarce frequency sources in order to maximize spectrum efficiency. As a result, cell-edge users experience a low SINR. The researchers proposed that deploying IRS at the cell boundary can help enhance the signal from the serving base station by minimizing the interference from the other. Simulation results have revealed that the sum-rate achieved by an IRS-aided system with 80 reflecting elements might double that of a system without IRS [70].

IRS-aided Non-orthogonal Multiple Access (NOMA) is a promising technique in which multiple users simultaneously use orthogonal resource blocks. It improves the spectral efficiency of conventional Orthogonal Multiple Access (OMA). Furthermore, NOMA is ideal when all users' channel vectors represent the same angular direction [70].

2.5.6. IRS for Localization and Mapping

IRS can operate as a reconfigurable lens or mirror. A local control unit controls it and adjusts its phase profile. Based on operating modes, IRS can be an anomalous reflector, receiver, or transmitter where the direction of the reflected wave is steerable. IRS concept can be implemented at various wavelengths-from sub-6GHz to 28GHz mmWave bands. At a higher wavelength, benefits can be significant, but technology is immature. Several issues can crop up at 0.1-1THz because of increased sensitivity to blockages, hardware limitations, and atmospheric absorption. These properties of IRS make it an exciting application for localization and mapping [77].

There is a shortage of literature concerning localization and mapping in IRS systems. Recently, Wymeersch, He and Denis have argued that IRS is beneficial for localization and mapping in terms of extensive physical coverage and enhanced

accuracy. However, appropriate models and algorithms need to be developed to fetch the gains. The immaturity of models has hampered progress in this direction, and further investigation and validation are necessary.

Different visions of the IRS can provide distinct merits and demerits concerning localization and mapping. Though authors have shown possibilities of low costs, deploying additional base stations can be challenging. Still, IRS can be a game-changer for localization and mapping applications. Therefore, it deserves heft from signal processing, propagation, communication, and antenna communities.

In sync with the analysis done in [77] performed Fisher's information analysis and proposed a 2-stage localization algorithm. Results showed that when a-priori information is present, positional beamforming can fetch good performance. However, random beamforming is a better option when a-priori information is lacking.

2.5.7. IRS and Security

Wireless communication is prone to security threats such as information leakage and jamming attacks. Physical layer security (PLS) techniques have received much attention in recent years because of their suitability for latency-sensitive applications. Researchers have suggested artificial noise and multiple antennas to maximize the rate of the secure communication link. Still, issues can crop up when eavesdroppers and legitimate users have correlated channels or eavesdroppers are situated closer to the base station than legitimate users. These conditions can affect the achievable secure rate. Deploying IRS can increase the signal power for legitimate users while mitigating information leakage to eavesdroppers [78, 79].

The authors in [70] have studied the IRS-assisted OFDM system. They have proposed a practical transmission protocol to reduce training overhead in channel training. They proposed a unique grouping method where each group is comprised of adjacent IRS elements sharing a common reflection coefficient. This grouping method facilitated the researchers to propose a protocol wherein only the combined channel of each group needed to be estimated as it reduced training overhead. Simulation results revealed that the proposed design colossally improved OFDM link rate performance compared to the scenarios where IRS was not used.

2.6. Chapter Summary

This chapter presented the key concepts of ML and DL for the physical layer and their potential. Next, we discuss some general background about DL concepts, we focus our work on DL for the physical layer of the communication system. Moreover, a key literature review of ML and DL on wireless communication systems. Finally, the IRS concept, application and implementation were presented.

Chapter 3

Applied Autoencoder(AE) Techniques

Autoencoder (AE) is a type of artificial NN that has been mainly implemented throughout this chapter. This chapter aims to describe the fundamental aspects of AE and how it can be applied in wireless communication. Section 3.1 introduces the ML and AE in wireless communication. Section 3.2 explain the neurons in AE while the activation functions are covered in section 3.3. Section 3.4 illustrates the architecture of AE followed by types of AE in section 3.5. Section 3.6 introduces the applications of AEs, while section 3.7 illustrates how the AE works and learns in section 3.8. Section 3.9 explains the gradient descent algorithm followed by the backpropagation algorithm in section 3.10. Section 3.11 presents some basics for DL AE and its training. Section 3.12 investigates the different configurations of the AE with the aim of understanding its potential before extending the existing research. Section 3.13 presents the simulation and results of AE under different channels. Finally, section 3.14 summarize the chapter.

3.1. Introduction

ML has become quite common in the modern era of digitization and technology [80, 81]. It finds extensive applications in weather forecasting, anomaly detection, speech recognition, and content filtering in social networks [81, 82]. For decades, ML algorithms focused only on supervised learning. Then came the notion of DL which carries the capability to automatically extract features and patterns from data [81]. At its core, the DL functions through multiple levels of representation

with the help of nonlinear modules or functions for representation of the preceding level. The recent advances in the DL field have given rise to artificial NNs that carry the built-in capability to model data points through multiple layers of hidden representations. These NNs are referred to as AE.

The basic functioning of the AE involves a layer-wise pre-training method involving specific features and representations. Once all individual layers are pre-trained, they are then integrated to form an AE that requires fine-tuning for improving the functioning of the overall DL model [80]. The concept of unsupervised learning lies at the heart of the AEs in which some sort of relationship among input functions enables them to learn and adapt the model parameters [80]. Moreover, the presence of the bottleneck or limiting constraints is essential for constructing features that could force learned compression of the input data [80]. The ideal AE balances in terms of sensitivity to the inputs to accurately build the reconstruction and insensitivity enough to an extent that the model simply doesn't repeat the training data [80]. Due to this tradeoff. The model focuses only on the variations in data to reconstruct the input.

3.2. Neuron in Autoencoders

The term NN or neurons refers to the links associated with the inputs and outputs in the DL algorithms [83]. The AE learns significant patterns from the data by minimizing the reconstruction error between the input and output data. In AEs, the number of neurons present in the output layer is exactly equal to the number of neurons in the input layer. AEs have a large number of nodes in the input and output layers [83]. The training is carried out to redesign the input vector instead of mapping it to a target label. The number of neurons in a hidden layer is less as compared to the input layer neurons so that encoding of the data is performed in a lower-dimensional space and abstract features are derived.

The role of neurons in DL algorithms and AEs is an estimation of the representations for updating the system. The measurement of the loss function and certain regularization terms requires designing an objective-oriented representation function. Similarities and differences among hidden neurons provide input to the regularization of the estimated neural networks and AEs. Robustness and development of discriminative features are essential for building reliable AEs for various applications. The activation functions and other features

make deep learning models for AEs quite meaningful. The relationship between the input and output parameters is important to establish frameworks for the training of data and feature extraction.

3.3. Activation Functions

Activation is the mapping capability of the AE that uses the nonlinear transformations for associating the encoded coefficients into a specific range [84]. The application of the activation functions lies in the mathematical formulation of the DL algorithms and AEs. In modern digital networks, the activation functions are generally nonlinear which allows it to learn more complex functional mappings between input and outputs. Moreover, the constraints associated with the linear functions are avoided with the help of such complex modelling of activation functions.

There are various activation functions commonly used in the AEs domain. For example, the sigmoid activation function $\sigma(x)$ maps a real number x into an interval between 0 and 1 [82]. It could be expressed as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{3.1}$$

The hyperbolic tangent function (tanh) is another example of the activation function that maps the real numbers to the range between -1 and 1 [82]. Unlike the sigmoid function, the output passes through zero. Mathematically, we have:

$$\tanh = \frac{1 - e^{-2x}}{1 + e^{-2x}}. \tag{3.2}$$

Learning of the deep networks and AEs could improve in utilizing the recurrent models [85]. The concept of memory cells could be employed for mapping input and output features [85]. The connections could be easily set up between representations for the dynamic adaptability of the NNs. The standard nonlinear networks could be dynamically updated to adapt to the innovative features and provide the required accuracy for AEs. The main objective of the development of the activation functions of the AEs is to derive the loss terms that minimize the derivatives of the hidden layer activations, concerning the input training examples. This strategy penalizes instances where a minor change in the input could lead to

a larger change in the encoder nonlinear mapping.

3.4. Architecture of Autoencoders

The architecture of the DNN is built by concatenating the basic models hierarchically [84]. Due to their simple modelling and representation, the AEs have found various applications. The NN architecture comprises output representations in the form of hidden layers. The depth of the network is ascertained with the help of a group of experiments and training of the model. The inputs of the successive layers are linked to the output of the previous layer [84]. The performance of the AE is dependent on its architecture as research showed that stacked AE is much better as compared to the single hidden-layer model [84].

The architecture of the AE comprises several parameters for the representation of the input and output. The parameters like code size, number of layers, loss function, activation function at each layer, regularizer, and optimizer determine the architecture of the AE. The DL models don't require labelled information for data. Training of AEs in the field of DL is difficult due to the diversity of the architectural features like differences in the magnitudes of the gradients in lower and higher layers, the landscape of the objective function is tough for SGD for finding the good choice for local optimum. Moreover, DL networks involve various parameters for remembering the training data. Pretraining the AEs, the architecture of the NN scheme is considered reliable when it involves accurate linkages and adaptable nodes between inputs and outputs.

3.5. Types of Autoencoders

There are various types of encoders as described below. The types of encoders stem from various architectures that can be employed for hidden representations of deep non-linear neural networks.

3.5.1. Sparse Autoencoder

Sparsity is an important concept in the field of DL and AEs. In general, the sparse AEs have the characteristic of getting only a few neurons, roughly 5%, for any given computation in the encoding layer. For forcing sparsity, one could easily introduce sparsity measure and feature its minimization during the training of

the neurons [86]. The architecture of the sparse AE is overcomplete in the sense that it allows many hidden units in the code, but most of the hidden neurons result in only a minor activation. For every hidden node, the average activation value should be nearly zero [82]. The goal is to activate neurons only for a small proportion of the training examples [82]. Since the samples have different features, the activation of the neurons must not be held similarly in all neurons and must be coordinated.

The training of the sparse encoder includes a penalty term in its loss function for minimizing deviation from the desired sparsity. With a higher level of sparsity, the network seemed to capture local fragments of digits, and it required learning a more comprehensive representation of each digit [82].

3.5.2. Denoising Autoencoder

The NNs could be trained by adding noise to the network and then mapping the inputs to the outputs. The noise in the input layer provides a disturbance to the system for enforcing its adaptability towards actual and undesired inputs [82]. The inputs provided to the system are corrupted and the system automatically adapts its representations to filter input from noise and establish related neurons. The loss function associated with denoising input involves the corrupted version instead of the original input to minimize the error of mapping. Traditionally, the AEs minimize the following loss function:

$$L(x, g(f(x))), \tag{3.3}$$

where L is the loss function that penalizes the function g(f(x)) from being dissimilar with x. Ideally, the convolution of g and f would yield an identity function. The denoising AE minimizes the following function:

$$L(x, g(f(\widetilde{x}))). \tag{3.4}$$

In the above equation, $\widetilde{x}$ is the copy of the original function x that has been corrupted by some form of noise. The denoising AEs must remove the corruption from the input instead of copying inputs to outputs. Denoising AEs provide an example of how useful properties could be extracted from NNs by simple reconstruction of systems. Moreover, denoising systems also provide insight

into high-capacity models as long as preventing them from learning the identity function.

The application of denoising AE could be found in the image processing systems in which the original image is corrupted by glitches or noise [87]. Reconstruction of the image requires encoding the corrupted input to the representations of the system and identifying features from it [87]. After minimizing the loss function and training the system to minimize noise, the original image can be extracted at the output.

3.5.3. Stacked Autoencoder

As the name suggests, the stacked AE is a deep representation of the simple encoder with layer-wise pre-training of the system. It functions by removing the top layer of the simple encoder while the bottom and middle layers stay intact [84]. The value of the middle layer could be named h_1, and the corresponding layer could also be called from the same name. Secondly, with the help of layer h_1 and the old middle layer as inputs, two more layers are added as h_2 and $\widehat{h}_1$ [86]. The resulting three layers could be considered as stacked encoders and are trained according to standard rules. The same process could be repeated to add more layers to get more representations and stacked neural network structure or deep AE.

The decrease in the number of features in successive layers occurs according to diverse heuristic architectures or with the help of regularization parameters. The components of the deep AE were parametrized in the same manner for each training method. The input data fed into the AE could be normalized to the original values.

3.5.4. Contractive Autoencoder

The regularization of AEs could be performed in a variety of ways. In contractive encoders, a specific penalty Ω is imposed on the parameters of the system such that the function doesn't change much with slight variations in the input. Since this penalty is only applied to the training examples, it makes the AE learn the features that capture information regarding the training distribution. The AE regularized in this manner is called a contractive AE. It functions by defining an explicit regularizer on the code such that the derivatives are as small as possible.

3.5.5. Deep Autoencoder

The deep AEs are like contractive AEs in terms of functioning and model implementation [88]. Each layer of the deep AE is pretrained to form a series of single-layer AEs. Since each layer is independent in terms of self-learning, the overall model is quite different from the architecture in which all layers are pretrained in parallel. The penalty is imposed on the Jacobian of the deep model, but still, it possesses several desirable features.

The deep AE contains at least one hidden additional layer inside the encoder itself. It can approximate any kind of mapping from input to code from the given hidden units. The main advantage of depth is an exponential reduction in the computational cost of representing source functions. Depth can also minimize the amount of training data required to learn some functions. Deep AEs yield a high degree of compression than the corresponding linear type of AEs. A common method to train a deep AE is rigorous pretraining the deep architecture by training a hierarchy of shallow AEs. The limitation of the deep AE is information loss should be minimized through the compression hidden layer concept. Due to the minimization of the loss function, the deep AEs function very well.

3.6. Applications of Autoencoders

AEs are successfully applied in terms of dimensionally reducing information retrieval tasks. Dimensionality reduction is one of the interesting applications of DL and represents self-learning regimes. In one of the applications, the researchers trained a set of hidden layers by using their weights to form a deep AE [89]. The overall structure culminated itself in the bottleneck of 30 units. The self-learning model was qualitatively easier to implement and then relate to the underlying categories, with these features depicting themselves as well-separated clusters.

The applications of low-dimensional models are numerous. Models of smaller spaces use less memory and computation time. Many different forms of dimensionality reduction utilize semantically related examples. The hints offered by mapping to the lower-dimensionality space help in the generalization of the models.

One major application of AEs is information retrieval from a huge database that resembles a query entry. This task is efficient not only in terms of

dimensionality reduction, but the computational time for searching index terms could be very small for certain stacked implementations. Specifically, we can easily train the AE model that plays the function of dimensionality reduction for storing the database entries in a kind of hash-table for mapping of binary code vectors to different entries. The hash table would allow information retrieval by returning all kinds of database entries that carry the same binary code as the index search term. This approach towards information retrieval with the help of dimensionality reduction and binarization is called semantic hashing.

For producing binary codes for semantic hashing, one typically utilizes an encoding function with sigmoid on the final layer. These units must be trained to provide binary outputs for the given input values. One trick to accomplish this objective is the addition of additive noise just before the nonlinearity for the training period. The magnitude of the noise must have an increasing trend over time. For minimization of noise to the overall system, the magnitude of the inputs is increased to the function, until saturation occurs [90].

3.7. How do the Autoencoders Work?

AEs work by self-learning from a set of modular inputs, big data, NNs, and DL schemes. The overall process of coding the inputs for learning the process is called encoding. The concept of self-learning is important for pretraining a set of inputs against which the model sets the base-level implementation. After pretraining, the self-learning of the AEs improves with time by processing the inputs.

The objective of the AEs is the implementation of functions that could adapt according to the environment and the input. The logic associated with the AEs ensures that the system's adaptability to the conditions improves with time. For quantitative assessment of the AEs, the loss function is important. For different categories of AEs, the loss function is designed that must be minimized to reduce the error between the expected output and the actual output.

The function of AEs must also be understood from the perspective of layering or DL. The contractive or deep AEs train layers independently to form hidden layers and the resulting stacked network finds various applications. The learning of AEs could improve by using different schemes for DL. When you consider applications of AEs in binary hash applications, the learning schemes could be diverse [24, 91].

3.8. How do Autoencoders Learn?

Learning of AE occurs with the help of NNs by recognizing the pattern among the structural layers. The learning of AEs occurs with the help of pretraining of inputs and outputs mapping to each other. Designing AEs involves imposing the NN, which imposes the bottleneck in the architecture. The resulting framework is a compressed knowledge representation of the original input. The input features could be dependent on each other. The applications in which the training set involves independent inputs couldn't utilize AEs for training functions.

For training the AEs, a set of inputs could be chosen for representation and self-learning. The task is generating a model for the reconstruction of the original inputs by learning the patterns and dependencies among the inputs. The output of the AEs could be a reconstructed version of inputs by minimizing the error between actual inputs and reconstructed inputs. The loss function determines the mathematical formulation of error generated due to representation or self-learning.

The bottleneck is an important function of the AE that determines the presence of the information source as the representable gradient. Without the presence of the information bottleneck, the system could easily learn to memorize the input values by transferring these values along with the network. The bottleneck allows forced compression of the data and traversing of information from input through the channel to the output. The activation functions work in layers to help the reconstruction of inputs and perform a learned representation of the original input data. The key to the representation or the learning function is to be sensitive to the input data for accurately building the reconstruction and insensitive enough to those inputs that involve simple memorization or overfitting the training data. The tradeoff allows the system to maintain the variations in the data instead of copying the system values and generating their image. The terms included in the model for representation provide the trade-off between different objectives [92,93].

3.9. Gradient Descent

It is an optimization algorithm in the domain of neural networks and AEs that involves self-representation of data. It is based on the convex optimization problem and manipulates the system parameters to minimize the given objective function. The system works by defining a set of input parameters and adjusting them such

that the objective function is maximized or minimized.

The concept of the gradient is generic that measures how much the output changes if you change the input slightly in a small increment. It simply calculates the variations in the weight of the parameters of the function. The higher the gradient, the steeper the slope of the function and the faster the system can learn. If the slope of the gradient is zero, the model will stop learning and the self-learning will be ineffective in the case of AEs. This concept is important for ML algorithms that involve the learning curves and functions for which gradient descent could be used for minimizing the given objective function.

Consider the problem of minimizing the cost function in the form of losses incurred due to the representation of the inputs in the AE. The loss function involves errors concerning differences in input and reconstructed input. For minimizing the loss function, the parameters could be tweaked such that the reconstruction error could be optimized. For finding the right values of the convex optimization problem, the variables could be tweaked in a manner that minimizes the overall functional error. The learning rate could be set to achieve the desired highest or lowest value [94–96].

3.10. Backpropagation

It is the main essence of the neural network training that is employed in the AEs. It is the method of tweaking the weights of the AEs based on the rate of error in successive iterations. Proper tuning allows the reduction of error rates and makes the model reliable by improving its generalization.

Backpropagation is the short form of backward propagation of errors, and it represents the standardized method of training artificial NNs. With the help of backpropagation, we can measure the gradient of the loss function concerning the weights in the network. The system requires studying the group of inputs and activation values for developing the relationship between the input and the hidden unit layers. It helps in assessing the impact that a given input variable has on different kinds of network outputs. It is especially useful for DNNs, by working on error-prone projects such as speech recognition and image processing. It takes advantage of the chain and power rules that allow backpropagation to function with any number of inputs [25, 97].

3.11. Basic Deep Learning Based Autoencoders

An autoencoder as feed-forward NN, or multilayer perceptron (MLP) with L number of layers, can be described as mapping $f(r_0;\theta) : \mathbb{R}^{N_0} \mapsto \mathbb{R}^{N_L}$ of the input vector $r_0 \in \mathbb{R}^{N_0}$ to the output vector $r_L \in \mathbb{R}^{N_L}$ via L iterations:

$$r_\ell = f_\ell(r_{\ell-1};\theta_\ell), \qquad \ell = 1, ..., L, \tag{3.5}$$

where $f_\ell(r_{\ell-1}) : \mathbb{R}^{N_{\ell-1}} \mapsto \mathbb{R}^{N_\ell}$ is the mapping on the ℓth layer. This mapping depends on the output vector $r_{\ell-1}$ from the previous layers and on a series of parameters θ_ℓ. We define $\theta = \theta_1, ...\theta_L$ in order to denote the set of parameters in each layer of all L layers. The ℓth layer, called dense layer or fully-connected layer if $f_\ell(r_{\ell-1};\theta_\ell)$ can be written as:

$$f_\ell(r_{\ell-1};\theta_\ell) = \sigma(W_\ell r_{\ell-1} + b_\ell), \tag{3.6}$$

where the weight is $W_\ell \in \mathbb{R}^{N_{\ell-1}\times N_\ell - 1}$, the bias is $b_\ell \in \mathbb{R}^{N_\ell}$ and $\sigma(.)$ is the activation function. The set of parameters in the dense layer are $\theta_\ell = [W_\ell, b_\ell]$. Table 3.1 shows a list of different layer types with mapping functions and parameters. All the L layers with random mapping generate random mapping each time. For example, the noise layer involves adding a Gaussian noise vector to the input a zero means and variance matrix $\beta I_{N_{\ell-1}}$. Therefore, it generates a different output, for the same input each time. The activation function $\sigma(.)$ in equation 3.7 is a non-linear function that is important because, without non-linearity, there will be no advantage of heaping different layers on top of each other. The activation function is generally applied individually to each entry vector. Table 3.2 lists some frequently used activation functions. Generally, we train the NNs by labelling the training data, i.e. a pair of input and output vectors $(r_{0,i}, r^*_{L,i}), i = 1, ...S$ where $r^*_{L,i}$ is the targeted output, and $r_{0,i}$ is the input of the NN. Our goal is to attain the minimized loss

$$L(\theta) = \frac{1}{S}\sum_{i=1}^{s} l(r^*_{L,i}, r_{L,i}), \tag{3.7}$$

where $l(u, v) : \mathbb{R}^{NL} \times \mathbb{R}^{NL} \mapsto \mathbb{R}$ is the loss function, and $r_{L,i}$ is the prediction of the NN, while $r_{0,i}$ is the input. Table 3.3 shows several relevant loss functions. One of the most popular algorithms for finding suitable sets of coefficients is the

stochastic gradient descent method (SGD) [98], which begins with some initial random values of $\theta = \theta_0$ and then iteratively updates θ as

$$\theta_{t+1} = \theta_t - \eta \bigtriangledown \widetilde{L}(\theta_t), \tag{3.8}$$

where $\eta > 0$ is the learning step while the result of the loss function calculated from a random mini-batch of the training set is $\widetilde{L}(\theta_t)$. $S_t \subset 1, 2, ..., S$ in each training epoch, i.e.

$$\widetilde{L}(\theta) = \frac{1}{S_t} \sum_{i \in s_t} l(r^*_{L,i}, r_{L,i}). \tag{3.9}$$

When we select a small S_t compared with the S, the calculation complexity will reduce and continue reducing the variance of the weights. There are many different kinds of SGD algorithms that can modify the learning step dynamically to promote convergence [99]. In this work, we have used adaptive moment estimation (Adam) [100] to optimize and update the parameters.

Table 3.1: A list of layer types.

Name	$r_\ell = f_\ell(r_{\ell-1}; \theta_\ell)$	θ_ℓ
Dense	$\sigma(W_\ell r_{\ell-1} + b_\ell)$	$[W_\ell, b_\ell]$
Noise	$x + \eta$	none
Normalization	$\frac{x}{\sqrt{E.[x^2]}}$	none

Table 3.2: A list of activation functions.

Name	$[\sigma(u)]_i$	**Range**
ReLU	$max(0, u_i)$	$[0, \infty)$
tanh	$\tanh(u_i)$	$(-1, 1)$
sigmoid	$\frac{1}{1+e^{u_i}}$	$(0, 1)$
softmax	$\frac{e^{u_i}}{\sum_j e^u_j}$	$(0, 1)$

Table 3.3: A list of loss functions.

Name	$l(u, v)$
MSE	$\|u - v\|_2^2$
Categorical cross-entropy	$-\sum_j u_j log(v_j)$

The power of AE is derived from its nonlinear nature. AE can learn rather powerful representations of the input data in lower dimensions with

little information loss when nonlinearity is added (such as nonlinear activation functions)

3.12. End-to-End Communications System Based Autoencoders

As we introduced in the previous chapter, using AE based on DL end-to-end communications system has garnered interest due to it is flexibility and simplicity. In this section, we have investigated the different configurations of the AE with the aim of understanding its potential before extending the existing research. We investigate the AE performance based on end-to-end communication systems, and we start this chapter by explaining the concept of AE as proposed by [3] and its application, then we formulate the AE system model and its layout. Next, we present the implementation of the system before presenting the results.

3.12.1. System Model

A communication system consists of a transmitter, a channel and a receiver, as shown in Fig. 3.1 The transmitter sends one message out of M possible messages $s \in M = 1, 2..., M$ to the receiver making n use of the channel. In general, the hardware of the transmitter imposes power constraints on x, such as energy constraint $x_2^2 \leq n$ or an average power constraint $\mathbb{E}[x_i^2] \leqslant 1 \forall i$.

Figure 3.1: A structure of a simple communication system.

The communication system will represent each message s in $k = log_2(M)$ using a communication system rate $R = k/n$ [bits/channel use]. The conditional probability density function $p(y|x)$, can describe the channel as it causes distortion to the transmitted symbols so that the receiver, upon the reception of y, it produces $\widehat{s}$ of the transmitted message s.

3.12.2. End-to-End Learning Using Autoencoders

One of the artificial neural network (ANN) types is the AE which uses an unsupervised ML algorithm to reconstruct its input at the output. Originally, the main goal of an AE is to reduce the dimensionality of its input, which allows

for a reconstruction of the output in an unsupervised manner with minimum error. However, the main purpose in our case is to learn to represent input x of the message s which is robust then the transmitter can recover the message with a small probability of error. The AE consists of a number of layers forming a representation of a DL network, or there can be multiple hidden layers that form a DNN. The hidden layers, h describe a code used to represent the input data x. The hidden layers are constructed to learn about representation from its input dataset, so it can copy it into the output layer. In order to maximize the end-to-end recovery accuracy, the weights/parameters of the DNNs are trained in a supervised learning manner. It is possible to simulate the effect of wireless channels by adding noise to the hidden layers in order to obtain robustness to noise. As shown in Fig. 3.2 the network of an AE as an end-to-end communication system consists of three parts:

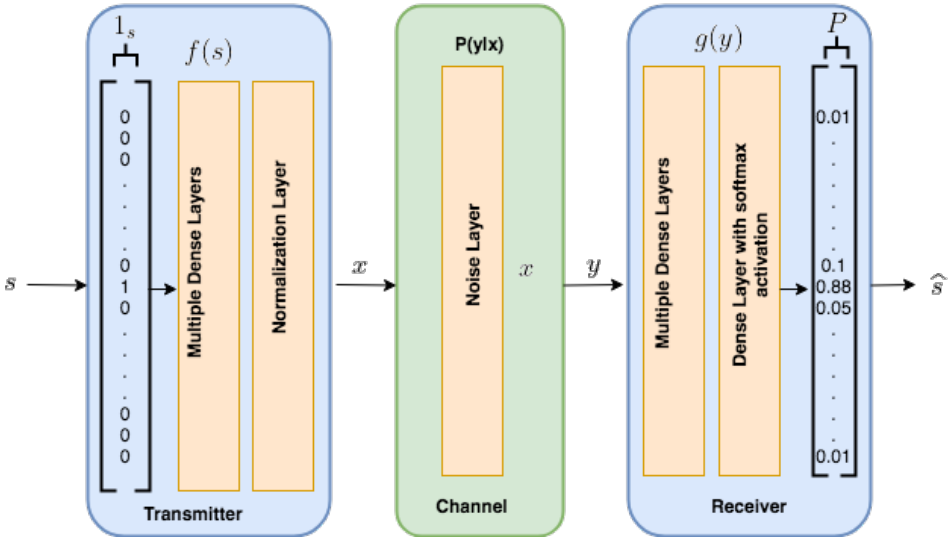


Figure 3.2: A communication system represented as an autoencoder.

Encoder:

the transmitter wants to transmit one input symbol s out of $M = 2^k$ messages. The input symbol is encoded as a one-hot vector $1_s \in \mathbb{R}^M$ as M-dimensional vector, then fed into the input layer after the encoder recognizes a transformation of $f : \mathbb{R}^k \to \mathbb{R}^n$ in order to transfer $\widehat{s} \in \mathbb{R}^k$ to the signal $x = f(s) \in \mathbb{R}^n$. As shown in Fig. 3.2 the encoder consists of multiple dense layers; a dense layer of M units with a ReLU activation function and a dense layer of n units with a

linear activation function. After, at the normalization layer, an energy constraint $\mathbb{E}[\|x\|^2] = n$. As mentioned above, the use of AE will add some redundancy by making $n \geq k$ for x of the message s so it can fight the channel impairments mapping x to y. The end-to-end AE uses the communication rate $R = k/n$ [bits/channel use].

Channel:

The channel layers come after the encoder layer, and its a flat fading channel that consists of fading layer and a noise layer. The channel layers add some perturbations to $\mathbf{x}$, which is the output of the encoder.

$$\mathbf{y} = \mathbf{h} \odot \mathbf{x} + \mathbf{w}, \tag{3.10}$$

where $\odot$ stands for the element-wise multiplication, $\mathbf{h} \in \mathbb{R}^n$ is the impulse response data fed to the fading layer, $\mathbf{x} \in \mathbb{R}^n$ is the encoder output and $\mathbf{w}$ is the AWGN characterized as distribution $\mathbb{CN}(0, \beta^2\mathbf{I})$. The channel is represented by AWGN due to the assumption that the main source of the noise is on the receiver side [101]. The channel uses a fixed variance $\beta = (2RE_b/N_o)^{-1}$, where (E_b/N_o) is the energy per bit E_b to the noise power spectral density N_o ratio.

Decoder:

The decoder is on the receiver side of the communication system that consists of two dense layers of M units. The decoder transforms $g : \mathbb{R}^n \to \mathbb{R}^k$ to the noise signal $\mathbf{y} \in \mathbb{R}^n$ to message $\widehat{\mathbf{s}} \in \mathbb{R}^k$. The decoder uses ReLU as an activation function on one dense layer and a softmax activation function on the other dense layer. The result of the decoder is a probability distribution with M probabilities and the highest index probability is converted to a decoded message $\widehat{s}$. Training an end-to-end AE can be done by minimizing the reconstruction loss function. The loss function L, such as categorical cross-entropy or MSE, is used to penalize the change of $g(f(x))$ to the input x.

$$\mathcal{L} = -\sum_{i=1}^{N}(s_1 \log \widehat{s}_1^{(i)} + s_2 \log \widehat{s}_2^{(i)} + ... + s_M \log \widehat{s}_M^{(i)}), \tag{3.11}$$

where the original message has been sent by the transmitter is $s = (s_1; s_2; s_3; ....; s_M)$ and the decoded message at the receiver is $\widehat{s} = (\widehat{s}_1; \widehat{s}_2; \widehat{s}_3; ....; \widehat{s}_m)$. For training the AE, we used Adam optimizer to execute the SGD as mentioned earlier. Table 3.4 lists the layout of AE model.

Table 3.4: Layout of the end-to-end autoencoder

Component	**Layer**	**Activation Function**	**Output Dimension**
Transmitter	Input		M
	Dense	ReLU	M
	Dense	Linear	M
	Normalization		n
Channel	Fading		n
	Noise		n
Receiver	Dense	Linear	M
	Dense	ReLU	M
	Dense	softmax	M

3.13. Simulation and Results

In this section, we present the AE implementation and results as an end-to-end communication system in comparison with a conventional communication system. Our performance benchmark is the standard communication system using BPSK modulation and a Hamming code as the coding method. The AE model we implemented was proposed by [3]. For implementing the model, we used Python with Keras [102], and Tensorflow [103], as its backend for training and testing uses the parameters in Tabel 3.4.

The AE has the input symbol s, which has a M finite state, while the channel with some random features acting on the encoded signal x then generates the output signal y. We can consider these random features of the channel as a sort of regularisation to prevent the problem of overfitting. The training was done on different channels as follows: AWGN channel, Rayleigh channel and Rician channel. All the training over these different channels was done with 50 epochs. Each of these epochs uses a different training symbol of 10^6 and at a fixed Adam optimisation learning rate of 0.001, and at fixed $E_b/N_o = 7$dB.

3.13.1. AWGN Channel

To evaluate our coding performance, we began training our communication system over the AWGN channel as it is widely used in coding theory.

Fig. 3.3 shows the bit error rate (BER) of the end-to-end AE and Hamming Code with the binary phase-shift keying (BPSK) modulation operating rate at $R = 4/7$. Also, to achieve a clear code gain, we simulate uncoded BPSK (4,4). To achieve optimal performance, the Hamming code uses maximum likelihood decoding (MLD). The AE (7,4) at rate $R = 4/7$ is almost the same BER as the Hamming (7,4). While at rate $R = 4/7$, the AE (7,4) outperforms the BER of the uncoded BPSK. These results indicate that the AE without any prior knowledge of the encoder and decoder function achieves the same performance as the Hamming (7,4) and better performance in (4,4). This result means that improving the complexity of coding and decoding will increase the coding gain.

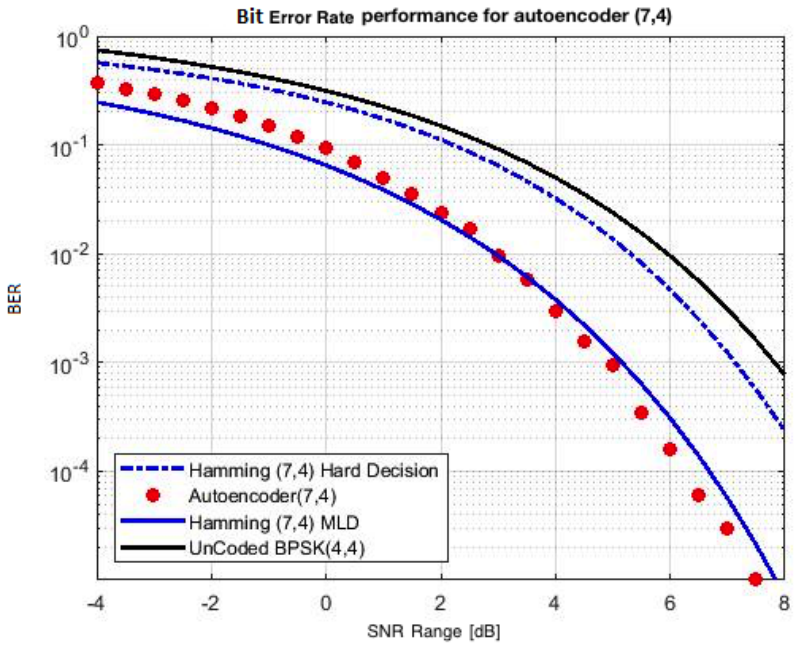


Figure 3.3: BER versus (E_b/N_o) for the Uncoded BPSK (4,4), Hamming (7,4), Autoencoder (7,4) and Hamming (7,4) MLD.

Fig. 3.4 shows a comparison of a communication system for (2,2) and (8,8). We notice that the BER of the AE is almost the same as the uncoded BPSK (2,2). On the other hand, the AE (8,8) outperforms the uncoded BPSK (8,8). From this comparison, we can conclude that the AE achieved some coding gain by learning

some join coding and modulation schemes.

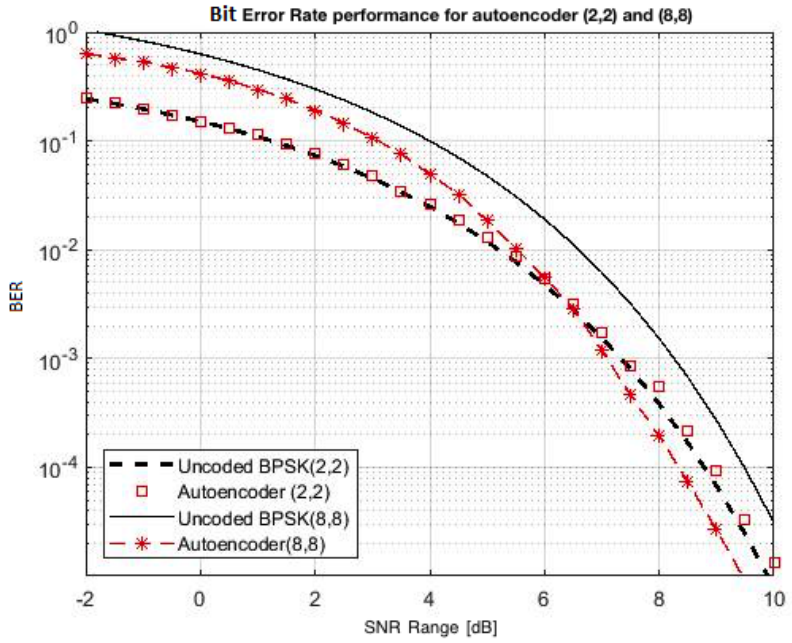
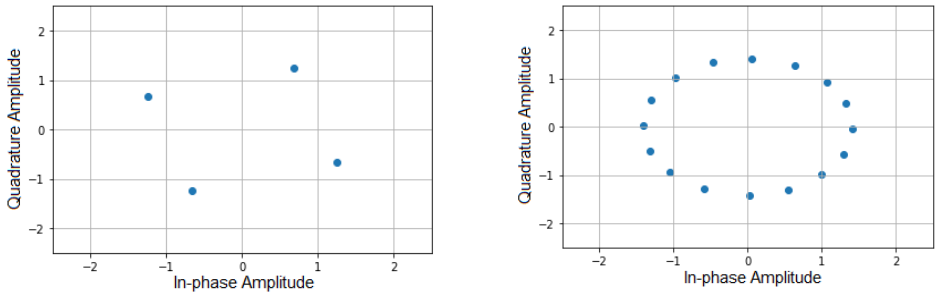


Figure 3.4: BER versus (E_b/N_o) for the Uncoded BPSK(2,2), Autoencoder (2,2), Uncoded BPSK(8,8) and Autoencoder(8,8).

Learned Constellations

After training the AE model, we can split the system into three parts: an encoder, a channel and a decoder. The encoder part is where the encoded symbol is generated at the transmitter before it sends through the channel and then to the receiver where the decoder is implemented to recover the message. Fig. 3.5 (a) shows the scatter plots x for all messages with different values of (n, k). The simple (2,2) system with some arbitrary rotation, learned constellations similar to (QPSK). In Fig.3.5 (b), we can observe that the (2,4) system leads to a rotated 16-PSK.



(a) (b)

Figure 3.5: Constellations produced by autoencoders using parameters (n, k): (a) (2, 2) and (b) (2, 4).

Effects of Training (E_b/N_o) Values and Hyperparameter Tuning

During the training of the AE, we notice that the architecture of the deep model and parameters of training, such as the length of the encoded signal, batch size and learning rate, are affecting the performance of the training model. Fig. 3.6 shows a comparison of the BER of the AE (2,4) to the AE (8,4). We notice that the BER had become lower when the length of the encoded signal x grows longer. This means that reducing the source transmission rate resulted in BER gain. While growing the length of x, the code gain decreases until it vanishes completely.

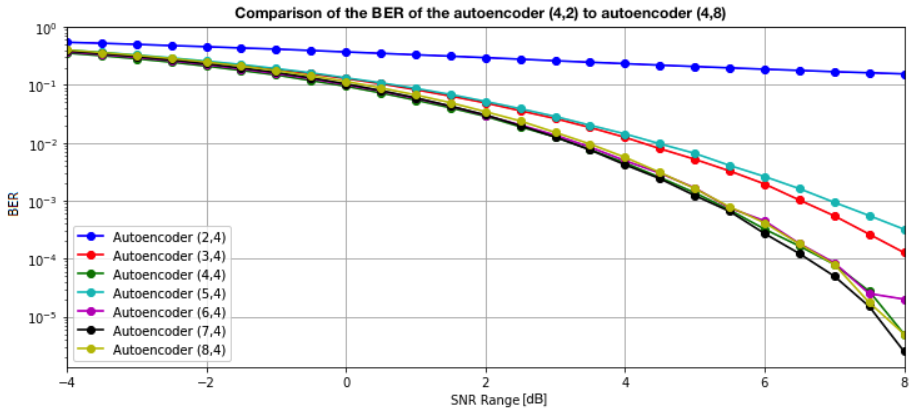


Figure 3.6: BER versus (E_b/N_o) for the Autoencoder (k,4), k=2, 3, 4, 5, 6, 7, 8.

From Fig. 3.7 we can observe that when we fixed the (E_b/N_o) value of 7dB, the small batch size of 100 resulted in improving the BER performance compared with the large batch size of 1000.

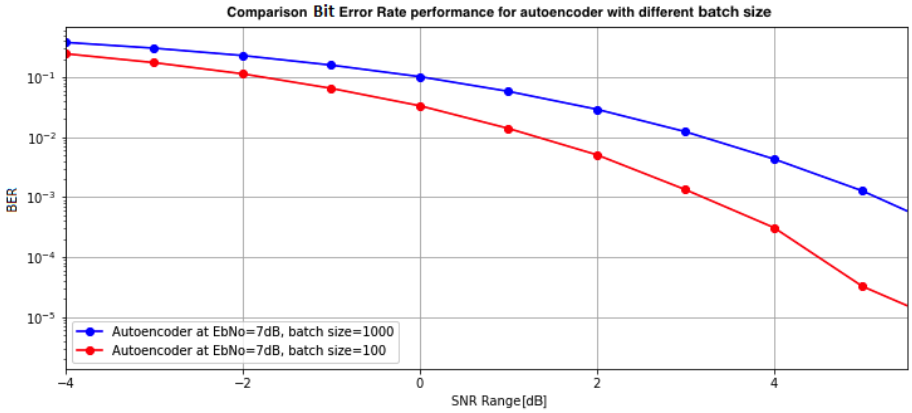


Figure 3.7: BER performance for different training batch size values.

Fig. 3.8 shows that at a fixed batch size of 1000, the performance of the trained autoencoder is better with a low (E_b/N_o) value at 4dB compared with 7dB.

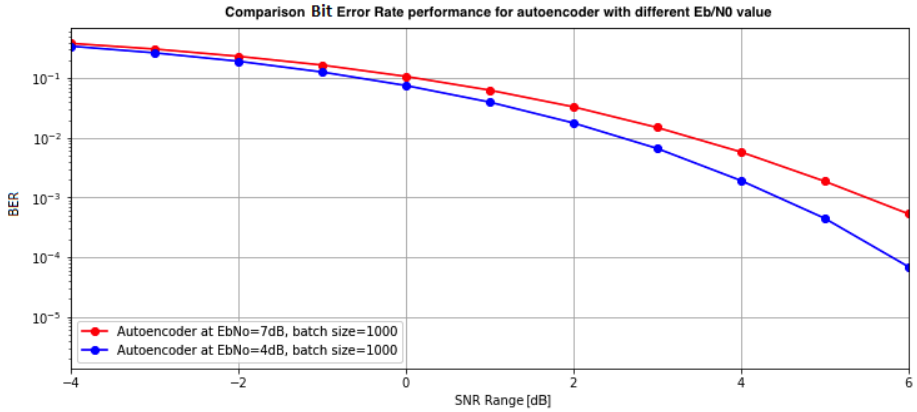


Figure 3.8: BER performance for different training (E_b/N_o) values.

3.14. Chapter Summary

In this chapter, we have introduced and presented the key concepts of AE before we start to investigate the different configurations of the AE with the

aim of understanding its potential before extending the existing research. In addition, we evaluated the implementation of machine learning in end-to-end wireless communication based on existing neural network technologies in the AE. In particular, we compared the BER performance for SISO systems of the uncoded communications systems with the equivalent AE models. Our simulation results show that the performance of the AE-based systems is comparable to the conventional systems. We also investigated the effect of training (E_b/N_o) and hyperparameters on the AE system and the effect of these parameters on the performance of the system. Then, we trained the AE model using various wireless channel types and compared the AE performance under AWGN. The AE has shown better performance than the Hamming code in BER performance, with less processing complexity compared with the conventional system. Moreover, the parallel architecture of the DL allows the processing of information faster. This has proven the power of the DL in the physical layer and in terms of error performance in the next chapter.

Chapter 4

Design of A Deep Learning-Based Detector for IM-MIMO-OFDM

This chapter is organised as follows: section 4.1 introduces the chapter and states the main contributions. Section 4.2 states the related works, followed by deep learning-based detection in section 4.3. Section 4.4 introduces the experiment setup and training procedure. The numerical simulation comparing the results with other works is presented in section 4.5. Finally, section 4.6 summarises the chapter. The work in this chapter has been published in P.1.

4.1. Introduction

Deep learning (DL) is playing an increasingly important role in the design of next-generation communication systems. In this chapter, we apply DL algorithms to enhance signal detection and performance of multiple-input-multiple-output (MIMO) based orthogonal frequency-division multiplexing (OFDM) systems with index modulation (IM). The proposed detector termed DLIM is used as fully connected layers of a deep neural network (DNN) and adopted to achieve minimum bit error rates (BER) in IM-MIMO-OFDM over Rayleigh wireless channels. To show the enhancement of the proposed algorithm, the DL model is trained initially offline using data generated from simulation based on common statistical wireless channel models. DLIM is then adopted to recover the online transmitted data. Simulation results confirm that the proposed DLIM can detect the transmitted

symbols with a performance comparable to near-optimal BER in a shorter runtime than required by the existing classical detectors.

4.2. Related Works

Recently, IM has attracted attention as means to achieve high-required energy and spectrum efficiency (SE) for OFDM [104]. Studies have shown that orthogonal frequency division multiplexing with index modulation (OFDM-IM) [50] is a promising multicarrier system that can replace conventional OFDM. In OFDM-IM, a subset of the sub-carriers is active, and the indices for them are used to convey information bits. This allows the OFDM-IM to be more reliable and energy efficient than the conventional OFDM due to the fact that there is no need for bandwidth or extra power when using indices of the active subcarriers to carry the data bits.

In view of the above characteristics and benefits, many studies in recent years have proposed OFDM-IM. For example, analysing the frameworks of the symbol probability in [105] and the BER of OFDM-IM with uncertain channel state information (CSI) in [106]. Researchers have also studied the potential to improve the performance of OFDM-IM [107–109]. Further, [110], combined OFDM-IM with multiple-input multiple-output (MIMO) systems to improve SE. Based on an energy detector, [111] proposed a low complexity greedy detector (GD) and [112] analysed its BER.

Compared to conventional OFDM systems, the OFDM-IM performs better but still suffers from high detection complexity. For example, the maximum-likelihood detector (MLD) achieved optimal BER performance with exponentially high complexity. With precise knowledge of the receiver noise power spectral density, a low-complexity log-likelihood-ratio (LLR) detector to achieve near-ML performance was proposed in [50]. Like an OFDM detector, GD has low complexity; however, it suffers from high-performance loss compared to the ML detector.

MIMO systems play an important role in most modern communication channels. A MIMO system can provide substantial performance gain since the dimensions can account for different resources, such as frequency resources, multiple users, multiple antennas, and time. However, MIMOs present computational complexities resulting in detection problems. We note that IM-

aided MIMO-OFDM (IM-MIMO-OFDM) cannot be used directly with a classical detector in MIMO-OFDM systems. This is due to the increase of in-detector complexity since part of the information bits are conveyed on the indices of the active transmit antennas (TAs) and subcarriers that need to be detected in addition to the modulated symbols [50].

A maximum likelihood-based detector has been analyzed in [113] and [59]; however, with the rises in TAs, sub-carrier numbers, and the constellation signal size, detector complexity has increased rapidly. In order to reduce the complexity of detection, a number of low-complexity detectors have been proposed, such as a log-likelihood ratio (LLR) detector, a matched filtering (MF) detector, a simple minimum mean square error (MMSE) detector, and a signal vector-based list (SVBL) detector. However, all of the proposed low-complexity detectors suffer from a notable degradation in error performance.

In this chapter, we introduce a DL-based detector for IM-MIMO-OFDM using a DNN in terms of error performance. In the DLIM design, based on domain knowledge, the channel information and the received signal are preprocessed before being fed to the DNN. Using the simulated data, the DLIM is trained offline in order to minimize the BER and then, with low runtime, the trained model can be used as an online detector. Our initial results using this detector design have significantly reduced the complexity while maintaining a near-optimal performance compared to the existing hand-designed detectors.

4.3. Deep Learning Based Detection

4.3.1. Deep Learning Methods

DL and artificial neural networks (ANNs) have been successfully applied to many applications, including solving a complex issue in computer vision (CV), automatic speech recognition (ASR), and natural language processing (NLP), which has inspired communication experts to apply DL to their fields.

Recently, DL methods have been responsible for improving the performance of several communication applications. For example, localisation based on CSI [106], channel equalisation [111], replacing both the transmitter and the receiver of the communication system [3], reducing the peak-to-average-ratio of an OFDM system [114], and for designing a joint-channel estimation and signal-detection

receiver for an OFDM system [115]. Using a DL approach to improve detection for OFDM-IM was proposed in [116]. In [117], a learning-based detector for an IM-aided MIMO-OFDM system was proposed. Previous work investigating the use of DL in OFDM-IM detection and complexity reduction in [116] has resulted in a near-optimal performance. Our goal was to apply DL to an IM-MIMO-OFDM system. In this chapter, we propose a DL-based detector for IM-MIMO-OFDM that can detect the transmitted symbols with performance comparable to near-optimal BER with shorter runtime than the existing hand-crafted detectors.

4.3.2. System Architecture

Similar to the current IM-MIMO-OFDM detector schemes, the system is equipped with N_t transmit antennas and N_r receive antennas, as shown in Fig. 4.1. The block diagram of IM-MIMO-OFDM at each transmit antenna is given in Fig. 4.2.

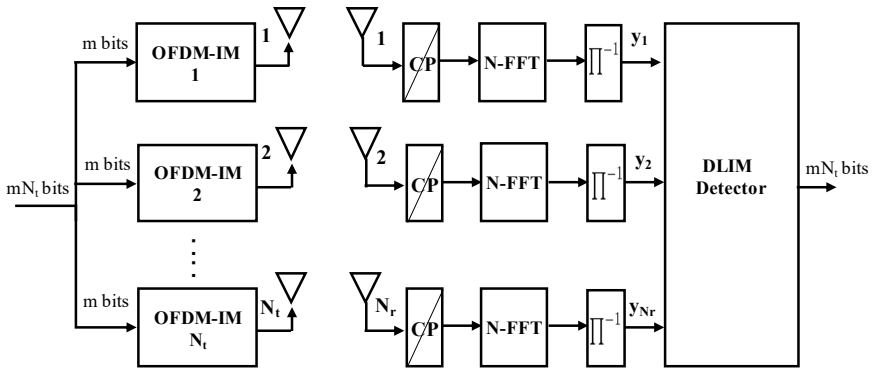


Figure 4.1: Transceiver structure of the IM-MIMO-OFDM scheme.

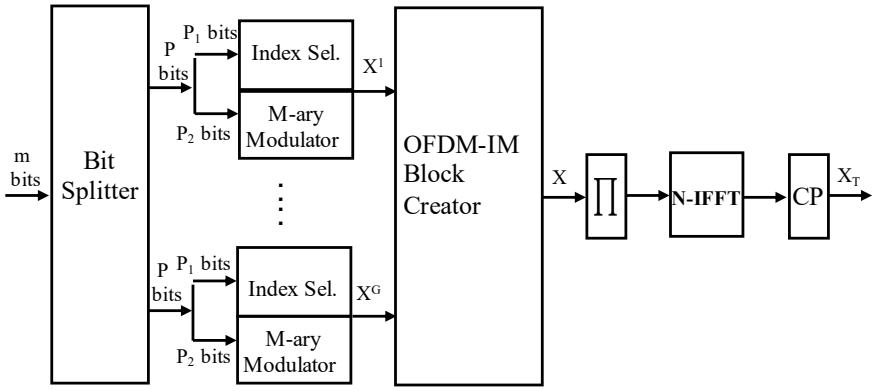


Figure 4.2: Block diagram of the OFDM-IM at each branch of the transmitter.

The proposed structure of the IM-MIMO-OFDM detector using DNNs, termed DLIM, is shown in Fig. 4.3. We consider the IM-MIMO-OFDM system as a system with N_c subcarriers split into a G group of N subcarriers. The signal of each group is processed both the same and independently of each other on the transmitter side. To send a total of p data bits, only k out of N subcarriers are activated and carried by the K complex data symbol $p1 = Klog_2M$ bits (M is the M-ary modulation size) and $p2 = [log_2C(N, K)]$ bits carried by indices of active subcarriers, resulting in $p = p1 + p2$.

We use the look-up table or a combination of methods to implement the mapping from p1 bits to a combination of k active indices [50,116]. The transmitter vector $x = [x_1, ..., x_N]$ contains the simultaneously transmitted symbols from all transmit antennas and can have zero terms due to index selection in each branch of the transmitter. In other words, the transmitter vector x will be formed depending on the incoming p bits and by assigning K non-zero data symbols to corresponding K active subcarriers i.e., x_i is non-zero if subcarrier i is active and $x_i = 0$, otherwise, $i = 1, ..., N$. $x = f_{OFDM-IM}(b)$, is a function of the bit-to-symbol mapping, where b is a sequence of p incoming bits in one group.

On the receiver side, the receiver signal for N consecutive subcarriers of a given sub-block g can be expressed in a frequency domain by

$$y^g = H^g x^g + w^g, \tag{4.1}$$

for $g = 1, 2, ..., G$, where $y^g \in C^{RN\times 1}$ is the vector of the stacked received signal for the corresponding subblock, while the block-diagonal channel (h) matrix, with dimensions $R \times T$, that contains the channel coefficients between the transmit and receive antennas is assumed to be perfectly known and the receiver is $H^g \in C^{RN\times TN}$, $x^g \in C^{TN\times 1}$ is the equivalent data vector, and the noise vector is $w^g \in C^{RN\times 1}$. The average received signal-to-noise ratio (SNR) is $\overline{\gamma} = E_s/\sigma^2$ and E_s is the average energy of a transmitted M-ary symbol.

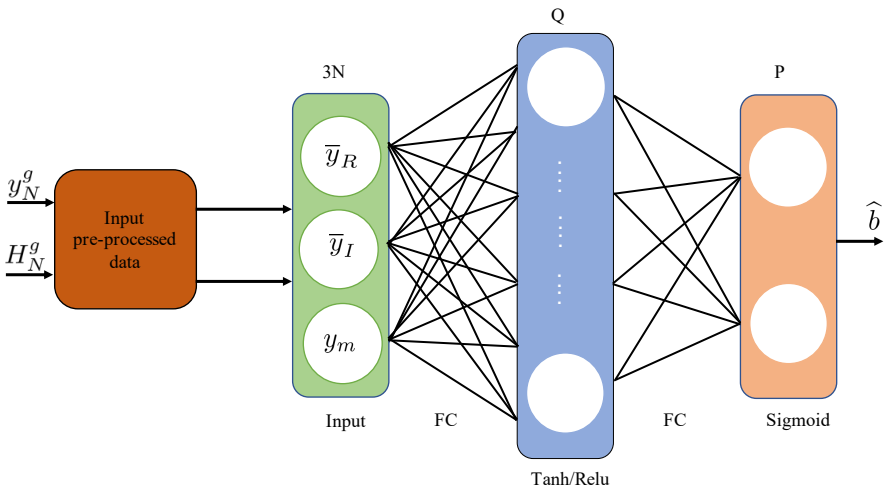


Figure 4.3: The structure of the proposed DLIM detector for IM-MIMO-OFDM.

4.3.3. Model Training

In the current IM-MIMO-OFDM detection scheme, the channel information is assumed to be known at the receiver. However, the received signal y^g and the channel H^g will be pre-processed based on the domain knowledge of IM-MIMO-OFDM and considered as the inputs for the DNN. The index selection of the active subcarrier is performed using look-up tables of the OFDM index modulators of the transmitter. At each branch of the transmitter, the OFDM index modulators will obtain the subblocks of the OFDM-IM first before concatenating these subblocks to form the main OFDM blocks. In particular, to improve the reconstruction of the M-ary symbol of the active subcarriers and to achieve an equalised received signal as $\overline{y^g} = y^g \odot h^{-1}$, the zero-forcing (ZF) equaliser is applied. Then, $y_m = [|y_1|^2, ..., |y_N^2|]$ is computed, which is the energy of the received signal before it is combined with y to create the input of the DNN. Finally, we concatenated the real $\overline{y}_R$ and the imaginary parts $\overline{y}_I$ of $\overline{y}$ with the received energy vector y_m to form a 3N-dimensional input vector $z = [\overline{y}_R, \overline{y}_I, y_m]$. As shown in Fig. 1, the DNN consists of one hidden layer of Q nodes with two fully connected (FC) layers and one output FC layer of p nodes. For the hidden layer, we used two different activation functions. The rectifier linear unit (Relu), $f_{Relu(x)} = max(0, x)$, or the hyperbolic tangent (Tanh) function, $f_{Tanh(x)} = \frac{1-e^{-2x}}{1+e^{-2x}}$. To get the estimated output of the transmitted data $\widehat{b}$ at the output layer, we applied the sigmoid function $f_{Sig(x)} = \frac{1}{1+e^{-x}}$. Since the output of the sigmoid function is between 0

and 1, b will be 1 bit when its value is more than 0.5 or 0 bits otherwise. Moreover, it is worth noting that in order to accurately detect the M-ary and the index bits of the DLIM, only two nonlinear layers are required. We can express the output of the DLIM model as

$$\widehat{b} = f_{Sig} W_2 f_{Tanh/Relu}(W_1 z + b_1) + b_2, \tag{4.2}$$

where W_1, b_1 and W_2, b_2 are the weights and the biases of the first and second FC layers, respectively. In particular, the system parameters (i.e., N, K, and M) determined the length of the input and output of the DLIM. However, in order to achieve the desired performance, the value of Q needs to be properly selected for every system configuration. In other words, the value of Q needs to be large enough that the number of transmitted bits p increases in order to guarantee the pre-determined performance. Another advantage of adjusting Q is that we can reach a satisfactory trade-off between performance and complexity.

4.4. Experiment Setup and Training Procedure

In this section, we present the training procedure before using the proposed system and then compare it with the existing schemes, such as ML and GD detectors, for different system configurations.

4.4.1. Offline Training

To collect the data from the simulation, we started training the model offline by randomly generating a sequence of p bits b in order to achieve a corresponding set of transmitted vectors, such that $x = f_{OFDMIM}(b)$. Next, we sent all the vectors that were subjected to the effects of the Rayleigh fading channel and AWGN noise to the receiver. Based on known statistical models, the vectors of the channel and the noise were randomly generated and changed from a one-bit sequence to another. The collected received signal y and channel vectors h were pre-processed in order to obtain the input dataset z whose label corresponded to bit sequence b.

Our model was trained using the collected data to minimize the BER or trained to minimize the difference between b and its prediction $\widehat{b}$. Thus, we use the mean-

squared error (MSE) loss function for the training

$$\mathcal{L}(b,\widehat{b}) = \frac{1}{p}\left\|b-\widehat{b}\right\|^2, \tag{4.3}$$

where $\theta = (W_i, b_i)_{i=1,2}$ are the weights and the biases of the model. Using the SGD algorithm, the parameter can be updated for every randomly picked up batch from the datasets using the learning rate η, which defines the step size of the SGD as follows:

$$\theta^{+} := \theta - \eta \bigtriangledown \mathcal{L}(b,\widehat{b};\theta). \tag{4.4}$$

To train our model based on the SGD, we used Adam optimizer and implemented it using Tensorflow and Keras for DL. In order to efficiently train our model, it is essential to choose the best level of SNR $\overline{\gamma}_{train}$ since the performance of the model is very sensitive to it. Thus, the model trained by $\overline{\gamma}_{train}$ will still perform well under any other SNRs of interest. Moreover, in order to prevent overfitting in training, the number of the training datasets (z,b) must be large enough.

4.4.2. Online Training

After offline training, the model is utilized for the online deployment of signal detection in the DLIM. Now, we can implement it in real-time in order to estimate the data bits over the different channel fading conditions with no extra training for θ. In other words, every time the received signal and the channel information are fed to the DLIM, it will output the estimated bits in a very short computation time.

4.5. Simulation Results

In this section, we present simulation results for different configurations of the DLIM in order to analyse the performance of our proposed scheme against the ML detection and GD detector. All simulations were run over a Rayleigh fading channel with AWGN noise. Our proposal was trained with 300 epochs with 20-batch sizes; the learning rate η was set to 0.01. The rest of the parameters varied for each experiment.

4.5.1. BER Performance

Fig. 4.4 compares the BER results for all antennas of our DLIM proposal with ML detector and GD detectors when $(N,\ K,\ M) = (4,\ 2,\ 2)$, $\overline{\gamma}_{train}$ is set to 15 dB, and perfect CSI is assumed. For the proposed DLIM, different values of Q and corresponding activation functions are considered at the hidden layer. For example, for a large value of Q, i.e., $Q = 128$, the activation function Relu is used, while for smaller values of Q, i.e., $Q = 64,\ 32,\ \ldots$, the activation function Tanh is used since it is not limited to being a non-negative value as is the case with Relu. This provides a higher model capacity to perform the detection task. As shown in Fig. 4.4, DLIM with $Q = 128,\ 64$ achieves a BER very close to ML detector: there is just a slight performance gap between the proposed detector and ML detector. However, the GD performance is worse than the proposed detector. While with small values of Q, i.e., $Q = 32$ and 16, the performance is very close compared to GD. Moreover, for the same $Q = 16$, the Tanh is much better than Relu.

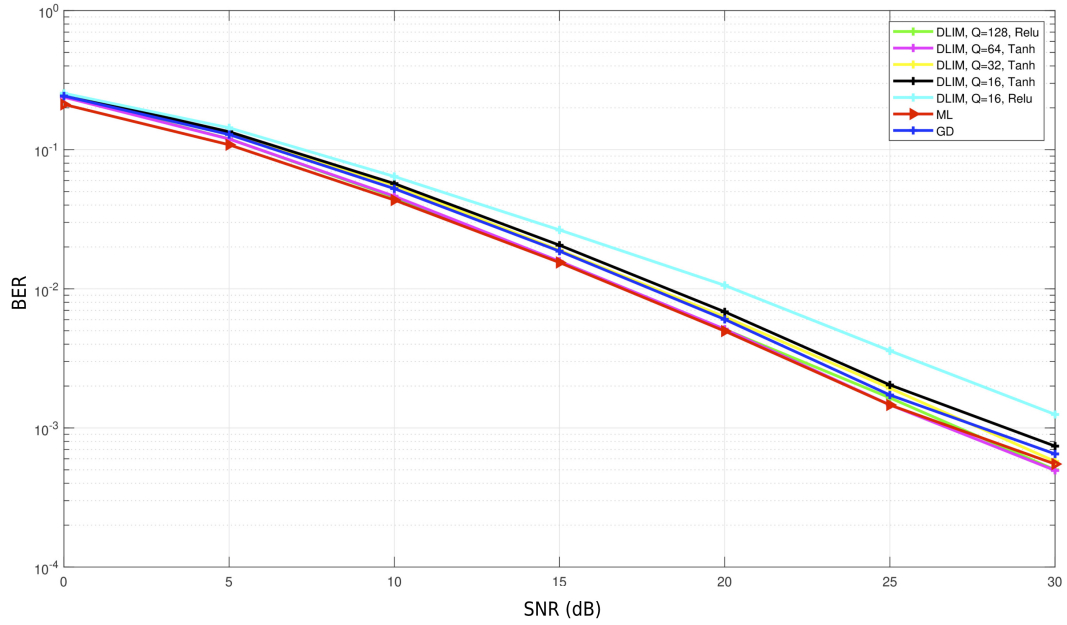


Figure 4.4: BER comparison between the proposed DLIM and the reference detectors and $(N,\ K,\ M) = (4,\ 2,\ 2)$ and 15 dB SNR .

Fig. 4.5 illustrates the BER comparison for higher $\overline{\gamma}_{train}$, i.e., 5 dB when $(N,\ K,\ M) = (4,\ 2,\ 4)$. DLIM with $Q = 128$ achieves a BER very close to ML detector. There is just a slight performance gap between the proposed detector

and ML detector. However, the GD performance is much worse than the proposed detector. With a small Q, i.e., $Q = 32$ and 16 the performance is still better than GD and very close compared to ML detector. While for the same $Q = 32$, the Tanh is much better than Relu. From these two figures, we notice that there is a trade-off between the performance and the complexity every time we adjust Q.

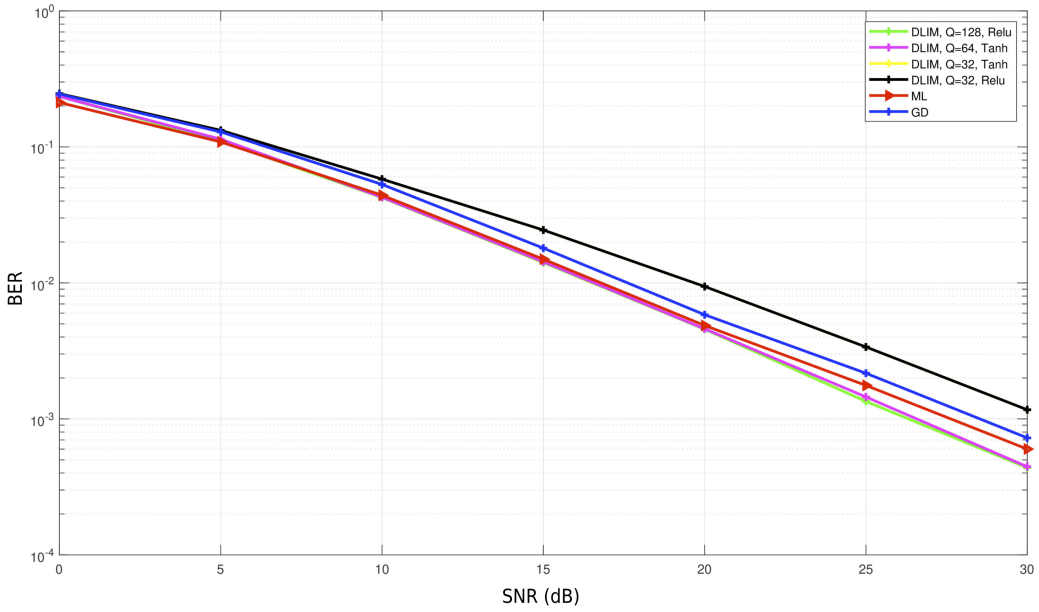


Figure 4.5: BER comparison between the proposed DLIM and the reference detectors and $(N, K, M) = (4, 3, 4)$ and 5 dB SNR .

Fig. 4.6 presents the BER comparison when $(N, K, M) = (4, 3, 4)$ and the $\overline{\gamma}_{train}$ is set to 5 dB. We demonstrate this by adapting the MMSE-based variable-imperfect CSI, where the CSI error variance ε^2 is a function of the average SNR as in [105]. From Fig. 4.6, we can see that DLIM with $Q = 128$ achieves a BER very close to ML detector and significantly outperforms GD. From Fig. 4.5 and Fig. 4.6, it is clear that when the numbers of activated antennas and subcarriers increase, the BER performance has a specific gain when using the proposed DLIM detector.

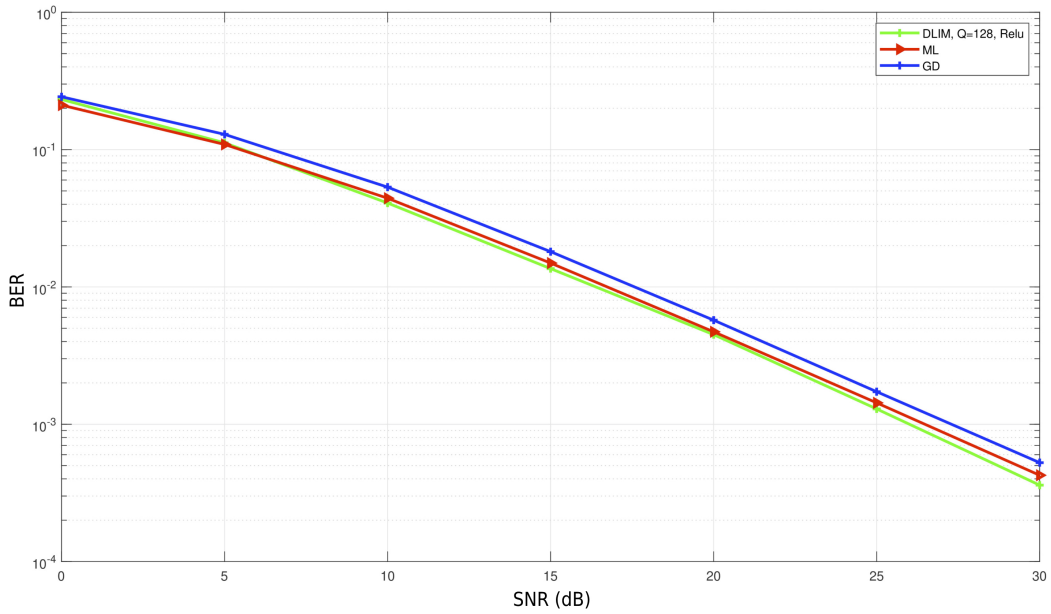


Figure 4.6: BER comparison between the proposed DLIM and the reference detectors and $(N, K, M) = (4, 3, 4)$ and 5 dB SNR.

4.5.2. Complexity Comparison

To showcase the low complexity of the proposed DLIM against the other two traditional detectors (ML detector and GD), a $Q = 128$ with Relu and $Q = 64$ with Tanh were employed for all the cases. As shown in Table I, the runtimes are almost identical in the DLIM with the Relu and Tanh, and far less than ML and GD. From Table I, we can see that when M increases, the runtime of DeepIM with the Relu and Tanh shortens compared to ML and GD. Moreover, as K gets larger, the runtime of DeepIM with the Relu and Tanh is greatly shortened compared to ML and GD. From Tabel 4.1, we can observe that unlike the current detectors, the proposed DLIM complexity is less and depends on the K and M values. Moreover, when we increase the number of subcarriers and activate more antennas to convey information, naturally, the computational complexity also increases.

Table 4.1: A COMPARISON OF COMPLEXITY AMONG DLIM, ML, AND GD DETECTORS

(N, K, M)	ML	GD	DLIM/Relu	DLIM/Tanh
(4, 2, 2)	0.2	0.099	0.038	0.04
(4, 2, 4)	0.38	0.08	0.04	0.042
(4, 3, 4)	1.9	0.09	0.044	0.047

4.6. Chapter Summary

In this chapter, we illustrated the effectiveness of employing a DNN for the detection of IM-MIMO-OFDM in terms of error performance. The proposed model, DLIM, trained the received signal and channel vectors offline based on domain knowledge, such as a ZF equalizer and energy detection. Then, the fully connected DNN was trained to recover the data bits. The results of this simulation show that a DL method has advantages. Once the model is trained, we can employ the DLIM in an online manner with very low runtime while achieving near-optimal performance. The next chapter introduces a classical IRS-assisted communication system transformed into an end-to-end AE based on DL.

Chapter 5

Design of Autoencoder Model for End-to-End Deep Learning IRS-assisted Communications Systems

This chapter is organised as follows: section 5.1 introduces the chapter. Section 5.2 states the related works and states the main contributions. The end-to-end IRS-assisted design is illustrated in section 5.3. Section 5.4 introduces the experiment setup and AEs training. The numerical simulation comparing the results with other works is presented in section 5.5. Finally, section 5.6 summarises the chapter. The work in this chapter has been published in P.2.

5.1. Introduction

In this chapter, we are re-modelling the IRS-assisted communication systems using the AE DL technique to represent the classical IRS system as an end-to-end communication system. The cascaded channels from source to sink through the IRS have been transformed into a DNN that learns how to reduce the wireless environment impairments effect by optimizing the representation of transmitted symbols. The proposed system design shows superior symbol error rate (SER) performance under the AWGN channel compared to both classical IRS and conventional AE end-to-end systems. The relation between improvement of performance and the capability of the proposed AE to learn optimized presentation

for transmitted symbols is explained by observing and comparing the baseline AE constellations learning with the ones that the proposed model learned.

5.2. Related Works

Recent research has identified IRS as one of the key technologies for wireless networks beyond 5G. Among the future wireless systems, the IRS has emerged as a promising solution for improving coverage and rate. IRS can be implemented using nearly passive elements with parameters that can be reconfigured [2, 118–121]. Moreover, the IRS has a unique wireless communication capability, where an integrated printed electronic board is placed on a surface coated with electromagnetic (EM) material, which controls the reflected electromagnetic signal by changing the phase of the incident signal to minimize the effects of stochastic wireless environments at the receiver. Smart radio environments are well suited to IRS-assisted communications due to the simplicity of their implementation and their nearly-passive elements [122, 123]. In this case, however, the IRS is unable to perform any radio frequency (RF) processing, which limits the estimation of both the CSI and the signal phase angles at the receiver end. For this reason, the receiver must estimate the channel and received phase angle without heavily relying on the IRS. [10, 124]. Another disadvantage of the IRS is the large number of elements it requires [124], [9]. There have been several different designs introduced in [125,126] for conventional reflect-arrays and software-defined meta-materials in [64], [127]. There has been the development of DL tools in [10,128–131]; the objective of these studies is to learn directly from the sampled channel knowledge how to predict the optimal IRS reflection matrices. The IRS array geometry is not assumed in this approach, nor does it require knowledge of the sparse channels.

As DL has proven tremendously successful in various applications such as modulation, it has increased researchers' interest in the area of communications in recent years [3], signal detection [132], channel estimation [3] and channel coding [133]. Based on the end-to-end DL model in [3], the authors demonstrate significant gains upon jointly designing the modulation and coding together. By utilising two-way amplify-and-forward (TWAF) relay networks under block fading Rayleigh channels, the authors in [134] proposed and tested an end-to-end DL model for relay networks. In comparison to the conventional method, the

authors demonstrated a promising performance gain with the use of learning-based networks.

Inspired by AE network in [3], the authors in [135] use the AE to relay amplify-and-forward (AF) data in two ways: differential modulation coding schemes for one-way amplify-and-forward (OWAF) and TWAF relay networks, where BER performance is significantly improved in comparison to the classical method by combining coding and modulation in a DL-based framework under block fading channels.

The AE is typically viewed as a generative model comprised of an encoder, bottleneck, and decoder. As a result of the encoder, the input signal is reduced in dimension and the decoder takes this reduced information and decodes it back into an original signal. The design of the end-to-end communication system for the IRS is based on state-of-the-art research on AE and IRS, where the whole system is considered as one entity in DL. This allows optimizing the whole communication system as a single DNN, as opposed to optimizing each component separately as is the case with conventional systems. A hot vector is assigned to each input symbol, and this vector is used as the input to AE. AE encoder output is altered by an IRS layer, which then uses its output as an input to a noise layer. A final step in the decoding process occurs when the noise layer output is used to reconstruct the transmitted symbol.

5.3. Main Contributions

In this chapter, a new approach has been proposed and investigated with the help of AE. Our main contributions can be summarized as follows:

- As part of an end-to-end communication system that utilizes AWGN channels, we propose a DNN-based architecture based on AE to perform modulation and demodulation. This novel, fully data-driven system, relies on IRS assistance to learn effectively to encode and decode the transmitted $\mathbf{s}$ and the received $\widehat{\mathbf{s}}$ symbols with IRS help while minimizing the MSE between $\mathbf{s}$ and $\widehat{\mathbf{s}}$.

- We have demonstrated through numerical results that our AE-based scheme achieves higher performance gains than the existing classical IRS scheme [4] and the AE hand-crafted baselines [3]. Additionally, we explain how

AE learns constellations to improve system performance in support of these findings.

- The proposed AE design results demonstrate the possibility of achieving a higher SER performance even with a limited number of elements. As a result of this, the weaknesses of IRS-assisted communication systems performance that have a small number of IRS elements are overcome, which is mentioned in [9, 10].

5.4. End-to-End IRS assist AE design

AEs were originally designed in order to reduce the dimensionality of their inputs, which allowed them to reconstruct their output unsupervised with a minimum level of error. As a result, we are primarily concerned with learning how to represent the input $\mathbf{x}$ of the messages robustly so that they can be decoded with a small probability of error. This network is composed of a number of layers constituting a representation of a DNN, it may consist of a number of hidden layers that form a DNN. The hidden layers, $\mathbf{b}$, describe a bottleneck used to represent the input data $\mathbf{x}$. We propose to create an end-to-end DL model for a none-line of sight (N-LoS) communication system that includes a transmitter, receiver, and N meta-surfaces IRS, as shown in Fig.5.1, introduce two challenges. Firstly, how to model the propagation path of each signal through two channels, instead of one, for signals propagating between the transmitter and receiver. Secondly, how to represent the substantial improvement in SNR at the receiver caused by constructively reflected signals superposition with optimized phase shifts, this can be formulated under the assumption of fully optimized IRS to compensate the reflection coefficients as in [4]:

$$S_r = \mathbf{g}^{\mathrm{T}} \Phi \mathbf{h} S_t + n_o, \tag{5.1}$$

where $\mathbf{h} = \left[\alpha_1 e^{-j\theta_1}, \ \ldots \ , \alpha_i e^{-j\theta_i}, \ \ldots \ , \alpha_{_N} e^{-j\theta_N}\right]^{\mathrm{T}}$ is the channels coefficients from the transmitter to each IRS meta-surface for the transmitted signal S_t. $\mathbf{g} = \left[\beta_1 e^{-j\psi_1}, \ \ldots \ , \beta_i e^{-j\psi_i}, \ \ldots \ , \beta_{_N} e^{-j\psi_N}\right]^{\mathrm{T}}$ is the channels coefficients from each IRS meta-surface to the receiver, both $\alpha_i e^{-j\theta_i}$ and $\beta_1 e^{-j\psi_i}$ are sampled from $\mathcal{CN}(0,1)$, Φ is the diagonal matrix of the applied phase shifts by each meta-surface $\Phi = \mathrm{diag}(\left[e^{j\phi_1} \ e^{j\phi_2} \ \ldots \ e^{j\phi_N}\right])$ and the n_o is the AWGN noise $\sim \mathcal{CN}(0, \sigma^2)$, where

the σ^2 is the variance value contaminates the amplitude and phase of transmitted signal, this can be depicted for one signal route between the transmitter and receiver in Fig.5.2. However, by using an IRS with N number of meta-surface elements, Fig.5.3 portray Eq.(5.1) communication system block diagram Fig.5.1. The main contribution in this work is to transform this conventional system in Fig.5.3 to an end-to-end communication system using the AE as in Fig.5.4, which consists of :

5.4.1. Encoder

Suppose that the space of possible messages transmitted is $M = 2^k$ and that the number of bits necessary to represent each message m is k. Following this, transmitting input symbol $s_t \in \{1, ..., m, ..., M\}$ with space size M denoting the number of possible transmitted. After encoding the input symbol as a one-hot vector, $1_s \in \mathbb{R}^{2^k}$ as a 2^k-dimensional vector and recognizing a transformation of $f : \mathbb{R}^{2^k} \rightarrow \mathbb{R}^c$, the encoder feeds it into the input layer. In the encoder, c is the dimension of the last layer, which serves as the input for the next step, where the IRS, as well as the wireless channels, have been modelled as augmenting layers, which include the effect of AWGN noise on the encoder output code. This encoder can be formulated by:

$$\mathbf{y}_i = f(\mathbf{x}, \theta_T), \tag{5.2}$$

to describe the hyperparameters θ_T for the transmitter's DNN layers.

5.4.2. Signal Route

IRS layer

It has been necessary to add a new layer to increase the dimensionality of this study so that the particular features of physical single signal propagation routes, including IRS single element phase shift effects, can be analyzed: $\mathbb{R}^c \rightarrow \mathbb{R}^{Nc}$, where the N is the number of the meta-surfaces the IRS consists of. After, at the normalization layer, physical constraints on $\mathbf{x}$ average power $\mathbb{E}\left[|x_m|^2\right] \leq 1 \, \forall m$ applied. Lastly we reduce the dimensionality back to $\mathbb{R}^c$ using one more fully connected (FC) layer $z(f(\mathrm{x})) : \mathbb{R}^c \rightarrow \mathbb{R}^c$. This IRS layer can be formulated by:

$$\mathbf{y}_i = z(f(\mathbf{x}), \theta_{SR}), \tag{5.3}$$

to describe the hyper parameters θ_{SR} for the signal route's DNN layer.

Channel

In the proposed communication system, the channel layer is one of several layers that are integrally part of the signal pathway following the IRS layer, and it is described by $p(\mathbf{w}|\mathbf{y})$, where $\mathbf{w} \in \mathbb{R}^c$ and c are the number of channels that the system utilizes to send one message out of 2^k messages. There is a way to quantify the communications rate by using $r = k/c$ [bit/channel use], where $k = \log_2(M)$ and it is the number of bits that represent the message. The channel noise is an AWGN due to the assumption that the main source of the noise is on the receiver side [3]. The channel is characterized as a distribution $\mathcal{N}(0, \xi^2\mathbf{I})$ with a fixed variance $\xi = (2rE_b/N_o)^{-1}$, where (E_b/N_o) is the energy per bit E_b to the noise spectral density N_o ratio. Based on the conditional probability function, we can describe the input-output relationship of the signal route component including the IRS layer as follows:

$$p(\mathbf{w}|\mathbf{y}) = \frac{p(\mathbf{y}, \mathbf{w})}{p(\mathbf{y})}, \tag{5.4}$$

where to sample $w \sim p(\mathbf{w})$ requires to assume the knowledge of $p(\mathbf{w}, \mathbf{y})$ to find $\int p(\mathbf{w}, \mathbf{y})dy = \int p(\mathbf{w}|\mathbf{y})p(\mathbf{y})dy$, which is in general is difficult to find. However, at training time both $p(\mathbf{w}|\mathbf{y})$ and $p(\mathbf{y})dy$ are available respectively to give $p(\mathbf{w}) = \int p(\mathbf{w}|\mathbf{y})p(\mathbf{y})dy$.

5.4.3. Decoder

In the communication system, the decoder is located on the receiver side, which is composed of two layers of dense M units. Based on the output signal of the noise layer $\mathbf{w} \in \mathbb{R}^c$, the decoder transforms $k(\mathbf{w}) : \mathbb{R}^c \to \mathbb{R}^{2^k}$ in order to reconstruct the message probability output vector $\widehat{\mathbf{x}}$. The received symbol s_r decoded using $\widehat{\mathbf{x}}$, $s_r \in \mathbb{R}^{2^k}$. During decoding, one dense layer is activated using ReLU, while the other density layer is activated using softmax. The results of the decoding process are a probability distribution with M probabilities, and the highest index probability is converted into a decoded message s_r. This decoder can be formulated by:

$$\mathbf{y}_i = k(\mathbf{w}, \theta_R), \tag{5.5}$$

to describe the hyperparameters θ_R for the receiver's DNN layers.

Training an end-to-end AE can be done by minimizing the reconstruction loss function.

AEs are penalized based on loss function $\mathcal{L}$, such as categorical cross-entropy or MSE, which describes the difference between the desired input $\mathbf{x}_m$, where $m \in \{1, .., M\}$ and the observed output $\widehat{\mathbf{x}}_m^i, i \in \{1, ..., \mathcal{D}\}$, where $\mathcal{D}$ is the size of the training set. In our work we are using below categorical cross-entropy equation:

$$\mathcal{L}(\mathbf{x}, \widehat{\mathbf{x}}) = -\sum_{i=1}^{\mathcal{D}} \sum_{m=1}^{M} x_m \ln(\widehat{x}_m^{(i)}). \tag{5.6}$$

The stochastic gradient descent (SGD) method was used to train the AE using the Adam optimizer. This process leads to optimize all AE hyperparameters $(\theta_T, \theta_{SR}, \theta_R)$ as one entity [100].

Assuming that each IRS element receives and reflects a signal independently of the others, and the signals are constructively received, then the equation describing the expected message probability output vector is as follows:

$$\mathbb{E}[\widehat{\mathbf{x}}] = \frac{1}{N} \sum_{n=1}^{N} \widehat{\mathbf{x}}_n, \tag{5.7}$$

where N is the number of IRS meta-surfaces. Then the $\mathbb{E}[\widehat{\mathbf{x}}]$ will be used to have the final decoded symbol s_r.

While the training of the model has been summarized in algorithm 1, the deployed model has been outlined in algorithm 2. Moreover, Table 5.1 lists the layout and general configuration of each proposed IRS-assisted AE component we adopt in this proposal.

Table 5.1: Layout of proposed IRS-assisted end-to-end AE as in Fig. 5.4

Component	**Layer**	**Activation Function**	**Output Dimension**
Transmitter	Input		M
	Dense (FC)	ReLU	M
	Dense (FC)	Linear	c
Signal route	Dense (FC)	Linear	N $\times c$
	Normalization		N $\times c$
	Dense (FC)	Linear	c
	Noise		c
Receiver	Dense (FC)	ReLU	M
	Dense (FC)	softmax	M

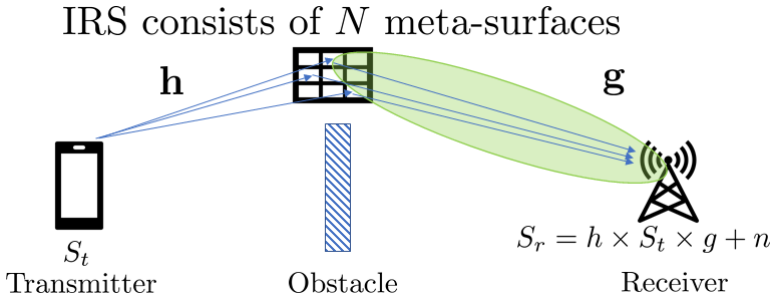


Figure 5.1: Physical N-LoS IRS communication system

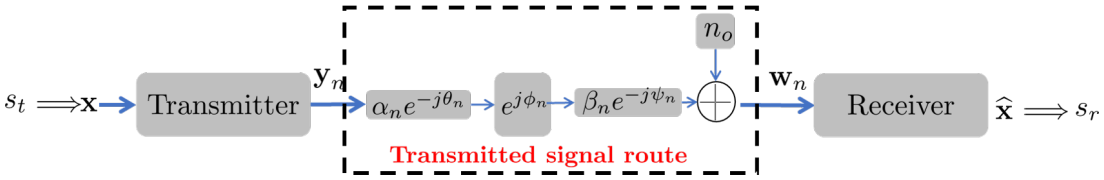


Figure 5.2: Block diagram for conventional single signal route including phase shift effect caused by one IRS element

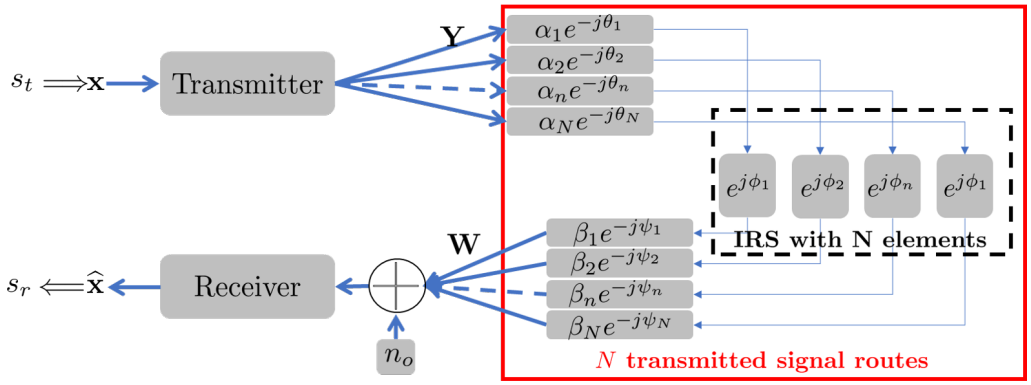


Figure 5.3: Block diagram for conventional IRS assisted -communication system

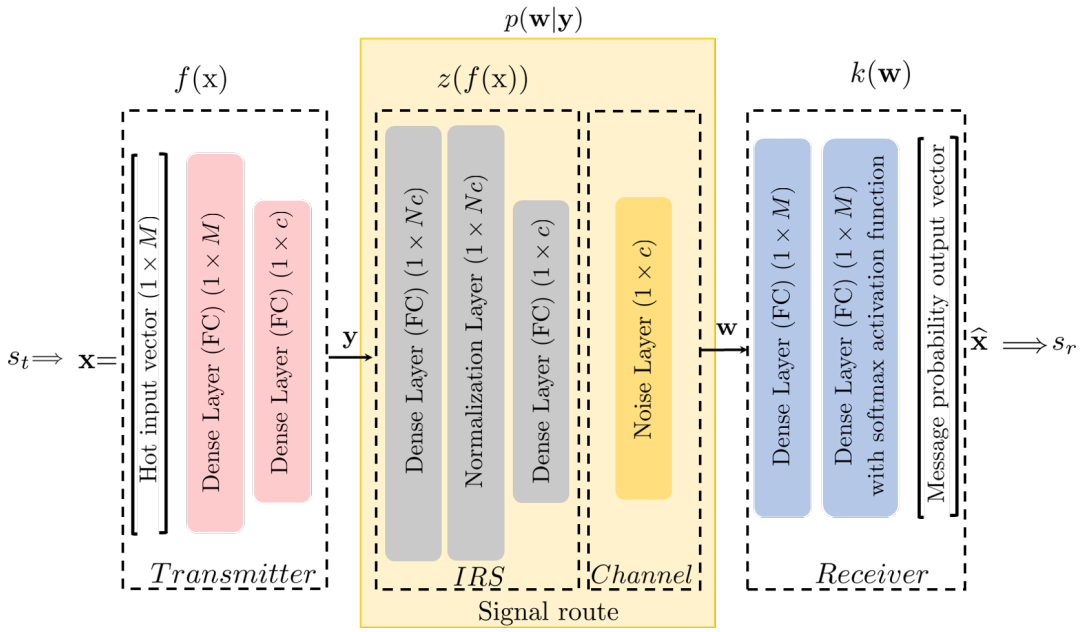


Figure 5.4: Proposed IRS-assisted end-to-end communication system using AE architecture

5.5. Experiment Setup and AEs training

Monte-Carlo simulations are used during the IRS part of the experiment to run 2×10^6 BPSK symbols. Following [4], gray encoded symbols and the (E_b/N_o) is the same as (E_s/N_o) for BPSK modulation scheme. Considering the symbol period time of the signal is the same as the sampling period of the signal, then the $SNR = (E_b/N_o)$. The $\mathbf{h_i}$ and $\mathbf{g_i}$ are sampled from $\mathcal{CN}(0, 1)$, $\Phi_i = \theta_i + \psi_i$ according to Eq. (5.1). A Maximum Likelihood Detector (MLD) decodes the received signal at the receiver, and the SER performance has been produced almost the same as in [4] at different (E_b/N_o). Furthermore, we used these results as a baseline for comparing with the results of our assisted end-to-end communication system. For our experiment, we have also used the AE designed by [3] to obtain results for $r = k/c$, where $k = 1$ and $c = 7$, to determine the BPSK SER performance for different (E_b/N_o) values. It is worth mentioning that the representation of the BPSK constellation has been changed from one dimension to 7 dimensions at the bottleneck of the AE. We used these results as a second baseline for evaluating the performance of our proposed assisted end-to-end communication system. We have configured and trained our proposed model setup to evaluate the SER performance

of BPSK over AWGN channels. We used python for our simulations, and the system parameters used during these simulations are listed in Tables 5.2 and 5.3.

Table 5.2: Layout of proposed IRS-assisted end-to-end AE as in Fig.5.4

Layer + Activation Function	**Output Dimension**
Input	$M = 2$
Dense (FC) + ReLU	$M = 2$
Dense (FC) + Linear	$c = 7$
Dense (FC) + Linear	$7 \times N$, $N \in \{4, 16, 64\}$
Batch Normalization	$7 \times N$, $N \in \{4, 16, 64\}$
Dense (FC) + Linear	$c = 7$
Noise	$c = 7$
Dense (FC) + ReLU	$M = 2$
Dense (FC) + softmax	$M = 2$

Table 5.3: Parameters used for simulations

Parameter	**Value**
Modulation scheme	BPSK
AWGN channel noise level type	E_b/N_o [dB]
IRS size	$N \in \{4, 16, 64\}$ [meta-surface]
No. of BPSK transmitted symbols	2×10^6 symbols
Used E_b/N_o in training AEs	7 dB
AEs main training parameters	Epo (epochs)=17
	Batch size=300
	Validation data size= 1500
Adam optimizer parameters	η=0.001
	(λ_1, λ_2)=(0.9, 0.999)
	$\varepsilon = 1^{-7}$

5.6. Experiment Results

In Fig.5.5 the results indicate that our proposed AE communication system without IRS elements outperforms the baseline AE communication system in [3]. This can be verified by looking at Fig.5.7 (a), where the learned constellations by the proposed system have larger euclidean distances in comparison to the

Algorithm 1 IRS-assisted AE training

Initialization: {Epo,Itr, $\mathbf{x}$, n_o, η, θ}, where
Epo: number of epochs training.
Itr: number of iterations per epoch.
$\mathbf{x}$: training input vector. Also the desired output.
n_o: noise sample $\sim \mathcal{N}(0, \xi^2)$
θ: DNN weights and biases matrix for θ_T, θ_{SR}, θ_R
for each Epo **do**
  for each Itr **do**
    use $\mathbf{x}$ for input layer to produce $\mathbf{y} = f(\mathbf{x}, \theta_T)$
    use $\mathbf{y}$ for IRS layer to produce $\mathbf{z} = z(\mathbf{y}, \theta_{SR})$
    use $\mathbf{z}$ for noise layer input to produce vector augmented random values $\mathbf{w} \sim \mathcal{N}(0, \xi^2)$
    use $\mathbf{w}$ as decoder input to calculate the message probability vector $\widehat{\mathbf{x}} = k(\mathbf{w})$
    Apply Eq. (5.6) to find the loss function:
    $\mathcal{L}(\mathbf{x}, \widehat{\mathbf{x}}) = \sum_{i=1}^{\mathcal{D}} \sum_{m=1}^{M} x_m \ln(\widehat{x}_m^{(i)})$
    Apply Adam optimization algorithm to optimize θ using initial parameters: η(learning rate), λ_1&λ_2 (the exponential decay rate for the 1st and 2nd moment estimates respectively), ε (a small constant value for numerical stability) to get the gradient $g_{_{Itr}}$:
    $g_{_{Itr}} \leftarrow \nabla_\theta \mathcal{L}(\mathbf{x}, \widehat{\mathbf{x}}, \theta)$
    use $g_{_{Itr}}$ to update $\theta_{_{Itr}}$ according to [134].
  end for
end for
Output: Return the up to date θ =0

baseline ones. Further, the constellations in each system have different phases to compensate for channel noise. In spite of this, there is no evidence that the proposed system learns better constellations phase in comparison with the baseline system since the distances between the constellations are not the same. Additionally, the SER performance improves with an increase in the number of IRS elements. The reason for this can be explained by looking at Fig.5.7 (a)-(c), which shows that even though the distance between the constellations of the proposed system in Fig.5.7 (b) and (c) is almost the same, the performance of $N = 16$ has a significant advantage over $N = 4$. The improvement in learning may be due to the better learning of the constellation phases, which may provide evidence that distance learning is not the only factor that is important, but also the constellation phase. Although we used 2×10^6 symbols in our simulation, it may be necessary to run more experiments in order to overcome the zero SER values.

Algorithm 2 IRS-assisted AE for inference

Initialization: create IRS-assisted communication model using θ parameters from **Algorithm 1**
IRS size: N [element]
E_b/N_o [dB] range
create all transmitter symbols: s_t
for each E_b/N_o **do**
  for each s_t **do**
    for 1 to N **do**
      produce the coded signal $\mathbf{z} = z(f(\mathbf{x}, \theta_T), \theta_{SR})$
      contaminate $\mathbf{z}$ (the coded signal) using channel noise layer to sample $w \sim p(\mathbf{w})$
      obtain the message probability vector $\widehat{\mathbf{x}}$ by decoding $\mathbf{w}$ using $k(\mathbf{w}, \theta_R)$
    end for
    apply Eq. (5.7) to find the expectation of all IRS elements probability messages: $\mathbb{E}[\widehat{\mathbf{x}}] = \frac{1}{N}\sum_{n=1}^{N} \widehat{\mathbf{x}}_n$
  end for
  transform the $\mathbb{E}[\widehat{\mathbf{x}}]$ to received signal s_r
  update, calculate and store the SER at the specific E_b/N_o
end for
store the system SERs at each E_b/N_o assisted by IRS size N
Output: Return all system SERs and plot SERs vs E_b/N_o assisted by IRS size N.
=0

In Fig.5.6 the result shows that our proposed AE communication system still outperforms the baseline in [4]. According to a comparison of the two systems with the same IRS size of N elements, the proposed system performs better, especially at lower IRS sizes. In particular, the proposed system with $N = 4$ is nearly as efficient as the classical IRS with $N = 16$ elements. Using this approach, IRS may be able to overcome its challenge of improving system performance by incorporating a large number of elements. In addition, increasing the number of IRS elements N reduces the gap between the two systems' performance. For instance, the performance gain between the two systems at $N = 4$ is much higher at $N = 16$ and at $N = 16$ the performance gain is higher than $N = 64$.

Fig.5.7 (a)-(f) Show how different modulation schemes such as BPSK and QPSK can be learned for different IRS sizes. Comparing Fig.5.7 (a) and (d) of the proposed system without IRS to the baseline in [3], it shows that the proposed system learned nearly the same constellations. While in Fig.5.7 (c) and (f), when the IRS ($N = 16$) assisted the communication system, the proposed system has augmented signal amplitudes and shifted constellation phases. Clearly,

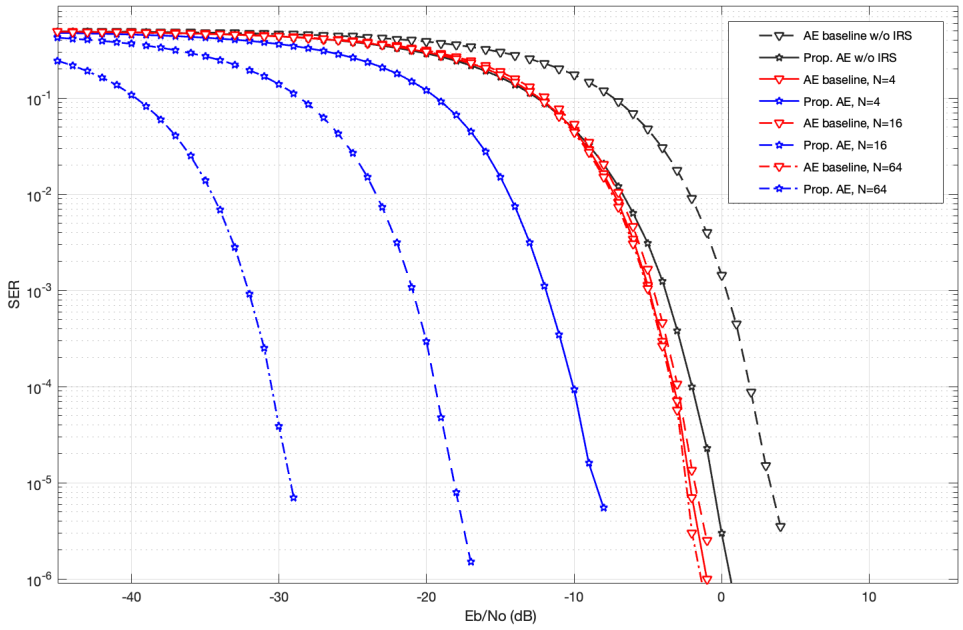


Figure 5.5: SER performance (BPSK) of the proposed IRS assisted AE-based scheme vs baseline [3].

this indicates that the proposed system learned both the phase and amplitude of the received signal when learning the IRS effect.

As a result of all of the above results, introducing the IRS deep layer and expectations of the message probabilities for all IRS N elements, the proposed design performs better than the conventional AE approach used in [3] as well as the conventional IRS in [4].

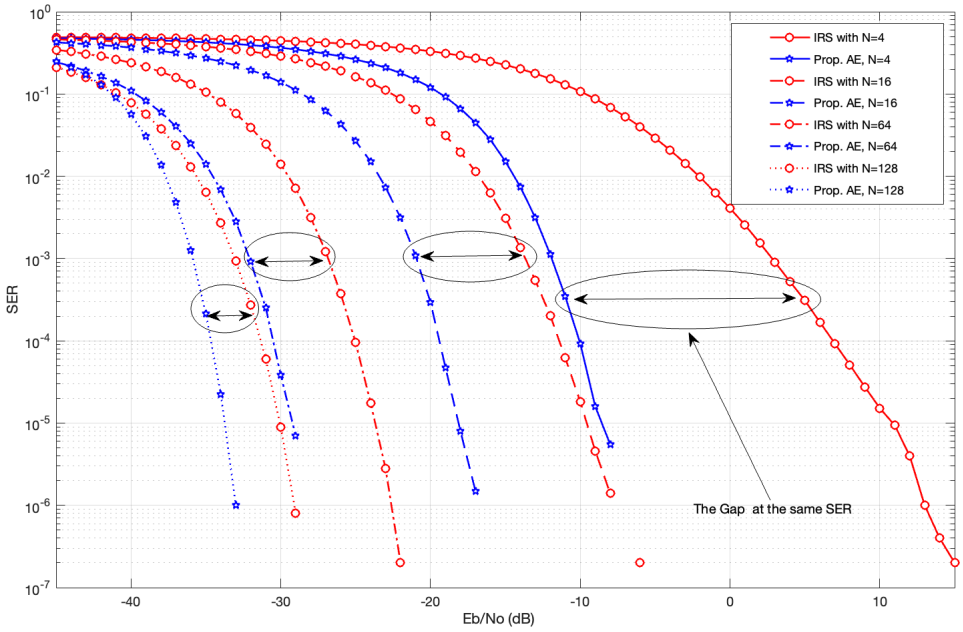


Figure 5.6: SER performance (BPSK) of proposed IRS assisted AE-based scheme vs baseline [4] for different IRS sizes.

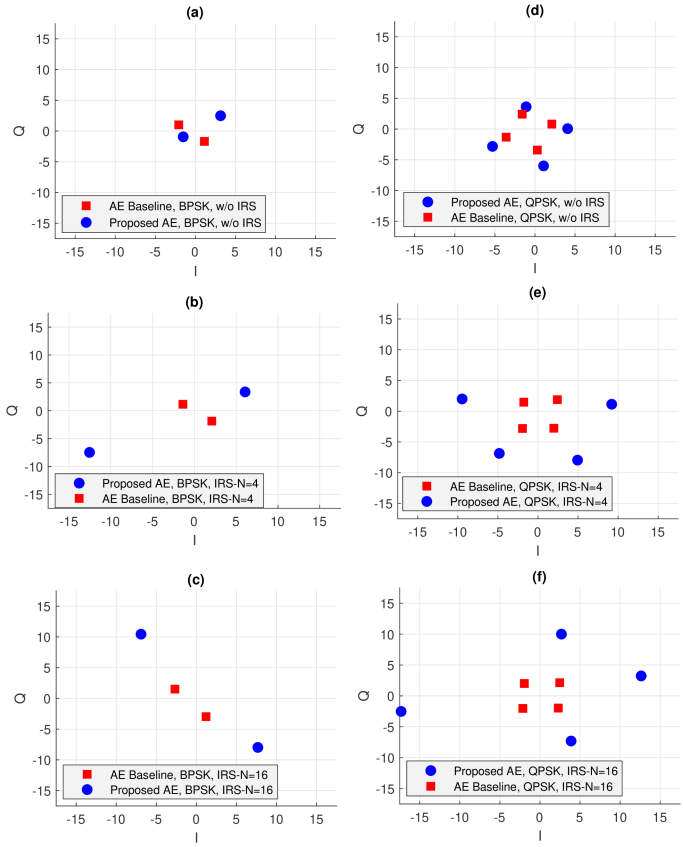


Figure 5.7: 2-D learned constellations presentation for IRS assisted AE-base scheme and baseline [3] for BPSK and QPSK modulations at different IRS sizes using AE-channels $c = 2$.

5.7. Chapter Summary

In this chapter, the classical IRS-assisted communication system has been transformed into an end-to-end AE. A piece of evidence has been provided to support that the proposed design models both the channels that the desired signal propagates through and the IRS augmentation effect on the signal as one DNN. The proposed DNN loss function shows the capability of optimizing the DNN parameters to reconstruct the transmitted signal by learning signal constellations representation pattern that reduces the wireless environment contamination impact on the received signal. The SER performance of the proposed system improves when the number of IRS elements is increasing at the same E_b/N_o. Lastly, the SER performance of the proposed system shows better performance in comparison to both the conventional AE and classical IRS. The next chapter introduces a novel approach using VAE as a probabilistic model to re-construct the transmitted symbol from LRVs' statistical parameters without sending the data bits out of the transmitter.

Chapter 6

Design of Innovative Variational Autoencoder Model for an End-to-End Communication System

This chapter is organised as follows: section 6.1 introduces the chapter. Section 6.2 states the related works, followed by the main contributions in section 6.3. The system model is presented in section 6.4. Section 6.5 introduces the experiment setup and the system training. The numerical simulation comparing the results with other works is presented in section 6.6. Finally, section 6.7 summarises the chapter. Part of the work in this chapter has been published in P.3 and P.4.

6.1. Introduction

Wireless networks and other related services are becoming more intelligent with innovative advances and unprecedented levels of computing capability. The advent of numerous unprecedented services, such as factories, self-driving cars, smart cities, and telemedicine and remote diagnostics, presents a challenge to the classical communication in terms of latency, flexibility, reliability, energy efficiency, and connection density. All of these technologies require new architectures, approaches, and algorithms in almost all layers of communications systems. An advanced AI-based approach can significantly improve the design and management of communication components. AI, represented by ML and DL, has attracted

tremendous attention as it has successfully transformed the manner in which humans work and communicate. This has been addressed in [136, 137]. Some of these techniques have been applied in the communication literature, have triggered extensive research, and have significantly impacted the solutions to some communication problems. Various emerging trends for the DL method are also considered based on information theory, probability, statistics, and solid mathematical modelling. The basic function of a communication system is to transmit a message, such as a bit stream, from the source to the destination over a channel through the accurate use of a transmitter and receiver. In order to achieve this optimally, the transmitter and receiver are segmented into strings of multiple independent blocks, each of which is responsible for a particular mini-task. Many approaches have been demonstrated in various applications such as modulation recognition [3], signal detection [132], channel coding [133, 138], channel decoding [29–31, 139–141], and channel estimation and detection [115, 142–147], and replacement of the total communication system with a novel architecture based on an AE. In [3, 148], the authors show a significant gain by introducing an AE as a communication system, in which the modulation and coding are jointly designed as one end-to-end DL model. The work in [3] showed how the use of block structures typically enables individual optimization, analysis, and control of each block, without the need for any domain-specific information; the end-to-end AE can achieve a performance similar to the conventional method in AWGN channels. However, the block-based approach is sub-optimal in certain cases [3]. Considering the DL-based communications system design, the optimization of end-to-end as one black box block is proposed in [3, 149].

All previous work has shown that the idea of end-to-end learning in communication systems has received widespread attention in the wireless communications community [150, 151]. In our work, we use generative models known as variational autoencoders (VAEs) [152,153], as they have been extensively used for unsupervised and semi-supervised DL. Moreover, since most of the current mobile systems generate unlabeled or semi-labelled data, the VAE is well suited to learning in wireless environments.

6.2. Related Works

As DL advances, the research paradigm can shift away from designing schemes using mathematical models to autonomously constructing end-to-end DL schemes based on observations of large quantities of data. For example, when DL is employed for image classification, feature detectors that are far more accurate than conventional detectors can be derived from a large set of image inputs using DNN structures. Therefore, in the age of DL, it starts with preparing, selecting, and pre-processing data to be used in the DNN structure. Then, determine the appropriate structure for the DNN. Lastly, interpreting the output of the DNN becomes increasingly important than developing analytic schemes from mathematical systems that typically contain assumptions necessary to enable analysis.

Recently, DL has been applied to many areas of wireless communications research. Besides improving conventional communication modules, DL-based end-to-end communication systems have recently been developed, in which DNNs represent both the transmitter and receiver. A framework with block structures under the AWGN channels was proposed in [3] and performs similarly to traditional approaches. There is also an end-to-end framework in the OFDM system [154], and singular value decomposition (SVD) precoding-based MIMO system [148], which view the channels as a group of independent sub-channels.

Moreover, Recent research has examined how to learn an end-to-end communication system without prior knowledge of channel models. A RL approach based on reinforcement learning was developed [155] to optimize the transmitter DNN without regard to the channel transfer function or CSI. The stochastic perturbation approach was used in [156] to design a model-free end-to-end communication framework. In [157], a conditional generative adversarial network (GAN) approach has been developed for building end-to-end communications, where the channel effects are modelled by a conditional GAN.

In contrast to other ML techniques that do not require communication resources, federated learning (FL) utilizes communication between the central server and distributed local clients in order to train and optimize the model. ML-based FL allows training models to be distributed between multiple clients, each with a certain amount of training data and coordinated through a central server. Therefore, the computation can be offloaded from the central server to the client.

In brief, in FL, the local clients communicate with the central server only using model parameters learned locally rather than raw data, preserving both privacy and communication overhead [158–160].

A part of the AI field is ML, which includes algorithms for classification, clustering, and dimensionality reduction (DR). Over the last decade, various classification algorithms have been developed, including Deep Convolutional Neural Networks (DCNNs) [161], and VAEs [152]. The VAE inherits the traditional AE architecture, meaning it is composed of two NNs, an encoder and a decoder, respectively. The encoder decreases the dimensionality of the inputs into a latent space. The decoder, on the other hand, can reconstruct the inputs from the latent space through learning. Thus, VAEs can be used for classification [162–164] and production [165–167]. Moreover, VAE can learn a data generation distribution that can take random samples from the latent space. It then generates unique images with features similar to those on which the network was trained after decoding the random samples using the decoder network. Using the Bayes rule [168], the VAE can learn the joint probability of input data and labels simultaneously. Bayesian inference is a method of statistical inference that provides a powerful framework for reasoning and prediction under uncertainty. However, the limitation of computing the posterior with only a few parametric distributions makes wider applications of Bayesian inference difficult [169]. Recently, to approximate the posterior by representing the variational distribution with a set of particles and update them through a deterministic optimization process, particle-based variational inference (ParVI) methods have been proposed [31, 170, 171]. Although the ParVI method can achieve computational efficiency and asymptotic accuracy, it restricts the fixed number of particles and lacks the ability to draw new samples beyond the initial set of particles [31]. Generally, variational inference and Markov chain Monte Carlo (MCMC) methods have been used to give tractable approximate inference, but these approaches bring their own set of challenges when the space's dimensionality is particularly high. Bayesian neural networks (BNNs) are a recent example of interest. These apply Bayesian inference to DNN training to provide a principled mechanism to analyze model uncertainty. The purpose of this is to represent the posterior of every parameter in all weight tensors from each layer of deep networks. Developing efficient computer strategies to estimate this intractable posterior with exceptionally high dimensionality, on the other hand, remains challenging.

On the basis of the above and the development of DL, semantic communication is again being considered a key technology and has received significant attention. As the 5G system has approached the Shannon limit, semantic communication aims to retain the successful transmission of semantic information by the source rather than the accurate reception of each bit or single symbol regardless of its meaning. Semantic communication is at the second level of communication based on Shannon and Weaver [172], aiming to accurately convey the semantic information of the transmission symbols rather than accurately recovering the transmitted information.

Recently, several semantic communication concepts have been developed based on NNs to replace conventional communication blocks. In [173], the conditional GAN was designed to represent channel effects, while in [32], a complete point-to-point communication system in the physical layer was developed using NNs. The authors of [174], show that the network can learn a projection function from feature space to a semantic embedding space in zero-shot learning (ZSL) models. The work in [175] developed a DL-based semantic communication system (DeepSC) for text transmission, with the aim of maximizing the capacity of the system and minimizing semantic errors, as it would recover the meaning of sentences rather than the bit or symbol error. Moreover, the authors in [176] proposed a semantic communication approach based on AE for the wireless relay channel (AESC) to extract and compress semantic information and reconstruct its semantic features. However, there are some key differences between semantic communication systems and conventional communication that can be defined as follows [175]:

- The design and optimization of the information transmission module in conventional systems are contained in the transceiver, unlike the semantic system, where the whole information processing block is jointly designed from the source information to sink.

- Recovering the exact data is the focus of conventional communication systems; however, semantic communication systems are intended for transmission decisions.

- Conventional communication systems compress data in the entropy domain, while semantic communication systems process data in the semantic domain.

6.3. Main Contributions

In this chapter, a new approach has been proposed and investigated with the help of a VAE as a probabilistic model to reconstruct the transmitted symbol without sending the data bits out of the transmitter. Our main contributions are summarized as follows:

- We propose an end-to-end communication system that represents the symbol as packet hot vector (PHV) and operates over BPSK modulation in AWGN channels, where modulation and demodulation are performed by a DNN based on a VAE architecture.
- We extend our experiment to investigate the QPSK modulation, Rayleigh, Rician fading channels, shadowing, and Doppler effect for a limited range of doppler frequency shifts and phase offsets.
- While the baseline AE uses 4 and 7 channels in [3] to achieve their results. In our work, we efficiently use two channels only to achieve better performance than AE baseline.
- Our work considers a VAE with two LRVs, and a simple classifier can reconstruct the transmitted message by sending only the LRVs' parameters and the message error rate (MER). The result shows that the performance of our proposed system is better than that of the existing classical scheme.

6.4. System Model

In this work, the wireless communication system model has a simple setup to allow the reader to follow the proposed idea. Our goal is to design a probabilistic model that can reconstruct the transmitted information without sending the exact bits or the deterministic transformed bits of the exact symbol (e.g, channel coding using Hamming codes), but by transmitting the statistical parameters of the LRVs through the physical layer rather than sending the data bits of the original symbol out of the transmitter.

6.4.1. Variational AutoEncoder (VAE)

A brief description of the basic VAE, on which this work builds, are required to grasp what follows clearly. The VAE is a popular generative model, allowing us to solve problems in the framework of probabilistic graphical models with latent variables [152, 177]. As opposed to AE, VAE provides the generative capability to the entire space by addressing the problem of non-regularized latent space. Rather than output vectors in the latent space, the encoder of VAE outputs parameters for a predefined distribution in the latent space for each input. In the VAE, a constraint is applied to this latent distribution, forcing it to be a normal distribution. By imposing this constraint on the latent space, it ensures that it is regularized.

VAEs can be considered as two independently parameterized models: the recognition model, known as the encoder, and the generative model or decoder. The encoder delivers an approximation to its posterior over latent random variables to the decoder, which is required to update its parameters inside the iteration of expectation maximization learning. Conversely, the decoder is a scaffolding of sorts for the encoder to learn meaningful representations of the data besides class-labels. In other words, the VAE helps the encoder infer the distribution of original data rather than the original data itself. By employing a properly designed object function, the distribution of original data can be encoded into certain low-dimensional distributions. Similarly, the decoder training allows the decoder to transform the distributions into the approximate original data distribution to obtain a new sample that represents the reconstruction of the original ones.

Moreover, as probabilistic models, VAEs also contain data and unknowns. Therefore we need to assume some level of uncertainty around this aspect of the model. This uncertainty can be specified in terms of a conditional probability distribution, where the model can contain both discrete and continuous variable values. In addition, between these variables, this probabilistic model is able to specify all correlations and higher-order dependencies in the form of a joint probability distribution.

As shown in Fig. 6.1, VAEs can learn the stochastic mappings between the observed x-space that has distribution $q_{\mathbf{D}}(\mathbf{x})$ and the latent z-space. The generative model learns the joint distribution $p_{\theta}(\mathbf{x},\mathbf{z})$, which is factorized as

$p_\theta(\mathbf{x},\mathbf{z})$=$p_\theta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})$ with a prior distribution over latent space $p_\theta(\mathbf{z})$ and a stochastic decoder $p_\theta(\mathbf{x}|\mathbf{z})$. The inference model or the stochastic encoder $q_\phi(\mathbf{x}|\mathbf{z})$ approximate the true but intractable posterior $p_\theta(\mathbf{x}|\mathbf{z})$ of the generative model [177].

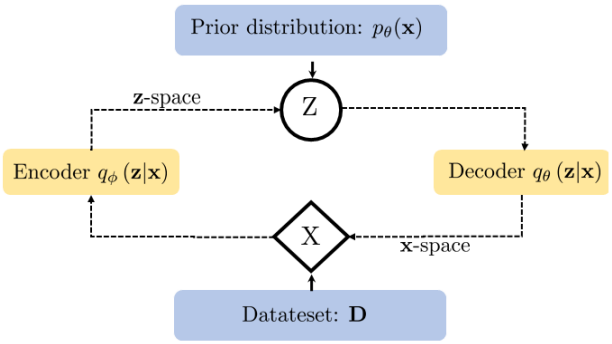


Figure 6.1: General VAE stochastic mapping.

Specifically, we use the vector $(\mathbf{x})$ to represent the set of all observed variables that we want to model its joint distribution. We assume the observed variable $(\mathbf{x})$ from an unknown underlying process is a random sample that has unknown probability distribution $p^*(\mathbf{x})$. To approximate this underlying process, we used a chosen model $p_\theta(\mathbf{x})$, with parameters θ which can be written as:

$$\mathbf{x} \sim p_\theta(\mathbf{x}). \tag{6.1}$$

To find the value for the parameter θ, we used the *learning*[1] process, which is the most commonly used search process. Since the probability distribution function is given by the model $p_\theta(\mathbf{x})$ and approximates the true distribution of the data, denoted by $p^*(\mathbf{x})$, therefore, for any observed $(\mathbf{x})$:

$$p_\theta(\mathbf{x}) \approx p^*(\mathbf{x}). \tag{6.2}$$

Often, in the case of classification or regression problems, we are interested in a learning conditional model such as $p_\theta(\mathbf{y}/\mathbf{x})$ that approximates the underlying conditional distribution $p^*(\mathbf{y}/\mathbf{x})$, where the distribution of the value over the variable y is conditioned on the value of the observed variable x. In this case, x

[1]*learning*: In terms of ML, the concept of learning can be formulated as Tom Michell defines it, as a "problem of searching through a predefined space of potential hypotheses for the hypothesis that best fits the training examples." [178]

is the input of the model. As in the previous paragraph, the model p_θ $(\mathbf{y}/\mathbf{x})$ is chosen and optimized to be close to the unknown underlying distribution for any x and y:

$$p_\theta(\mathbf{y}/\mathbf{x}) \approx p^*(\mathbf{y}/\mathbf{x}). \tag{6.3}$$

One of the most common examples of conditional modelling is image classification, where $(\mathbf{x})$ is an image, and $(\mathbf{y})$ is the image's class that we want to predict.

We can extend the models discussed above into directed models with latent variables, where the latent variables can be defined as variables that are part of the model but are not part of the data-set, and which, therefore, we do not observe. Normally, we use $\mathbf{z}$ to denote the latent variables. In the case of unconditional modelling of the observed variable $\mathbf{x}$, we can represent the directed graphical model by a joint distribution $p_\theta(\mathbf{x}\ ,\ \mathbf{z})$ over the observed variable $\mathbf{x}$ and the latent variables $\mathbf{z}$. The marginal distribution over the observed variables $p_\theta(\mathbf{x})$ can be written as:

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z}. \tag{6.4}$$

The model $p_\theta(\mathbf{x}\ ,\ \mathbf{z})$ can be conditioned in some context, such as $p_\theta(\mathbf{x}\ ,\ \mathbf{z}\ |\ \mathbf{y})$ and for this, we use the term "deep latent variable model" (DLVM), which is when the distributions are parameterized by NNs. The advantage of the DLVM is that when each factor in the directed model, whether its prior or conditional distribution, is relatively simple, the marginal distribution $p^*(\mathbf{x})$ can be very complex. This expression makes the DLVM attractive for approximating complicated underlying distributions. One of the most common and simplest DLVM is known as factorization, which can be defined as follows:

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{z}) p_\theta(\mathbf{x}|\mathbf{z}). \tag{6.5}$$

Fig. 6.2 shows a simple schematic of computational flow in the VAE with the evidence lower bound (ELBO), which is the optimization objective of the VAE.

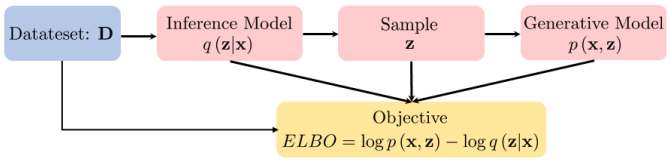


Figure 6.2: Simple schematic of computational flow in a VAE.

More details about the VAE can be found in [152, 177, 179].

6.4.2. Wireless System Models

We build up an end-to-end communication system that consists of a transmitter sending the desired signal to the receiver, as shown in Fig. 6.3. We assume that the wireless channels have AWGN. The equation below formulates the received signal vector $\mathbf{s}_r$:

$$\mathbf{s}_r = \mathbf{H}\mathbf{s}_{_d} + \mathbf{n}_o, \tag{6.6}$$

where $\mathbf{H} = \text{diag}\ (\mathbf{h})$ is the channel coefficient vector, $\mathbf{h} = [\mathbf{h}_1, ...\mathbf{h}_i, ..., \mathbf{h}_N], h_i \in \mathbb{C}^{1\times 1}$, $\mathbf{s}_{_d}$ is the desired transmitted signal vector for propagated s_d from the transmitter to the receiver, and $\mathbf{n_o}$ is the AWGN noise vector, $n_0 \sim \mathcal{CN}(0, \sigma)$, where $N_o = \sigma^2$ is the noise power variance that contaminates the transmitted signal power as shown in Fig.6.3.

By definition, the signal-to-noise ratio (SNR) is :

$$\Gamma = S_r/N_o, \tag{6.7}$$

where S_r is the power of the desired signal received, and N_o is the AWGN power. For t bits per symbol in E_b/N_o, (6.6) and (6.7) can be written as:

$$\gamma = \frac{S_r}{N_o \times t}. \tag{6.8}$$

Moreover, $h_i \in \mathbb{C}^{1\times 1}$ sampled for Rayleigh distribution, Rician distribution, and long normal Shadowing for Rayleigh, Rician and Shadowing models, respectively [180]. While in the Doppler model, we use the theoretical flat Doppler spectrum $S(f)$, where $S(f) = \frac{1}{2f_d}$, and phase shift ϕ_d [181].

6.4.3. The VAE as a Wireless System Model

The proposed VAE model design learns the noise, multi-path, line of sight, and non-line of sight effects features using a directed probabilistic graph model (DPGM) as in Fig. 6.5, where z represents the LRVs that are used to infer the signal features from the PHV. Using this method, the relation between the transmitted signal and the received signal patterns can be presented using inferred

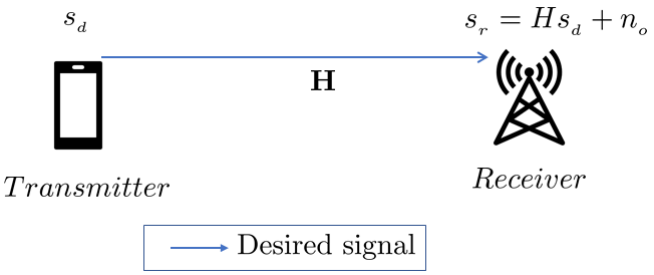


Figure 6.3: Simple wireless system with AWGN channel.

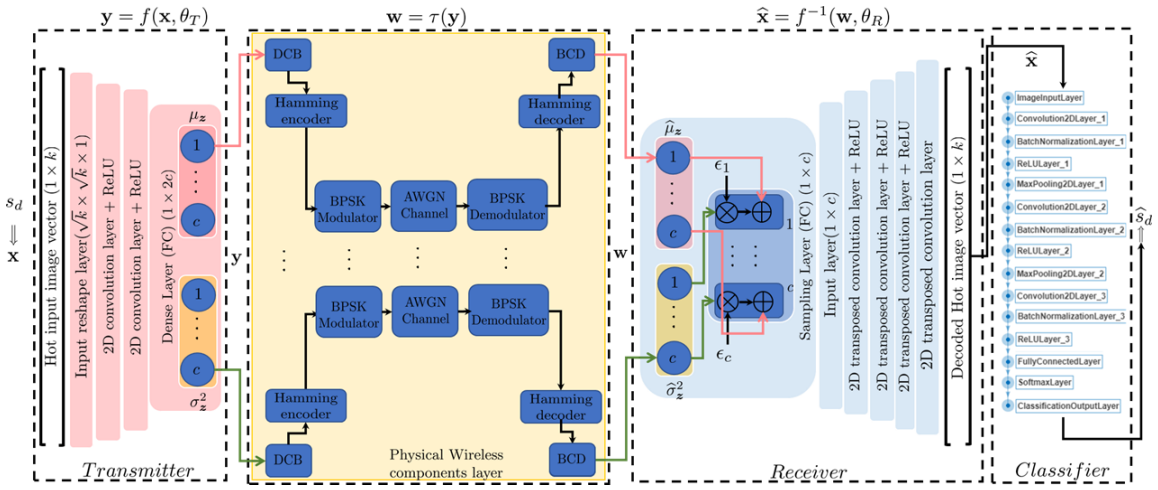


Figure 6.4: End-to-end wireless communication architecture consists of the VAE and classifier DNNs.

LRVs.

Inspired by the semantic level communication and VAE, our work considers the use of variational inference for generative modelling; however, we reinterpret the variational inference from a new perspective. We use generative modelling, which refers to the process of valid samples from $p(\mathbf{x})$. Fig. 6.5, shows our generative model. In this work, the samples of $\mathbf{x}$ are generated from a latent variable $\mathbf{z}$, and θ represents the associated parameters, while the solid lines denote the generative model p_θ $(\mathbf{z})$ p_θ $(\mathbf{x} \mid \mathbf{z})$. For example, to generate valid samples of $\mathbf{x}$, we first sample $\mathbf{z}$, then use $\mathbf{z}$ and θ to generate $\mathbf{x}$. The dashed lines represent the inference procedure with a variational approximation of the intractable posterior p_θ $(\mathbf{z} \mid \mathbf{x})$. Moreover, we apply DL that is proposed by a stochastic optimization-based technique to approximate the inference $p(\mathbf{z} \mid \mathbf{x})$ with appropriate prior on $p(\mathbf{z})$ using an encoder network $q_\phi(\mathbf{z} \mid \mathbf{x})$. After that comes the decoder network $p_\theta(\mathbf{x} \mid \mathbf{z})$ to compute the reconstruction $\hat{\mathbf{x}}$ of the message $\mathbf{x}$, where this will be learned

during the training phase. Given a NN model with sufficient learning capability and good prior distribution $p(\mathbf{z})$, this high-capacity model will approximate the posterior by $q_\phi\ (\mathbf{z} \mid \mathbf{x}) \approx \mathrm{p}_\theta(\mathbf{z} \mid \mathbf{x})$. Since this model is structured as an encoder-decoder, the technique is known as autoencoding variational Bayes (AVB), where the expected marginal likelihood $p_\theta(\mathbf{x})$ of the datapoint $\mathbf{x} \in \mathcal{X}$, under an encoding function, $q_\phi(.)$, can be computed as in [182]:

$$\mathbb{E}_{p(\mathbf{x})} \log p_\theta\left(\mathbf{x}\right) = \mathbb{E}_{p(\mathbf{x})} \mathcal{D}_{KL}\left(q_\phi\left(\mathbf{z}|\mathbf{x}\right)||p_\theta\left(\mathbf{z}|\mathbf{x}\right)\right) + \mathcal{L}_{\theta,\phi}\left(\mathbf{x}\right). \tag{6.9}$$

The first term in (6.9) is the Kullback-Leibler (KL) divergence between $q_\phi\left(\mathbf{z}|\mathbf{x}\right)$ and $p_\theta(\mathbf{z}|\mathbf{x})$.

The second term in (6.9) is called the evidence lower bound (ELBO):

$$\mathcal{L}_{\theta,\phi}\left(\mathbf{x}\right) = \mathbb{E}_{p(\mathbf{x})} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left(\log \frac{p_\theta\left(\mathbf{x},\mathbf{z}\right)}{q_\phi\left(\mathbf{z}|\mathbf{x}\right)}\right), \tag{6.10}$$

and

$$\mathcal{D}_{KL}\left(q_\phi\left(\mathbf{z}|\mathbf{x}\right)||p_\theta\left(\mathbf{z}|\mathbf{x}\right)\right) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left(\log \frac{q_\phi\left(\mathbf{z}|\mathbf{x}\right)}{p_\theta\left(\mathbf{z}|\mathbf{x}\right)}\right). \tag{6.11}$$

We have to maximize the $\mathcal{L}_{\theta,\phi}\left(\mathbf{x}\right)$ by minimizing the $\mathcal{D}_{KL}\left(q_\phi\left(\mathbf{z}|\mathbf{x}\right)||p_\theta\left(\mathbf{z}|\mathbf{x}\right)\right)$ in order to maximize the penalized likelihood of the reconstruction of x from z using:

$$\mathcal{L}_{\theta,\phi}\left(\mathbf{x}\right) = \mathbb{E}_{p(\mathbf{x})} \log p_\theta\left(\mathbf{x}\right) - \quad \mathbb{E}_{p(\mathbf{x})} \mathcal{D}_{KL}\left(q_\phi\left(\mathbf{z}|\mathbf{x}\right)||p_\theta\left(\mathbf{z}|\mathbf{x}\right)\right). \tag{6.12}$$

Moreover, since backpropagation through a random operation is not possible in the training stage, we use the reparameterization trick to move the random sampling operation to an auxiliary variable ε that is shifted by the mean μ_i and scaled by the standard deviation σ_i, respectively, representing the distribution Φ that the network is trying to learn, as in Fig. 6.6. This allows backpropagation through the deterministic nodes f, z, Φ. The idea here is that sampling from $N(\mu_i, \sigma_i^2)$ is the same as sampling from $(\mu_i + \varepsilon.\sigma_i)$, where $\varepsilon \sim N(0,1)$.

Next, we describe the architecture of the VAE in the proposed end-to-end wireless communication system shown in Fig. 6.4 and compare this transformation with a simple wireless system as shown in Fig. 6.3.

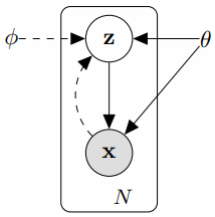


Figure 6.5: DPGM for used VAE.

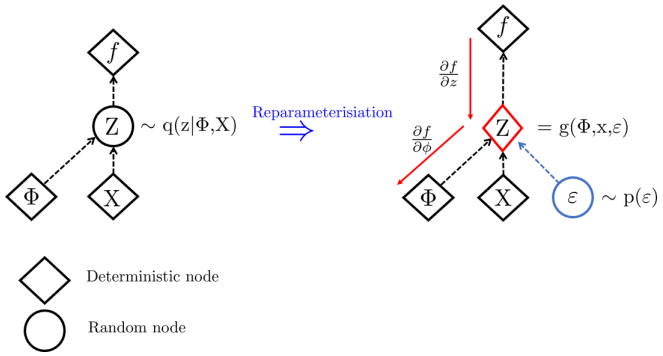


Figure 6.6: Reparameterization trick used for training VAE.

VAE input

The hot vector in [3] can be replaced with a new concept known as the PHV in the same way that [179] used to represent the constellation of a symbol. However, in this work, we present the symbol as a packet of ones and zeroes where the inputs s_0 and s_1 to the transmitter are encoded as a one-PHV $1s \in \mathbb{R}^M$. The sent BPSK message s_0 has been presented by a packet of B bits. This packet consists of K sub-packets, where each sub-packet $k_i, i \in \{1, ..., K\}$ contains b bits. For example, this means that the total length of our PHV is $1 \times bK$. Let the space of possible messages be $M = 2^{bK}$ and bK be the necessary number of bits to represent each message m. Then transmit input message $s_t \in \{1, ..., m, ..., M\}$, where M is the space size of the possible messages as in Fig.6.7.

VAE encoder

Each PHV $\mathbf{x}$ fed into the input layer will be transformed by $f : \mathbb{R}^{1\times bK} \rightarrow \mathbb{R}^{1\times c}$, where c is the dimension of the last layer in the encoder. Looking at Fig.6.4, the encoder layers include two-dimensional convolution (2DConv) layers, each of which is configured with several filters (each filter has a size of $\hbar$ height and ϖ width). The features output by each layer is mapped to a number of filters ν_1 and ν_2,

respectively. The filter shifts by ς strides at each convolution step, while the padding size $\wp$ can be calculated using: size $(2DConv) = \frac{k\check{}\hbar\varpi+2\wp}{\varsigma} + 1$, to keep the output size equal to the input. A rectified linear unit (ReLU) layer is used after each 2DConv to eliminate any negative output value. A final fully connected (FC) layer was added to the encoder with the dimension of $1 \times 2c$. The output of the FC layer is divided into two sets $\boldsymbol{\mu}_z = [\mu_1, ..., \mu_c]$ and $\boldsymbol{\sigma}_z = [\sigma_{1+c}, ..., \sigma_{2c}]$, which represents the latent variables' distributions parameters (the expectation and the variance, respectively).

The transformation can be formulated using the DNN hyperparameters θ_T :

$$\mathbf{y}_n = f(\mathbf{x}_n, \theta_T), \tag{6.13}$$

where $\mathbf{x}_n \in \mathcal{X}$, $\mathbf{x}_n$ is the input data point and $\mathbf{y}_n$ is the output of the FC layer which has decimal format. After this, the FC decimal output values use the physical decimal to binary converter (DCB) component to start sending the LRVs' distribution parameters over the physical layer.

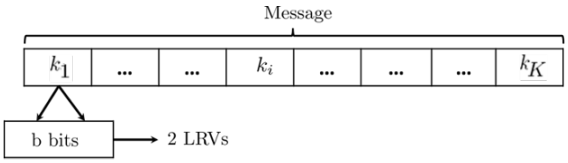


Figure 6.7: Packet representation for VAE input.

Physical Medium

In this work, our unique approach is to explain the practical aspect of implementing an end-to-end system that includes the realization of the physical wireless transmission and the receiving components, such as the digitization of μ and σ values for each LRV, the modulator, demodulator, and AWGN channel:

- Decimal-coded binary (DCB) and binary-coded decimal (BCD) converters: In the DCB component, the received decimal integer part will be represented by b_y number of bits and the same for the fractional part of the decimal value. In addition, an extra bit for the sign has been added as the most significant bit (MSB), which means $2 \times b_y + 1$ is the final length of bits code that the modulator receives. After signal demodulation, the BCD will use the binary decoded bits to convert it back as a decimal integer and fraction parts before

combining both using a fixed point radix to retrieve the decimal value. This proposed method eliminates any digitization error for the $\mathbf{y}$ values when b_y length satisfies the required significant figures for precision sf.

- BPSK modulation and demodulation components: The BPSK is used to modulate the output of the DCB using a standard modulation, whereas the demodulated output is used to feed the BCD input.
- AWGN noise channel: the physical AWGN noise is $\sim \mathcal{CN}(0, \xi)$, where ξ is the fixed standard deviation value that contaminates the amplitude and phase of the received signal.

The number of channels in the physical wireless component medium has the dimension of $\mathbb{R}^{1\times c}$, and c is the number of channels that the proposed communication system uses to send one message out of the 2^{bK} messages. The end-to-end rate of this communication can be measured by $r_{_{E2E}} = \frac{bK}{c}$ [bits/channel use]. However, over the physical wireless components medium, the rate of the physical transmission is $r_{_{PH}} = (2b_{_y} + 1)$ [bits/channel use]. This leads to the compression rate (CR) formula:

$$CR = \frac{(2b_{_y} + 1)c}{2^{bK}}. \tag{6.14}$$

The channel noise is an AWGN due to the assumption that the main source of the noise is on the receiver side [3]. The channel uses a fixed variance $\xi^2 = (E_b/N_o)^{-1}$ and is characterized as a distribution $\mathcal{N}(0, \xi^2\mathbf{I})$, where (E_b/N_o) is the energy per bit E_b to the ratio of the spectral density of noise power N_o that contaminates the desired signal at the receiver after converting the values from binary to a decimal using the BCD.

VAE Decoder

The BCD output of the physical medium represents the LRVs' contaminated expectation, and variance decimal parameters vector values as a function of E_b/N_o are $\widehat{\boldsymbol{\mu}}_z(E_b/N_o) = [\widehat{\mu}_1(E_b/N_o), ..., \widehat{\mu}_c(E_b/N_o)]$ and $\widehat{\boldsymbol{\sigma}}_z(E_b/N_o) = [\widehat{\sigma}_{1+c}(E_b/N_o), ..., \widehat{\sigma}_{2c}(E_b/N_o)]$, respectively. In this work, we proposed to use the sampling layer inside the receiver to realize a practical architecture of the end-to-end wireless system. The dimensions of the sampling input layers are equal

to those of the last encoder output layer $1 \times 2c$. However, for the following layers, the reparameterization trick is necessary to allow the VAE to perform the backpropagation at the training phase and to sample the $\widehat{z}$ as shown in Fig. 6.6, and has been formulated using $\boldsymbol{\epsilon} \in\sim [\mathcal{N}_1(0,1), ..., \mathcal{N}_c(0,1)]$ as:

$$\widehat{\boldsymbol{z}}(E_b/N_o) = \boldsymbol{\mu}_z(E_b/N_o) + \boldsymbol{\sigma}_z(E_b/N_o) \odot \boldsymbol{\epsilon}. \tag{6.15}$$

At high E_b/N_o values, $\widehat{\boldsymbol{z}} = \boldsymbol{z}$ as a result of eliminating any contamination of the z values, due to the AWGN channel effect, is mathematically proved by:

$$\lim_{E_b/N_o \to \infty} \widehat{\boldsymbol{z}}(E_b/N_o) = \boldsymbol{z}, \tag{6.16}$$

which is the input of the decoder that is transformed back to $f^{-1} : \mathbb{R}^{1\times c} \to \mathbb{R}^{bK}$ to reconstruct the input symbol s as $\widehat{s}$. The transformation can be formulated using the DNN hyperparameters θ_R :

$$\widehat{\mathbf{x}} = f^{-1}(\tau(f(\mathbf{x}, \theta_T)), \theta_R). \tag{6.17}$$

The DNN consists of one input layer, three transposed 2DConv layers, and a ReLU layer that is used to eliminate the negative values at each output. Lastly, a 2DConv is used to reconstruct the transmitted image.

6.4.4. Simple DNN Image Classifier

To classify the final reconstructed symbol $\widehat{\mathbf{x}} \to s_d \in \{1, ..., m, ..., M\}$, a simple DNN classifier has been used. Fig.6.4, shows the architect of the classifier block using convolution, batch normalization, ReLU and max-pooling layers to extract the feature of $\widehat{\mathbf{x}}$. The classifier output layer learns the final message $\hat{s_d}$ from the output size of the previous fully connected and softmax layers with output size M possible messages.

6.4.5. Proposed Numerical Performance Measurement Methods for the New End-to-End Wireless System

To measure the performance of the proposed end-to-end (E2E) VAE wireless system, we suggested the following methods:

- BER_{E2E} definition: This is the ratio of bits error of the transmitted PHV

from transmitter to receiver

$$BER_{_{E2E}} = \frac{\sum_{n=1}^{N}\sum_{i=1}^{bk}(|x_{_i} - \widehat{x}_{_i}|)}{NbK}, \tag{6.18}$$

where N is the number of transmitted PHVs (Symbols). x_i and $\widehat{x_i} \in \{1, ..., bK\}$ bits that produced by converting $\mathbf{x}$ and $\widehat{\mathbf{x}}$ from decimal to binary respectively.

- $BER_{_{PH}}$ definition: This is the ratio of bits error of the transmitted LRVs values between the DCB and BCD components.

$$BER_{_{PH}} = \frac{\sum_{n=1}^{N}\sum_{i=1}^{(2b_y+1)}(|DCB(y_i) - DCB(BCD(w_i))|)}{N(2b_{_y}+1)}. \tag{6.19}$$

- MER definition: This is the ratio of the wrongly classified messages at the receiver to the transmitted ones.

$$MER = \frac{\sum_{n=1}^{N}(s_d - \widehat{s}_d)}{N}. \tag{6.20}$$

It is important to mention that the SER is the analogy of the proposed MER measurement in classical wireless communication. More discussion regarding this point can be found in Section 6.6. However, the most important of the three methods is the MER, because it measures the final ratio of the correctly received messages out of the total transmitted ones, which is the ultimate goal of the proposed system.

6.5. Experiment Setup, End-to-End Wireless System Training and Simulation

6.5.1. Experiment Setup

The main parameters for the VAE, classifier and physical wireless component layers are summarized in Table. 6.1

Table 6.1: Parameters used for simulations

Parameter	**Value**
Modulation scheme	BPSK and QPSK
AWGN channel noise level type	$E_b/N_o \in [0, 9]$ [dB]
Channel Block coding type	Hamming (1,3)
DCB No. of bits b_y for Inte & Frac parts	b_y=8 [bits]
BCD decimal resolution	No. of of SF=6
No. of transmitted packet ho vectors	1×10^6 messages
VAEs main training parameters	E_b/N_o=7 dB
	Epo (epochs)=50
	Batch size=64
	Iteration=300/epoch
	Validation data size= 4000
VAE adaptive moment estimation (Adam) optimizer parameters	η_V=0.001
	No. of LRVs $\mathbf{z} = 2$
	(λ_1, λ_2)=(0.9, 0.999)
	$\varepsilon_V = 1^{-7}$
Image classifier main training parameters	E_b/N_o=0 dB
	Epo (epochs)=50
	Batch size=128
	Validation data size= 500
Image classifier SGDM optimizer parameters	η_c=0.001
	Iteration=170/epoch
	Υ=0.9

6.5.2. Classifier Training

A classifier stochastic gradient descent with momentum (SGDM) training type is used to train PHVs under AWGN contamination with a value of $E_b/N_o = 0$ dB to produce the final retrieved sent message. The SGDM algorithm can oscillate along the path of the steepest descent towards the optimum. Adding the momentum term with the contribution factor Υ to the parameter update is one way to reduce this oscillation as in (6.15). **Algorithm 3** describes the classifier training process [183].

$$\theta_{c_{l+1}} = \theta_{c_l} - \eta_c \nabla \mathcal{L}(\theta_{c_l}) + \Upsilon(\theta_{c_l} - \theta_{c_{l-1}}), \tag{6.21}$$

where θ_{c_l} is the vector of weight and bias parameters for the DNN classifier in iteration l, η_c is the learning rate, and $\mathcal{L}(\theta_{c_l})$ is the loss function, while $\nabla \mathcal{L}(\theta_{c_l})$ is the gradient of the loss function used to train the entire training set.

Algorithm 3 Classifier training

Initialization: {Epo,Itr, $\mathbf{x}$, N_o, η_c, Υ, θ}, where
Epo: number of epochs training.
Itr: number of iterations per epoch.
$\mathbf{X}$: training image hot vectors.
$\mathbf{s}_d$:the desired output message.
N_o: noise sample $\sim \mathcal{N}(0, \xi^2)$
θ_c: DNN perceptron weights and biases matrix.
for each Epo **do**
  for each Itr **do**
    The input layer passes the PHV values to the 2DConv layer.
    The 2DConv layer produces the first features map.
    The output of the 2DConv layer passes the batch normalization to speed up the training and reduce the sensitivity of network initialization. Then the output passes the ReLU layer to remove any negative values.
    To reduce the spatial size of the feature map and redundant spatial information, the ReLU output uses the max-pooling layer to down-sample the input.
    Repeat steps 5 to 7, to fine-tune the detection of the important features in the message. (The gradient threshold = + ∞)
    Apply SGDM algorithm to optimize θ_c as in (6.21) using initial parameters: η_c (learning rate), Υ (the momentum contribution factor) to get the gradient $g_{_{Itr}}$:
    $g_{_{Itr}} \leftarrow \nabla_{\theta_c} \mathcal{L}(\mathbf{x}, \widehat{\mathbf{x}}, \theta_c)$
    use $g_{_{Itr}}$ to update $\theta_{_c}$ according to [183].
  end for
end for
Output: Return the up-to-date θ_c and save the DNN "Classifier-PHV" model. =0

6.5.3. The VAE Training

VAE training aims to reconstruct the sent PHV from a meaningful continuous space produced by the LRVs $\mathbf{z}$ ranges using the ELBO as in:

$$\min_{\theta_V} ELBO = \mathbb{E}[\mathcal{L}(\theta_V) + \beta \times \mathcal{KL}(\theta_V)], \tag{6.22}$$

where

$$\min_{\theta_V} \mathcal{L}(\theta_V) = \sum_{1}^{k} \frac{1}{2}(\widehat{\mathbf{x}} - \mathbf{x})^2. \tag{6.23}$$

$$\max_{\theta_V} \mathcal{KL}(\theta_V) = \sum_{1}^{k} \frac{-1}{2}(1 + \log(\sigma_z) - \mu_z^2 - e^{\sigma z}). \tag{6.24}$$

However, unlike the existing references, the contamination of the LRVs' inferred parameters occurs at the transmitted binary (not decimal values) level bits while it is propagated through the wireless channel to imitate the practical aspects of the experiment. In addition, the sampling layer has been moved to the receiver

side to produce the contaminated LRV's z values from the received contaminated LRVs' inferred parameters. The optimization algorithm used for DL networks wights is Adam with an added momentum term. It keeps an element-wise moving average of both the parameter gradients and their squared values [100]. The VAE Training was done at a fixed value of $E_b/N_o = 7$ dB with a learning rate of 0:001 and batch size=64. More details about the training setup will be illustrated in section 6.6.

Algorithm 4 describes the proposed VAE training process.

Algorithm 4 VAE end-to-end wireless system training

Initialization: {Epo,Itr, $\mathbf{x}$, N_o, $\eta_{_V}$, $\theta_{_V}$}, where
Epo: number of epochs training.
Itr: number of iterations per epoch.
$\mathbf{x}$: training input vector. In addition, the desired output.
N_o: noise sample $\sim \mathcal{N}(0, \xi^2)$
$\theta_{_V}$: DNN weights and biases matrix for θ_T and θ_R.
n: number of re-sampled PHV at the receiver,
for each Epo **do**
  for each Itr **do**
    use $\mathbf{x}$ for input layer to produce $\mathbf{y} = f(\mathbf{x}, \theta_T)$
    use $\mathbf{y}$ for physical wireless layer to produce $\mathbf{w} = \tau(\mathbf{y})$
    use $\mathbf{w}$ for sampling layer input to produce the LRVs $\mathbf{z}$ values using (6.15).
    use the sampling layer output to reconstruct the PHV :
    $\widehat{\mathbf{x}} = f^{-1}(\mathbf{w}, \theta_R)$
    Apply (6.25) to find the ELBO:
    Apply Adam optimization algorithm to optimize $\theta_{_V}$ using initial parameters: $\eta_{_V}$(learning rate), $\lambda_1 \& \lambda_2$ (the exponential decay rate for the 1st and 2nd moment estimates respectively), $\varepsilon_{_V}$ (a small constant value for numerical stability) to get the gradient $g_{_{Itr}}$:
    $g_{_{Itr}} \leftarrow \nabla_{\theta_{_V}} \mathcal{L}(\mathbf{x}, \widehat{\mathbf{x}}, \theta_{_V})$
    use $g_{_{Itr}}$ to update $\theta_{V_{Itr}}$ according to [100].
  end for
end for
Output: Return the up-to-date $\theta_{_V}$ and save the DNN "VAE-Wireless" model. =0

6.5.4. End-to-End Wireless System Simulation Realization

Once both the $'$Classifier$'$ and $'$VAE $-$ Wireless$'$ models have been trained, the two models cascade as in Fig. 6.4, and then the real data transmission starts. In this experiment, 10^6 PHVs have been sent from the transmitter through the VAE-encoder, physical wireless component layer, and VAE-encoder and finally pass the classifier to each under observation E_b/N_o. The proposed system has a novel method to re-sample the retrieved message for N times using parallel computing

techniques and hardware such as graphics processing units (GPUs), then finding the mode (the data value with the highest count) of the N re-sampled messages $(\widehat{s}_d^n)_{n=1}^N$:

$$\widehat{s}_d = \text{ mode } \left((\widehat{s}_d^n)_{n=1}^N\right). \tag{6.25}$$

Algorithm 5 describes the realization process.

Algorithm 5 VAE end-to-end wireless system realization

Initialization: {$\mathbf{x}$, Exp, E_b/N_o, N, $'$Classifier$'$ model, $'$VAE − Wireless$'$ model}, where
Exp: number of transmitted messages
$\mathbf{x}$: Test PHV for each Exp messages.
E_b/N_o : the range of power contamination at the physical wireless layer
n: number of re-sampled PHV at the receiver,
for each E_b/N_o **do**
  for each Exp **do**
    use $\mathbf{x}$ for input layer to produce $\mathbf{y} = f(\mathbf{x}, \theta_T)$, where $\theta_T \in$ "VAE-Wireless"
    use $\mathbf{y}$ for physical wireless layer to produce $\mathbf{w} = \tau(\mathbf{y})$
    for each n **do**
      use $\mathbf{w}$ for sampling layer input to produce the LRVs $\mathbf{z}$ values using (6.15).
      use the sampling layer output to reconstruct the PHV :
      $\widehat{\mathbf{x}} = f^{-1}(\mathbf{w}, \theta_R)$ where $\theta_R \in$ "VAE-Wireless"
      Use $\widehat{\mathbf{x}}$ as input for the "Classifier-PHV" model
      provide the final class of received message $\widehat{x} \Rightarrow \widehat{s}_d \in \{1, ..., m, ..., M\}$
    end for
    find the $\widehat{s}_d = mode\left((\widehat{s}_d^n)_{n=1}^N\right)$ for the N re-samples message.
    compare $\widehat{s}_d$ to s_d
  end for
  Calculate the MER at specific E_b/N_o
end for
Output: Return MER for all E_b/N_o. =0

6.6. Numerical Results

In this section, a series of experiments will be implemented to evaluate the performance of the new approach proposed under various scenarios and compared with several benchmarks. In particular, we consider QPSK modulation in AWGN and BPSK modulation with the effects of AWGN, fading, shadowing and Doppler on the model. We compare our results with the commonly used QPSK and BPSK expert modulation schemes which have long been used [180].

We start this section with the training process by using BPSK modulation in AWGN with the parameter settings recommended by Adam [100]. To begin with, we fixed the learning rate to 0.001 and increased the batch size from 32 to 128. From the simulation results shown in Fig. 6.8, we can see that all the curves have a similar trend, but the curve for batch size = 64 is smoother and more stable than the other curves. This is due to the effect of underfitting and overfitting the data while calculating the loss function at the training stage [184]. As a result, we choose batch size = 64 in our training process.

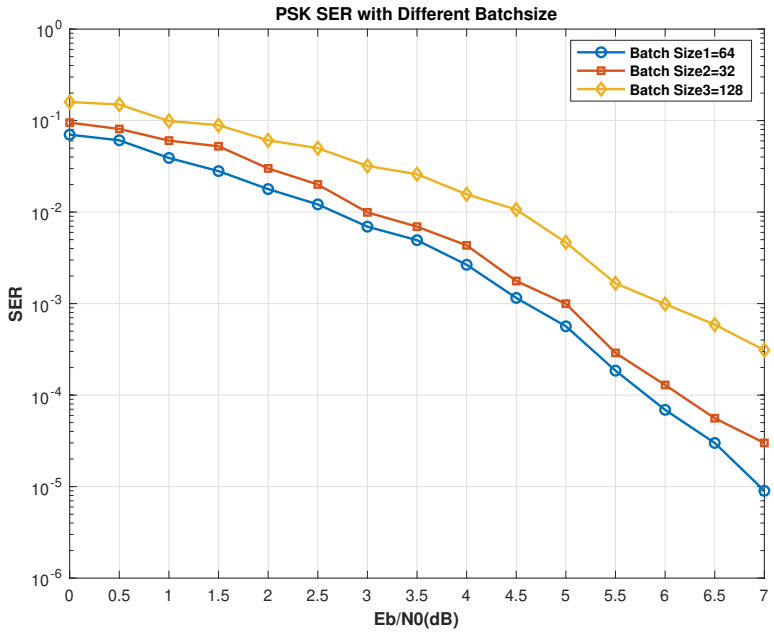


Figure 6.8: SER performance with fixed learning rate versus Eb/No for different batch sizes.

Next, it is important to choose appropriate learning parameters. The parameters are adjusted by observing the SER values as shown in Fig. 6.8, and Fig. 6.9. In this case, we fixed the batch size to 64 and increased the learning rate from 0.0001 to 0.01. The lowest SER can be obtained with a learning rate = 0.001, and the learning rate value we used in our training procedure was 0.001. The results using $\eta_Y = 0.0001$ show deterioration in SER as the search for the optimal solution required more iterations than the used one (in this work, the iterations: 300 iteration/epoch $\times$ 50 epoch=1500 iterations).

On the other hand, choosing $\eta_Y = 0.01$ produces results between the different choices due to utilising the iterations but with less resolution in loss function [185].

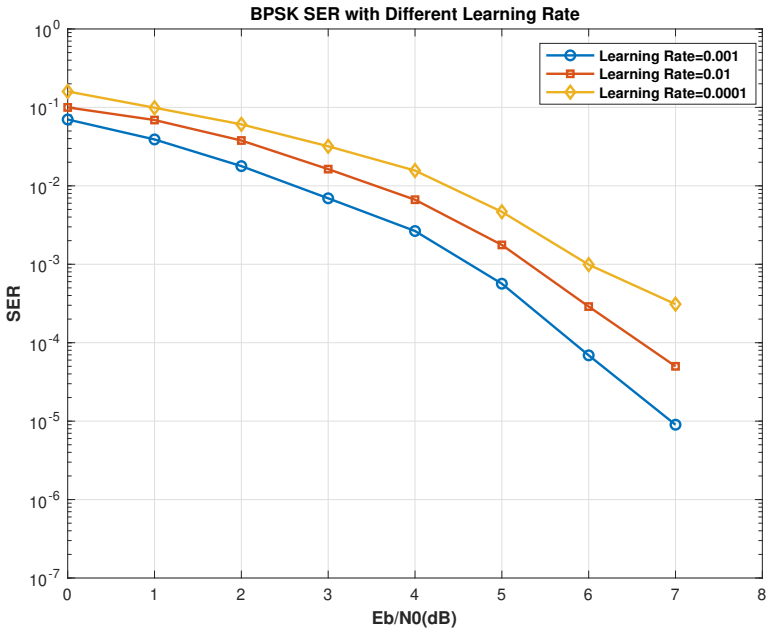


Figure 6.9: SER performance with fixed batch size versus Eb/No for different learning rate based.

Similarly, with a fixed learning rate and different batch sizes, we observe a similar trend in SER. As the E_b/N_o increase, the SER constantly decreases.

Having established the feasible learning parameters, we simulate the performance of the proposed algorithm as follows:

6.6.1. BPSK Channel

The numerically computed SER values versus $E_b/N_o \in [0,9]$ dB with BPSK modulation in AWGN are depicted in Fig. 6.10. The proposed VAE with two LRVs is capable of reconstructing the transmitted message by only sending the LRVs' parameters $(\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z)$, and the MER (in our work, $MER = SER$) decreases when the E_b/N_o increases as the green curve shows. As the AE and VAE state-of-the-art articles assume that the encoder output has decimal output values only, we used Hamming code to add protection and correction to the binary transmitted values of the encoder output after converting it to binary by adding two bits for each transmitted bit over the physical layer. However, when comparing the numerical performance of the VAE SER with the theoretical Hamming (3,1) decoded by the hard-decision method, our proposed VAE outperforms Hamming (3,1), as shown

in Fig.6.10. Moreover, the proposed VAE outperforms the hard-decision-decoded Golay scheme with a semi-constant gap (parallel) with an average of 0.5 dB.

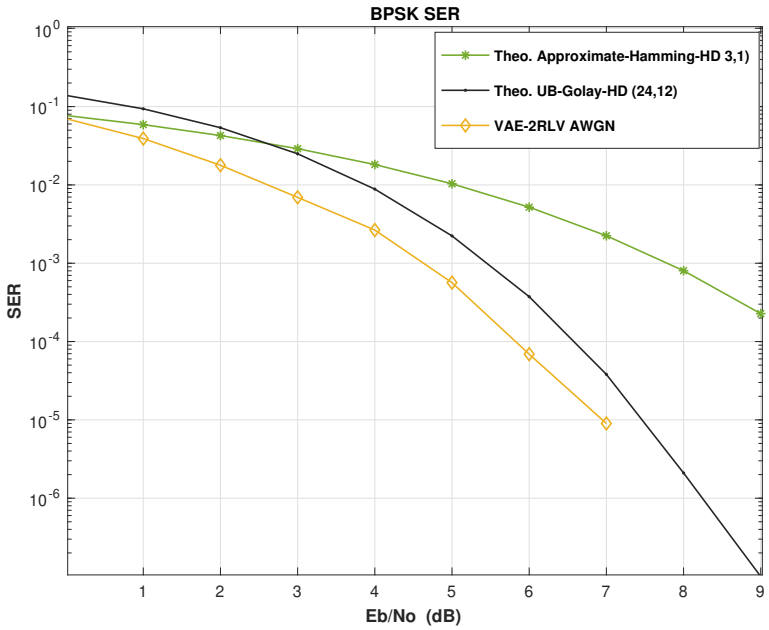


Figure 6.10: SER performance of the BPSK VAE vs baseline schemes.

Moreover, comparing the performance of the VAE SER to the classical AE [3], the dashed curves show that at the same number of channels used to transmit the encoder outputs (AE(1,4) in brown), the proposed VAE outperforms the AE scheme, as shown in Fig. 6.11. However, as the number of channels of the AE increased, the performance gap decreased as the blue dashed curve AE(7,4) in comparison to the amber curve (VAE with 2 LRVs), which means that the VAE use fewer channels than the classical AE to achieve the same SER numerical performance.

6.6.2. QPSK Channel

Fig. 6.12 shows a similar comparison, but for a higher-order modulation scheme, specifically, quadrature phase shift keying (QPSK) under AWGN channel to the classical AE [3] and the proposed VAE. This result shows that the proposed VAE with different modulation (BPSK and QPSK) achieve better performance than the classical AE. Notice that even the QPSK VAE still perform better than AE (7,4) at low E_b/N_O as in [3].

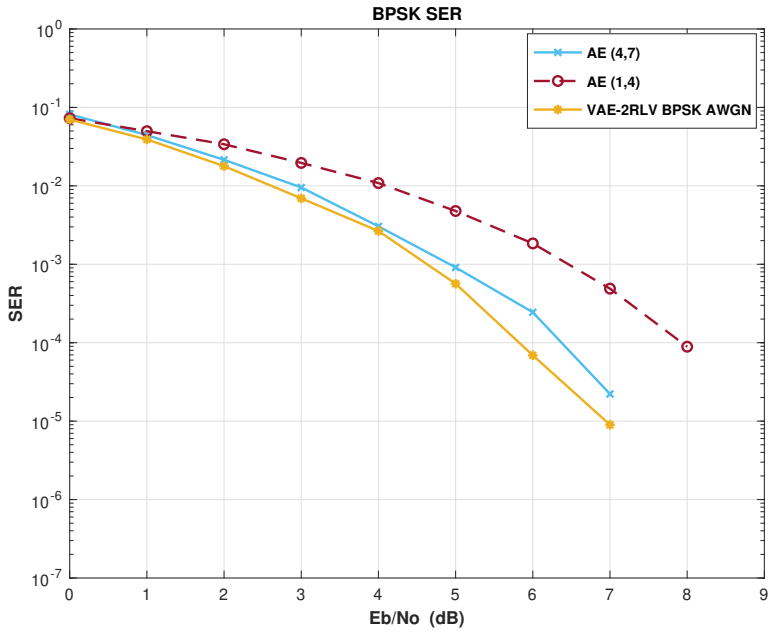


Figure 6.11: SER performance of the BPSK VAE vs AE baseline schemes.

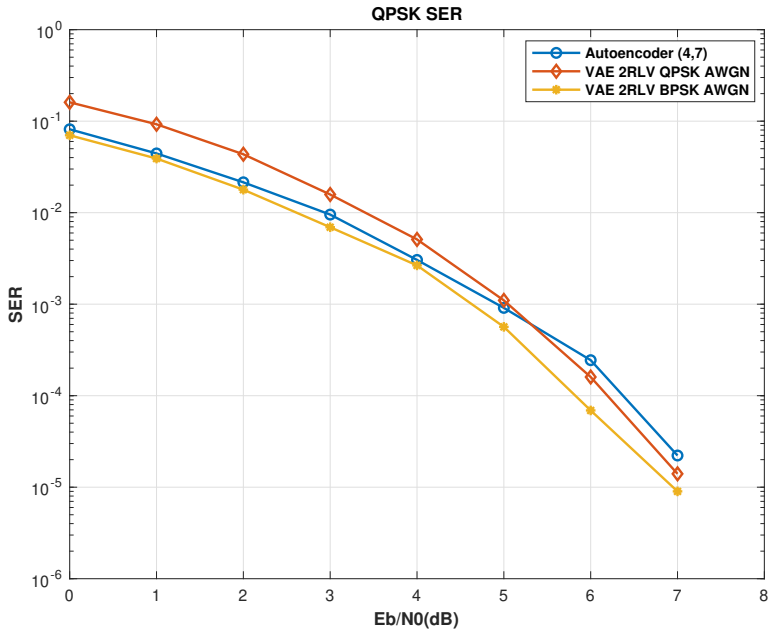


Figure 6.12: SER performance BPSK and QPSK of the VAE scheme under AWGN vs AE baseline schemes.

6.6.3. Rayleigh Fading Channel

The numerically computed SER values versus $E_b/N_o \in [0,20]$ dB with Rayleigh are depicted in Fig. 6.13. The proposed VAE with two LRVs is capable of reconstructing the transmitted message by only sending the LRVs' parameters $(\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z)$ and the SER decreases as the E_b/N_o increases, as in Fig. 6.13. As with BPSK VAE SER performance, when comparing the VAE SER numerical performance with the theoretical Rayleigh [180], our proposed VAE with Rayleigh outperforms the theoretical one.

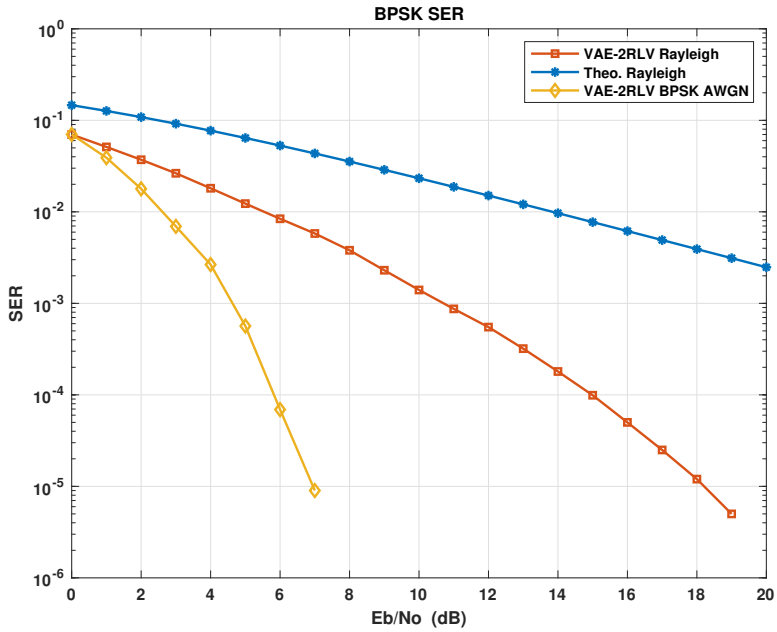


Figure 6.13: SER performance of the BPSK VAE scheme under AWGN vs Rayleigh channel.

6.6.4. Rician Fading Channel

The numerically computed SER values versus $E_b/N_o \in [0,16]$ dB with Rician are shown in Fig. 6.14. The proposed VAE with two LRVs is capable of reconstructing the transmitted message by only sending the LRVs' parameters $(\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z)$, and the MER (in the BPSK case the $MER = SER$) decreases as the E_b/N_o increases, as shown in Fig. 6.14. The numerical performance of VAE SER with the different Rician factors that measure the relative strength of the line of sight (LoS) component and measure the severity of fading, with $K = 2$ being the most

severe case of fading (very close to Rayleigh fading), while $K = 14$ represents almost no fading, and at $K = 7$ lay in-between.

Our proposed VAE with Rician performs better when the value of k increases until it gets close to the performance of the AWGN.

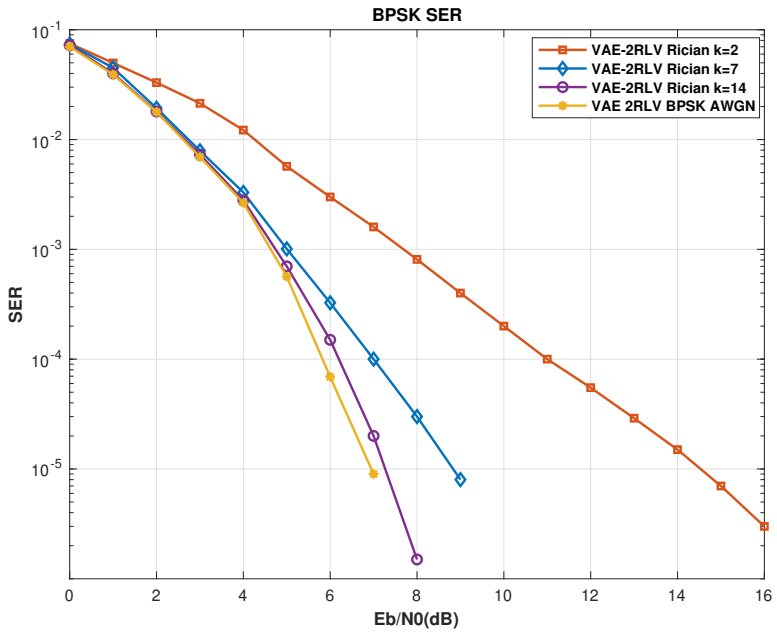


Figure 6.14: SER performance of the BPSK VAE scheme under AWGN vs Rician channel.

6.6.5. Shadowing Effect

Fig. 6.15 shows the proposed BPSK VAE SER performance compared to VAE with shadowing behaviour regarding the σ of lognormal fading for a different number of E_b/N_o. In this figure, it is possible to observe that increasing E_b/N_o in the presence of the shadowing effect decrease the SER.

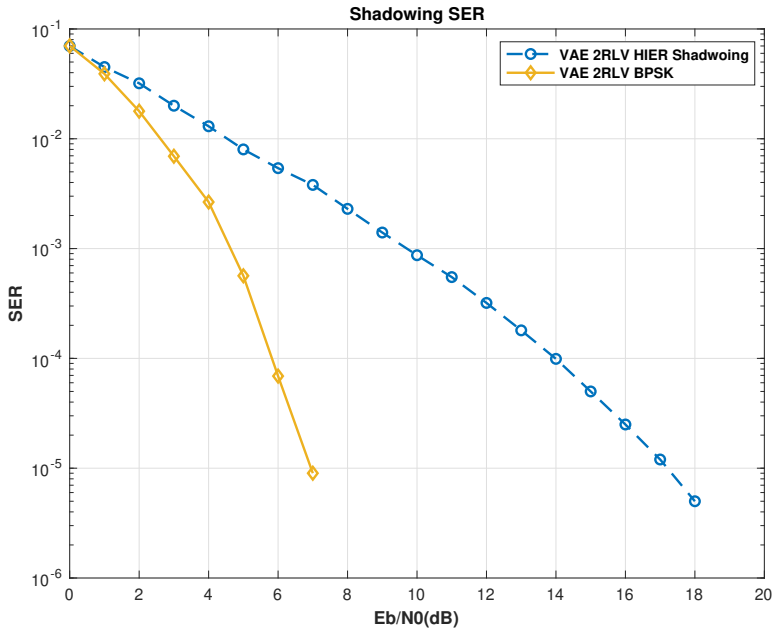


Figure 6.15: SER performance of the BPSK VAE with Shadowing σ=4.5 dB.

6.6.6. Doppler Effect

Fig. 6.16 presents the proposed VAE with a variation of the Doppler shift value under the non-stationary case. From the simulated results, we can notice that the SER increases as the Doppler shift increase if we assume that both transmitter and receiver are moving along the same axis with different phase offsets 5° and 45°, which demonstrates increasing mobility causing SER increment in compared to the stationary scenario.

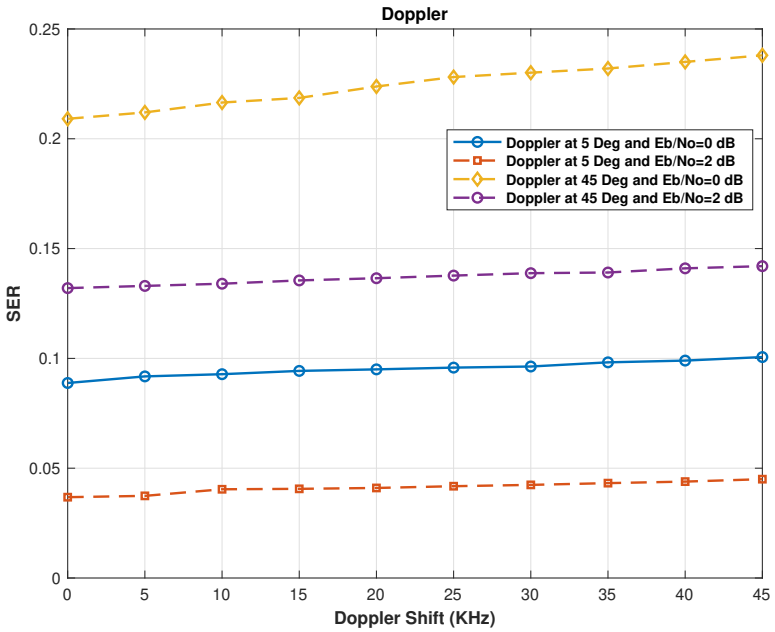


Figure 6.16: SER performance of the Doppler effect of the BPSK VAE with different phase and frequency offsets.

6.6.7. Further Results Discussion

We extend our experiment to investigate the shadowing, Rayleigh and Rician fading channels in addition to the AWGN. Moreover, the QPSK modulation under AWGN has been used to investigate the possibility of applying higher modulation schemes, which provides promising insight. However, due to the work limitation in focusing on the proof of the proposed concept where short packets can be transmitted through a wireless E2E VAE-based system. Further work can be conducted to find the SER performance for 64PSK and 128PSK. Furthermore, the Doppler effect has been added to the experiment to show that the proposed design has potential for non-stationary scenarios, and it requires to study of more varying channel parameters in order to overcome the limitation of this experiment from the perspective of the transmitter-receiver mobility.

6.7. Chapter Summary

This chapter introduced a novel approach using the VAE as a probabilistic model to reconstruct the transmitted symbol by transmitting the statistical parameters

of the LRVs through the physical layer instead of sending the data bits of the original symbol out of the transmitter. We show significantly improved PHV or SER performance compared to the baseline Hamming code with hard decision decoding and classical AE end-to-end, where increasing the E_b/N_o improves the SER of the proposed system in comparison to the baseline schemes. In addition, the proposed VAE shows a promising channel utilizing efficiency in comparison to the classical AE, where the results show that the VAE with two channels (BPSK and QPSK) under AWGN, outperforms the classical AE of 4 and 7 channels schemes. Moreover, the performance of the proposed approach in the presence of fading (Rayleigh, Rician and shadowing) is promising too, as the results show the performance improvement towards the BPSK VAE SER. Furthermore, other cases, such as the Doppler effect, have been simulated and discussed, showing that the proposed model can be generalized to the case in which the LVRs' parameters are transmitted rather than the original bits. Our findings illustrate the importance of using the VAE approach and may inspire other researchers to use a similar approach for future communication systems. Nevertheless, while we are concentrating on the proof of the proposed concept, there are some limitations to our work; further work can be conducted to find the SER performance for 64PSK and 128PSK. In addition, this paper probes the applications of the proposed design for a non-stationary case. Further investigation is required for both high mobility and higher modulation schemes to find how such limitations can be overcome.

Chapter 7

Summary, Conclusion and Future Work

7.1. Conclusion

In this thesis, we explore how machine learning techniques can be applied to some of the most advanced wireless communication technologies currently available and demonstrate how the potential and impact of such techniques can affect the future performance of wireless communication networks and future research areas. In accordance with the five research objectives set forth in section 1.3, our research contributes four new accomplishments to the literature for B5G enabling technologies such as end-to-end optimized wireless communication systems.

7.2. Summary

The study presented in this thesis can be summarized below:

In chapter 2, the background information on wireless communication, including DL and ML, was presented. An overview of DL for the physical layer of the communication system was presented, including background, applications, advantages, challenges, basics and related works.

Chapter 3 focused on the concept of AE as an end-to-end communications system and compared it with the conventional communication system. In addition, we analyse the performance of the AE as an end-to-end learning system and

compare it with the conventional communication system with different modulation schemes with the intention of achieving a better or similar BER.

In chapter 4, we illustrated the effectiveness of employing a DNN for the detection of IM-MIMO-OFDM in terms of error performance. The proposed model, DLIM, trained the received signal and channel vectors offline based on domain knowledge, such as a ZF equaliser and energy detection. Then, the fully connected DNN was trained to recover the data bits.

In chapter 5, the classical IRS-assisted communication system has been transformed into an end-to-end AE. Additionally, we proposed DNN loss function that shows the capability of optimizing the DNN parameters to reconstruct the transmitted signal, by learning signal constellations representation pattern that reduces the wireless environment contamination impact on the received signal.

In chapter 6, a model for the classical IRS-assisted communication system has been proposed based on an end-to-end AE. It is evident from the numerical results that the proposed design models both the channels through which the desired signal propagates as well as the IRS augmentation effect on the routed signal as a single DNN with loss function. It has been shown that the parameters of the DNN can be optimized to reconstruct the transmitted signal by optimizing DNN parameters.

7.3. Future Work

By exploring different machine learning techniques along with classical optimization, this thesis illustrates an awareness of the potential to enhance the performance of next-generation wireless networks using machine learning techniques. Some of the potential future directions of this research are listed below:

7.3.1. End-to-End Communication System Using Masked Auto-encoder (MAE)

Since its appearance, the Transformer [186], has been integrated in DL framework. Inspired by the semantic level communication, we will try to design a novel Masked Auto-encoder (MAE) based communication system to improve the reliability of AE for the end-to-end physical layer. The design of the system will be a

simple, effective, and scalable form of a masked autoencoder (MAE) for short-packet communications. That is reliable in the communications system, including asymmetric encoder-decoder architecture powered by NNs.

Our goal is to design an MAE model for an end-to-end communication system. Our MAE eliminates the low-level information by masking a large portion of the input packet patches, where the encoder takes only the visible patches while the lightweight decoder reconstructs the input from the latent representation. Due to the asymmetric encoder-decoder architecture of our MAE, a very high masking ratio of the input ($e.g., 75\%$) yields to achieve increase the optimization accuracy and at the same time allows the encoder to process only a small portion of the patch ($e.g., 25\%$). This can result in reducing the overall pre-training time and reducing memory consumption.

Moreover, we aim to design a neat and efficient scheme of MAE for end-to-end learning representation through masked message modelling, which masks a portion of the input signal and predicts the original message back at the masked part of the signal. The framework consists of:

- Masking approach: this component designs how to select the patch to mask and how to implement the masking before using it as an input.
- The encoder extracts the latent representation for the masked patch, which is then used to predict the signals at the masked patch.
- The decoder maps the latent representation back to the input to predict the missing patch that contains rich semantic information.

7.3.2. Terahertz (THz) Communication Technologies for the Sixth Generation (6G)

Research on how ML can be used to improve the throughput of small cells using the IRS technique by using the THz band wireless communications ML application.

In spite of the fact that mmWave spectrum may seem important for 5G communications, the demand for even higher data rates of up to terabits per second requires the utilization of 0.1-10 THz for beyond 5G applications [187].

By utilizing THz spectrum, existing wireless communication systems can be expanded to include novel applications such as ultra-high-speed wireless links for virtual reality and augmented reality, reliable wireless backhaul and

access networks, the Internet of nano-things, etc. However, the use of THz frequency suffers from severe propagation and penetration losses, which limits the communication distance of the system. Therefore, it becomes challenging to communicate even at distances of ten meters.

There are many challenges associated with distance in mmWave and THz frequency spectrum, requiring solutions such as distance-aware physical layer design, ultra-mMIMO communication, intelligent surfaces, and reflect arrays. It has been demonstrated that mmWave and THz spectrum are viable and beneficial in the context of line-of-sight and non-line-of-sight communication scenarios for distances of approximately 100 meters [187].

7.3.3. Other Extensions

- The work in chapter 5 can be extended to further examine the system performance after replacing the AE with VAE to be an IRS-assisted communication system into an end-to-end VAE.

- Lastly, investigate the use of transformer architectures [186] with VAE as a probabilistic model using a semantic communication approach based on AE for the wireless relay channel (AESC) to extract and compress semantic information and reconstruct its semantic features.

Bibliography

[1] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3072–3108, 2019.

[2] Q. Wu, S. Zhang, B. Zheng, C. You, and R. Zhang, "Intelligent reflecting surface-aided wireless communications: A tutorial," *IEEE Transactions on Communications*, vol. 69, no. 5, pp. 3313–3351, 2021.

[3] T. O'shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[4] E. Basar, M. Di Renzo, J. De Rosny, M. Debbah, M.-S. Alouini, and R. Zhang, "Wireless communications through reconfigurable intelligent surfaces," *IEEE access*, vol. 7, pp. 116 753–116 773, 2019.

[5] C. Zhang, P. Zhou, C. Li, and L. Liu, "A convolutional neural network for leaves recognition using data augmentation," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. IEEE, 2015, pp. 2143–2150.

[6] H. Zen and A. Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2014, pp. 3844–3848.

[7] R. Socher, Y. Bengio, and C. D. Manning, "Deep learning for nlp (without magic)," in *Tutorial Abstracts of ACL 2012*, 2012, pp. 5–5.

[8] R. Shwartz-Ziv and N. Tishby, "Opening the black box of deep neural networks via information," *arXiv preprint arXiv:1703.00810*, 2017.

[9] E. Björnson, Ö. Özdogan, and E. G. Larsson, "Intelligent reflecting surface versus decode-and-forward: How large surfaces are needed to beat relaying?" *IEEE Wireless Communications Letters*, vol. 9, no. 2, pp. 244–248, 2019.

[10] S. Khan, K. S. Khan, N. Haider, and S. Y. Shin, "Deep-learning-aided detection for reconfigurable intelligent surfaces," *arXiv preprint arXiv:1910.09136*, 2019.

[11] J. Bruck and M. Blaum, "Neural networks, error-correcting codes, and polynomials over the binary n-cube," *IEEE Transactions on information theory*, vol. 35, no. 5, pp. 976–987, 1989.

[12] I. Ortuno, M. Ortuno, and J. A. Delgado Penin, "Error correcting neural networks for channels with gaussian noise," in *?* IEEE AND INN SOCS. 1992 IEEE, 1992, pp. 265–268.

[13] A. Fehske, J. Gaeddert, and J. H. Reed, "A new approach to signal classification using spectral correlation and neural networks," in *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.* IEEE, 2005, pp. 144–150.

[14] E. E. Azzouz and A. K. Nandi, "Modulation recognition using artificial neural networks," in *Automatic Modulation Recognition of Communication Signals*. Springer, 1996, pp. 132–176.

[15] C.-K. Wen, S. Jin, K.-K. Wong, J.-C. Chen, and P. Ting, "Channel estimation for massive mimo using gaussian-mixture bayesian learning," *IEEE Transactions on Wireless Communications*, vol. 14, no. 3, pp. 1356–1368, 2014.

[16] S. Chen, G. Gibson, C. Cowan, and P. Grant, "Adaptive equalisation of finite non-linear channels using multilayer perceptrons," *Signal processing*, vol. 20, no. 2, pp. 107–119, 1990.

[17] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, "Core vector machines: Fast svm training on very large data sets," *Journal of Machine Learning Research*, vol. 6, no. Apr, pp. 363–392, 2005.

[18] B. S. Everitt, S. Landau, M. Leese, and D. Stahl, "Miscellaneous clustering methods," *Cluster analysis*, pp. 215–255, 2011.

[19] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[20] A. Burkov, *The hundred-page machine learning book*. Andriy Burkov Quebec City, Can., 2019.

[21] R. S. Sutton and A. G. Barto, "Reinforcement learning: an introduction cambridge," *MA: MIT Press.[Google Scholar]*, 1998.

[22] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Communications Surveys Tutorials*, vol. 21, no. 4, pp. 3072–3108, 2019.

[23] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[26] T. Wang, C.-K. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep learning for wireless physical layer: Opportunities and challenges," *China Communications*, vol. 14, no. 11, pp. 92–111, 2017.

[27] A. K. Nandi and E. E. Azzouz, "Algorithms for automatic modulation recognition of communication signals," *IEEE Transactions on communications*, vol. 46, no. 4, pp. 431–436, 1998.

[28] X.-A. Wang and S. B. Wicker, "An artificial neural net viterbi decoder," *IEEE Transactions on communications*, vol. 44, no. 2, pp. 165–171, 1996.

[29] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2016, pp. 341–346.

[30] T. Gruber, S. Cammerer, J. Hoydis, and S. ten Brink, "On deep learning-based channel decoding," in *2017 51st Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2017, pp. 1–6.

[31] S. Cammerer, T. Gruber, J. Hoydis, and S. Ten Brink, "Scaling deep learning-based decoding of polar codes via partitioning," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[32] S. Dörner, S. Cammerer, J. Hoydis, and S. ten Brink, "Deep learning based communication over the air," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, 2017.

[33] T. J. O'Shea, T. Roy, N. West, and B. C. Hilburn, "Physical layer communications system design over-the-air using adversarial networks," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 529–532.

[34] F. A. Aoudia and J. Hoydis, "End-to-end learning of communications systems without a channel model," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2018, pp. 298–303.

[35] M. Goutay, F. A. Aoudia, and J. Hoydis, "Deep reinforcement learning autoencoder with noisy feedback," *arXiv preprint arXiv:1810.05419*, 2018.

[36] D. Wu, M. Nekovee, and Y. Wang, "Deep learning based autoencoder for interference channel," *arXiv preprint arXiv:1902.06841*, 2019.

[37] D. Wu, Y. Wang, and M. Nekovee, "An adaptive deep learning algorithm based autoencoder for interference channels," in *International Conference on Machine Learning for Networking*. Springer, 2019, pp. 342–354.

[38] A. Sahin, I. Guvenc, and H. Arslan, "A survey on multicarrier communications: Prototype filters, lattice structures, and implementation aspects," *IEEE communications surveys & tutorials*, vol. 16, no. 3, pp. 1312–1338, 2013.

[39] P. Banelli, S. Buzzi, G. Colavolpe, A. Modenini, F. Rusek, and A. Ugolini, "Modulation formats and waveforms for 5g networks: Who will be the heir of ofdm?: An overview of alternative modulation schemes for improved spectral efficiency," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 80–93, 2014.

[40] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5g be?" *IEEE Journal on selected areas in communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[41] H. Haas, S. Sinanovic, C. Ahn, and S. Yun, "Spatial modulation," *IEEE Trans. Veh. Technol*, vol. 57, no. 4, pp. 2228–41, 2008.

[42] C.-X. Wang, F. Haider, X. Gao, X.-H. You, Y. Yang, D. Yuan, H. M. Aggoune, H. Haas, S. Fletcher, and E. Hepsaydir, "Cellular architecture and key technologies for 5g wireless communication networks," *IEEE communications magazine*, vol. 52, no. 2, pp. 122–130, 2014.

[43] J. Jeganathan, A. Ghrayeb, L. Szczecinski, and A. Ceron, "Space shift keying modulation for mimo channels," *IEEE Transactions on Wireless Communications*, vol. 8, no. 7, pp. 3692–3703, 2009.

[44] A. Younis, N. Serafimovski, R. Mesleh, and H. Haas, "Generalised spatial modulation," in *2010 conference record of the forty fourth Asilomar conference on signals, systems and computers*. IEEE, 2010, pp. 1498–1502.

[45] E. Başar, U. Aygölü, E. Panayirci, and H. V. Poor, "Space-time block coded spatial modulation," *IEEE transactions on communications*, vol. 59, no. 3, pp. 823–832, 2010.

[46] E. Basar, U. Aygolu, E. Panayirci, and H. V. Poor, "Performance of spatial modulation in the presence of channel estimation errors," *IEEE Communications Letters*, vol. 16, no. 2, pp. 176–179, 2011.

[47] M. Di Renzo and H. Haas, "Bit error probability of sm-mimo over generalized fading channels," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 1124–1144, 2012.

[48] N. Serafimovski, A. Younis, R. Mesleh, P. Chambers, M. Di Renzo, C.-X. Wang, P. M. Grant, M. A. Beach, and H. Haas, "Practical implementation of spatial modulation," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4511–4523, 2013.

[49] M. Di Renzo, H. Haas, A. Ghrayeb, S. Sugiura, and L. Hanzo, "Spatial modulation for generalized mimo: Challenges, opportunities, and implementation," *Proceedings of the IEEE*, vol. 102, no. 1, pp. 56–103, 2013.

[50] E. Başar, Ü. Aygölü, E. Panayırcı, and H. V. Poor, "Orthogonal frequency division multiplexing with index modulation," *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5536–5549, 2013.

[51] R. Abu-Alhiga and H. Haas, "Subcarrier-index modulation ofdm," in *2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2009, pp. 177–181.

[52] D. Tsonev, S. Sinanovic, and H. Haas, "Enhanced subcarrier index modulation (sim) ofdm," in *2011 IEEE GLOBECOM Workshops (GC Wkshps)*. IEEE, 2011, pp. 728–732.

[53] Y. Ko, "A tight upper bound on bit error rate of joint ofdm and multi-carrier index keying," *IEEE Communications Letters*, vol. 18, no. 10, pp. 1763–1766, 2014.

[54] R. Fan, Y. J. Yu, and Y. L. Guan, "Generalization of orthogonal frequency division multiplexing with index modulation," *IEEE transactions on wireless communications*, vol. 14, no. 10, pp. 5350–5359, 2015.

[55] M. Wen, X. Cheng, and L. Yang, "Optimizing the energy efficiency of ofdm with index modulation," in *2014 IEEE International Conference on Communication Systems*. IEEE, 2014, pp. 31–35.

[56] W. Li, H. Zhao, C. Zhang, L. Zhao, and R. Wang, "Generalized selecting sub-carrier modulation scheme in ofdm system," in *2014 IEEE International*

Conference on Communications Workshops (ICC). IEEE, 2014, pp. 907–911.

[57] Y. Xiao, S. Wang, L. Dan, X. Lei, P. Yang, and W. Xiang, "Ofdm with interleaved subcarrier-index modulation," *IEEE Communications Letters*, vol. 18, no. 8, pp. 1447–1450, 2014.

[58] X. Cheng, M. Wen, L. Yang, and Y. Li, "Index modulated ofdm with interleaved grouping for v2x communications," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014, pp. 1097–1104.

[59] E. Başar, "Ofdm with index modulation using coordinate interleaving," *IEEE Wireless Communications Letters*, vol. 4, no. 4, pp. 381–384, 2015.

[60] M. Wen, X. Cheng, M. Ma, B. Jiao, and H. V. Poor, "On the achievable rate of ofdm with index modulation," *IEEE Transactions on Signal Processing*, vol. 64, no. 8, pp. 1919–1932, 2015.

[61] B. Zheng, F. Chen, M. Wen, F. Ji, H. Yu, and Y. Liu, "Low-complexity ml detector and performance analysis for ofdm with in-phase/quadrature index modulation," *IEEE Communications Letters*, vol. 19, no. 11, pp. 1893–1896, 2015.

[62] E. Basar, "On multiple-input multiple-output ofdm with index modulation for next generation wireless networks," *IEEE Transactions on Signal Processing*, vol. 64, no. 15, pp. 3868–3878, 2016.

[63] J. Shaker, M. R. Chaharmir, and J. Ethier, *Reflectarray Antennas: Analysis, design, fabrication, and measurement*. Artech House, 2013.

[64] C. Liaskos, S. Nie, A. Tsioliaridou, A. Pitsillides, S. Ioannidis, and I. Akyildiz, "A new wireless communication paradigm through software-controlled metasurfaces," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 162–169, 2018.

[65] E. Björnson, L. Sanguinetti, H. Wymeersch, J. Hoydis, and T. L. Marzetta, "Massive mimo is a reality—what is next?: Five promising research directions for antenna arrays," *Digital Signal Processing*, vol. 94, pp. 3–20, 2019.

[66] M. D. Renzo, M. Debbah, D.-T. Phan-Huy, A. Zappone, M.-S. Alouini, C. Yuen, V. Sciancalepore, G. C. Alexandropoulos, J. Hoydis, H. Gacanin *et al.*, "Smart radio environments empowered by reconfigurable ai meta-surfaces: An idea whose time has come," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 1–20, 2019.

[67] Q. Wu and R. Zhang, "Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5394–5409, 2019.

[68] V. S. Asadchy, M. Albooyeh, S. N. Tcvetkova, A. Díaz-Rubio, Y. Ra'di, and S. Tretyakov, "Perfect control of reflection and refraction using spatially dispersive metasurfaces," *Physical Review B*, vol. 94, no. 7, p. 075142, 2016.

[69] N. M. Estakhri and A. Alu, "Wave-front transformation with gradient metasurfaces," *Physical Review X*, vol. 6, no. 4, p. 041008, 2016.

[70] C. Pan, H. Ren, K. Wang, J. F. Kolb, M. Elkashlan, M. Chen, M. Di Renzo, Y. Hao, J. Wang, A. L. Swindlehurst *et al.*, "Reconfigurable intelligent surfaces for 6g systems: Principles, applications, and research directions," *IEEE Communications Magazine*, 2021.

[71] Ö. Özdogan, E. Björnson, and E. G. Larsson, "Intelligent reflecting surfaces: Physics, propagation, and pathloss modeling," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 581–585, 2019.

[72] M. A. S. Sejan, M. H. Rahman, B.-S. Shin, J.-H. Oh, Y.-H. You, and H.-K. Song, "Machine learning for intelligent-reflecting-surface-based wireless communication towards 6g: A review," *Sensors*, vol. 22, no. 14, p. 5405, 2022.

[73] F. C. Okogbaa, Q. Z. Ahmed, F. A. Khan, W. B. Abbas, F. Che, S. A. R. Zaidi, and T. Alade, "Design and application of intelligent reflecting surface (irs) for beyond 5g wireless networks: A review," *Sensors*, vol. 22, no. 7, p. 2436, 2022.

[74] W. Long, R. Chen, M. Moretti, W. Zhang, and J. Li, "A promising technology for 6g wireless networks: Intelligent reflecting surface," *Journal of Communications and Information Networks*, vol. 6, no. 1, pp. 1–16, 2021.

[75] M. Elmossallamy, K. G. Seddik, W. Chen, L. Wang, G. Li, and Z. Han, "Ris optimization on the complex circle manifold for interference mitigation in interference channels," *IEEE Transactions on Vehicular Technology*, 2021.

[76] M. Jung, W. Saad, M. Debbah, and C. S. Hong, "On the optimality of reconfigurable intelligent surfaces (riss): Passive beamforming, modulation, and resource allocation," *IEEE Transactions on Wireless Communications*, 2021.

[77] H. Wymeersch, J. He, B. Denis, A. Clemente, and M. Juntti, "Radio localization and mapping with reconfigurable intelligent surfaces: Challenges, opportunities, and research directions," *IEEE Vehicular Technology Magazine*, vol. 15, no. 4, pp. 52–61, 2020.

[78] X. Yu, D. Xu, and R. Schober, "Enabling secure wireless communications via intelligent reflecting surfaces," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[79] A. Almohamad, A. M. Tahir, A. Al-Kababji, H. M. Furqan, T. Khattab, M. O. Hasna, and H. Arslan, "Smart and secure wireless communications via reflecting intelligent surfaces: A short survey," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 1442–1456, 2020.

[80] J. Jordan, "Introduction to autoencoders," *Jeremy Jordan, Mar*, 2018.

[81] T. Ghosh, "Supervised and unsupervised training of deep autoencoder," Ph.D. dissertation, Colorado State University, 2017.

[82] V. Kazak, "Unsupervised feature extraction with autoencoder: for the representation of parkinson´ s disease patients," Ph.D. dissertation, 2019.

[83] P. Raj and P. E. David, *The Digital Twin Paradigm for Smarter Systems and Environments: The Industry Use Cases*. Academic Press, 2020.

[84] G. Dong, G. Liao, H. Liu, and G. Kuang, "A review of the autoencoder and its variants: A comparative perspective from target recognition in synthetic-aperture radar images," *IEEE Geoscience and Remote Sensing Magazine*, vol. 6, no. 3, pp. 44–68, 2018.

[85] Q. V. Le *et al.*, "A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks," *Google Brain*, pp. 1–20, 2015.

[86] H. I. Choi, "Lecture 16: Autoencoders (draft: version 0.7. 2)," 2019.

[87] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," *arXiv preprint arXiv:2003.05991*, 2020.

[88] K. G. Kim, "Book review: Deep learning," *Healthcare informatics research*, vol. 22, no. 4, pp. 351–354, 2016.

[89] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[90] M. Kim, C. Yan, D. Yang, Q. Wang, J. Ma, and G. Wu, "Deep learning in biomedical image analysis," in *Biomedical information technology*. Elsevier, 2020, pp. 239–263.

[91] G. Alain and Y. Bengio, "What regularized auto-encoders learn from the data-generating distribution," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3563–3593, 2014.

[92] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning (adaptive computation and machine learning series)," *Cambridge Massachusetts*, pp. 321–359, 2017.

[93] M. Chatterjee, "Deep learning tutorial: What it means and what's the role of deep learning," *Great Learning [website]*, 2021.

[94] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," *Advances in neural information processing systems*, vol. 29, 2016.

[95] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild!: A lock-free approach to parallelizing stochastic gradient descent," *Advances in neural information processing systems*, vol. 24, 2011.

[96] A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser, K. Kurach, and J. Martens, "Adding gradient noise improves learning for very deep networks," *arXiv preprint arXiv:1511.06807*, 2015.

[97] M. A. Nielsen, *Neural networks and deep learning*. Determination press San Francisco, CA, USA, 2015, vol. 25.

[98] I. Chakroun, T. Haber, and T. J. Ashby, "Sw-sgd: the sliding window stochastic gradient descent algorithm," *Procedia Computer Science*, vol. 108, pp. 2318–2322, 2017.

[99] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," in *Advances in neural information processing systems*, 2014, pp. 2933–2941.

[100] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[101] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[102] F. Chollet *et al.*, "keras," 2015.

[103] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[104] E. Basar, "Index modulation techniques for 5g wireless networks," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 168–175, 2016.

[105] T. Van Luong and Y. Ko, "Impact of csi uncertainty on mcik-ofdm: Tight closed-form symbol error probability analysis," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1272–1279, 2017.

[106] T. Van Luong, Y. Ko, and J. Choi, "Repeated mcik-ofdm with enhanced transmit diversity under csi uncertainty," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 4079–4088, 2018.

[107] M. Wen, B. Ye, E. Basar, Q. Li, and F. Ji, "Enhanced orthogonal frequency division multiplexing with index modulation," *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, pp. 4786–4801, 2017.

[108] J. Choi, "Coded ofdm-im with transmit diversity," *IEEE Transactions on Communications*, vol. 65, no. 7, pp. 3164–3171, 2017.

[109] T. Van Luong and Y. Ko, "Spread ofdm-im with precoding matrix and low-complexity detection designs," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 11 619–11 626, 2018.

[110] B. Zheng, M. Wen, E. Basar, and F. Chen, "Multiple-input multiple-output ofdm with index modulation: Low-complexity detector design," *IEEE Transactions on Signal Processing*, vol. 65, no. 11, pp. 2758–2772, 2017.

[111] J. Crawford and Y. Ko, "Low complexity greedy detection method with generalized multicarrier index keying ofdm," in *2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2015, pp. 688–693.

[112] T. Van Luong and Y. Ko, "A tight bound on ber of mcik-ofdm with greedy detection and imperfect csi," *IEEE Communications Letters*, vol. 21, no. 12, pp. 2594–2597, 2017.

[113] J. Jeganathan, A. Ghrayeb, and L. Szczecinski, "Spatial modulation: Optimal detection and performance analysis," *IEEE Communications Letters*, vol. 12, no. 8, pp. 545–547, 2008.

[114] M. Kim, W. Lee, and D.-H. Cho, "A novel papr reduction scheme for ofdm system based on deep learning," *IEEE Communications Letters*, vol. 22, no. 3, pp. 510–513, 2017.

[115] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in ofdm systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2017.

[116] T. Van Luong, Y. Ko, N. A. Vien, D. H. Nguyen, and M. Matthaiou, "Deep learning-based detector for ofdm-im," *IEEE Wireless Communications Letters*, vol. 8, no. 4, pp. 1159–1162, 2019.

[117] J. Liu and H. Lu, "Imnet: A learning based detector for index modulation aided mimo-ofdm systems," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2020, pp. 1–6.

[118] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah, and C. Yuen, "Reconfigurable intelligent surfaces for energy efficiency in wireless communication," *IEEE Transactions on Wireless Communications*, vol. 18, no. 8, pp. 4157–4170, 2019.

[119] B. Zheng, C. You, W. Mei, and R. Zhang, "A survey on channel estimation and practical passive beamforming design for intelligent reflecting surface aided wireless communications," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 1035–1071, 2022.

[120] M. Cui, G. Zhang, and R. Zhang, "Secure wireless communication via intelligent reflecting surface," *IEEE Wireless Communications Letters*, vol. 8, no. 5, pp. 1410–1414, 2019.

[121] S. Gong, X. Lu, D. T. Hoang, D. Niyato, L. Shu, D. I. Kim, and Y.-C. Liang, "Toward smart wireless communications via intelligent reflecting surfaces: A contemporary survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2283–2314, 2020.

[122] X. Yu, D. Xu, Y. Sun, D. W. K. Ng, and R. Schober, "Robust and secure wireless communications via intelligent reflecting surfaces," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 11, pp. 2637–2652, 2020.

[123] Q.-U.-A. Nadeem, A. Kammoun, A. Chaaban, M. Debbah, and M.-S. Alouini, "Intelligent reflecting surface assisted wireless communication: Modeling and channel estimation," *arXiv preprint arXiv:1906.02360*, 2019.

[124] M. Di Renzo, A. Zappone, M. Debbah, M.-S. Alouini, C. Yuen, J. de Rosny, and S. Tretyakov, "Smart radio environments empowered by reconfigurable intelligent surfaces: How it works, state of research, and the road ahead," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 11, pp. 2450–2525, 2020.

[125] S. V. Hum and J. Perruisseau-Carrier, "Reconfigurable reflectarrays and array lenses for dynamic antenna beam control: A review," *IEEE*

Transactions on Antennas and Propagation, vol. 62, no. 1, pp. 183–198, 2013.

[126] X. Tan, Z. Sun, D. Koutsonikolas, and J. M. Jornet, "Enabling indoor mobile millimeter-wave networks based on smart reflect-arrays," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 270–278.

[127] Y. Zhang, J. Zhang, Y. Wang, Z. Yu, and B. Zhang, "A 4-bit programmable metamaterial based on vo 2 mediums," in *2018 IEEE/MTT-S International Microwave Symposium-IMS*. IEEE, 2018, pp. 984–986.

[128] A. Taha, M. Alrabeiah, and A. Alkhateeb, "Enabling large intelligent surfaces with compressive sensing and deep learning," *arXiv preprint arXiv:1904.10136*, 2019.

[129] C. Huang, R. Mo, and C. Yuen, "Reconfigurable intelligent surface assisted multiuser miso systems exploiting deep reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 8, pp. 1839–1850, 2020.

[130] C. Liaskos, A. Tsioliaridou, S. Nie, A. Pitsillides, S. Ioannidis, and I. Akyildiz, "An interpretable neural network for configuring programmable wireless environments," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2019, pp. 1–5.

[131] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "End-to-end deep reinforcement learning for lane keeping assist," *arXiv preprint arXiv:1612.04340*, 2016.

[132] X. Jin and H.-N. Kim, "Deep learning detection in mimo decode-forward relay channels," *IEEE Access*, vol. 7, pp. 99 481–99 495, 2019.

[133] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 3, pp. 567–579, 2019.

[134] T. Matsumine, T. Koike-Akino, and Y. Wang, "Deep learning-based constellation optimization for physical network coding in two-way relay networks," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[135] A. Gupta and M. Sellathurai, "End-to-end learning-based amplify-and-forward relay networks using autoencoders," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[136] M. E. Morocho-Cayamcela, H. Lee, and W. Lim, "Machine learning for 5g/b5g mobile and wireless communications: Potential, limitations, and future directions," *IEEE access*, vol. 7, pp. 137 184–137 206, 2019.

[137] M. E. M. Cayamcela and W. Lim, "Artificial intelligence in 5g technology: A survey," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2018, pp. 860–865.

[138] N. Farsad, M. Rao, and A. Goldsmith, "Deep learning for joint source-channel coding of text," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 2326–2330.

[139] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be'ery, "Deep learning methods for improved decoding of linear codes," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 119–131, 2018.

[140] H. Kim, Y. Jiang, R. Rana, S. Kannan, S. Oh, and P. Viswanath, "Communication algorithms via deep learning," *arXiv preprint arXiv:1805.09317*, 2018.

[141] W. Xu, Z. Wu, Y.-L. Ueng, X. You, and C. Zhang, "Improved polar decoder based on deep learning," in *2017 IEEE International workshop on signal processing systems (SiPS)*. IEEE, 2017, pp. 1–6.

[142] N. Farsad and A. Goldsmith, "Detection algorithms for communication systems using deep learning," *arXiv preprint arXiv:1705.08044*, 2017.

[143] T. Diskin, N. Samuel, and A. Wiesel, "Deep mimo detection," in *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2017, pp. 1–5.

[144] D. Neumann, T. Wiese, and W. Utschick, "Learning the mmse channel estimator," *IEEE Transactions on Signal Processing*, vol. 66, no. 11, pp. 2905–2917, 2018.

[145] M. A. Alawad and K. A. Hamdi, "A deep learning-based detector for im-mimo-ofdm," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*. IEEE, 2021, pp. 1–5.

[146] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based channel estimation for beamspace mmwave massive mimo systems," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 852–855, 2018.

[147] X. Gao, S. Jin, C.-K. Wen, and G. Y. Li, "Comnet: Combination of deep learning and expert knowledge in ofdm receivers," *IEEE Communications Letters*, vol. 22, no. 12, pp. 2627–2630, 2018.

[148] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Deep learning based mimo communications," *arXiv preprint arXiv:1707.07980*, 2017.

[149] T. J. O'Shea, K. Karra, and T. C. Clancy, "Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention," in *2016 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. IEEE, 2016, pp. 223–228.

[150] Z. Qin, H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 93–99, 2019.

[151] M. A. Alawad, M. Q. Hamdan, and K. A. Hamdi, "End-to-end deep learning irs-assisted communications systems," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*. IEEE, 2021, pp. 1–6.

[152] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[153] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *International conference on machine learning*. PMLR, 2014, pp. 1278–1286.

[154] A. Felix, S. Cammerer, S. Dörner, J. Hoydis, and S. Ten Brink, "Ofdm-autoencoder for end-to-end learning of communications systems," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2018, pp. 1–5.

[155] F. A. Aoudia and J. Hoydis, "Model-free training of end-to-end communication systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2503–2516, 2019.

[156] V. Raj and S. Kalyani, "Backpropagating through the air: Deep learning at physical layer without channel models," *IEEE Communications Letters*, vol. 22, no. 11, pp. 2278–2281, 2018.

[157] H. Ye, G. Y. Li, B.-H. F. Juang, and K. Sivanesan, "Channel agnostic end-to-end learning based communication systems with conditional gan," in *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2018, pp. 1–5.

[158] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[159] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[160] P. Bellavista, L. Foschini, and A. Mora, "Decentralised learning in federated deployment environments: A system-level survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 1, pp. 1–38, 2021.

[161] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[162] A. M. Abdelhameed and M. Bayoumi, "Semi-supervised eeg signals classification system for epileptic seizure detection," *IEEE Signal Processing Letters*, vol. 26, no. 12, pp. 1922–1926, 2019.

[163] X. Chen, Y. Sun, M. Zhang, and D. Peng, "Evolving deep convolutional variational autoencoders for image classification," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 5, pp. 815–829, 2020.

[164] J. Klys, J. Snell, and R. Zemel, "Learning latent subspaces in variational autoencoders," *Advances in neural information processing systems*, vol. 31, 2018.

[165] P. Cristovao, H. Nakada, Y. Tanimura, and H. Asoh, "Generating in-between images through learned latent space representation using variational autoencoders," *IEEE Access*, vol. 8, pp. 149 456–149 467, 2020.

[166] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "Cvae-gan: fine-grained image generation through asymmetric training," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2745–2754.

[167] A. Hawkins-Hooker, F. Depardieu, S. Baur, G. Couairon, A. Chen, and D. Bikard, "Generating functional protein variants with variational autoencoders," *PLoS computational biology*, vol. 17, no. 2, p. e1008736, 2021.

[168] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, and T. Melodia, "Machine learning for wireless communications in the internet of things: A comprehensive survey," *Ad Hoc Networks*, vol. 93, p. 101913, 2019.

[169] A. A. Pourzanjani, R. M. Jiang, B. Mitchell, P. J. Atzberger, and L. R. Petzold, "Bayesian inference over the stiefel manifold via the givens representation," *Bayesian Analysis*, vol. 16, no. 2, pp. 639–666, 2021.

[170] C. Liu, J. Zhuo, P. Cheng, R. Zhang, and J. Zhu, "Understanding and accelerating particle-based variational inference," in *International Conference on Machine Learning*. PMLR, 2019, pp. 4082–4092.

[171] Q. Liu, "Stein variational gradient descent as gradient flow," *Advances in neural information processing systems*, vol. 30, 2017.

[172] C. E. Shannon, *The mathematical theory of communication, by CE Shannon (and recent contributions to the mathematical theory of communication), W. Weaver*. University of illinois Press Champaign, IL, USA, 1949.

[173] H. Ye, L. Liang, G. Y. Li, and B.-H. Juang, "Deep learning-based end-to-end wireless communication systems with conditional gans as unknown channels," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3133–3143, 2020.

[174] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3174–3183.

[175] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2663–2675, 2021.

[176] X. Luo, B. Yin, Z. Chen, B. Xia, and J. Wang, "Autoencoder-based semantic communication systems with relay channels," in *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2022, pp. 711–716.

[177] D. P. Kingma, M. Welling *et al.*, "An introduction to variational autoencoders," *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019.

[178] T. M. Mitchell and T. M. Mitchell, *Machine learning*. McGraw-hill New York, 1997, vol. 1, no. 9.

[179] M. Q. Hamdan and K. A. Hamdi, "Variational auto-encoders application in wireless vehicle-to-everything communications," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–6.

[180] J. Oetting, "A comparison of modulation techniques for digital radio," *IEEE Transactions on communications*, vol. 27, no. 12, pp. 1752–1762, 1979.

[181] C. Neipp, A. Hernández, J. Rodes, A. Márquez, T. Beléndez, and A. Beléndez, "An analysis of the classical doppler effect," *European journal of physics*, vol. 24, no. 5, p. 497, 2003.

[182] V. Raj and S. Kalyani, "Design of communication systems using deep learning: A variational inference perspective," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1320–1334, 2020.

[183] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[184] H. Zhang, L. Zhang, and Y. Jiang, "Overfitting and underfitting analysis for deep learning based end-to-end communication systems," in *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 2019, pp. 1–6.

[185] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.

[186] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[187] P. Popovski, Č. Stefanović, J. J. Nielsen, E. De Carvalho, M. Angjelichinoski, K. F. Trillingsgaard, and A.-S. Bana, "Wireless access in ultra-reliable low-latency communication (urllc)," *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5783–5801, 2019.